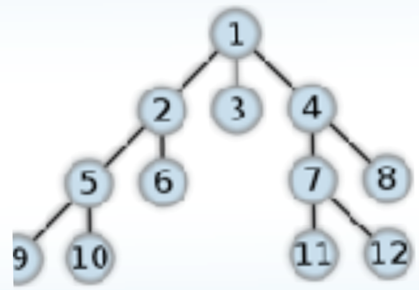


Kapitel 3.8: Laufzeit von DFS und BFS

Algorithmen und Datenstrukturen
WS 2020/21

Prof. Dr. Sándor Fekete

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y $\subseteq$ V, die von s aus erreichbar ist,

Kantenmenge T $\subseteq$ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v $\in$ R

    2.2. IF (es gibt kein w $\in$ V\Y mit e={v,w} $\in$ E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

        2.3.1. Wähle ein w $\in$ V\Y mit e={v,w} $\in$ E

        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }

3. STOP

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    2.1. Wähle $v\in R$
    2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

   2.1. Wähle v∈ R

   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

      2.2.1. R:=R\{v}

   2.3. ELSE {

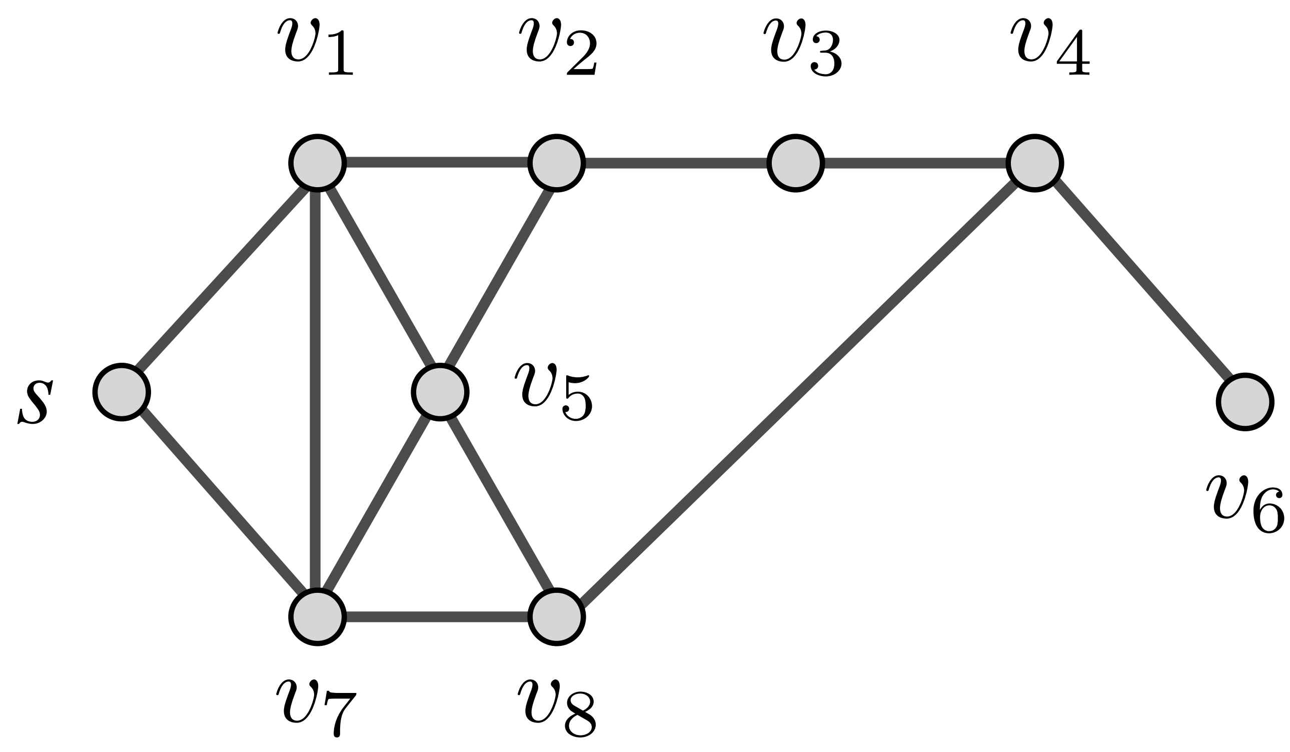
      2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

      2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

      }

   }

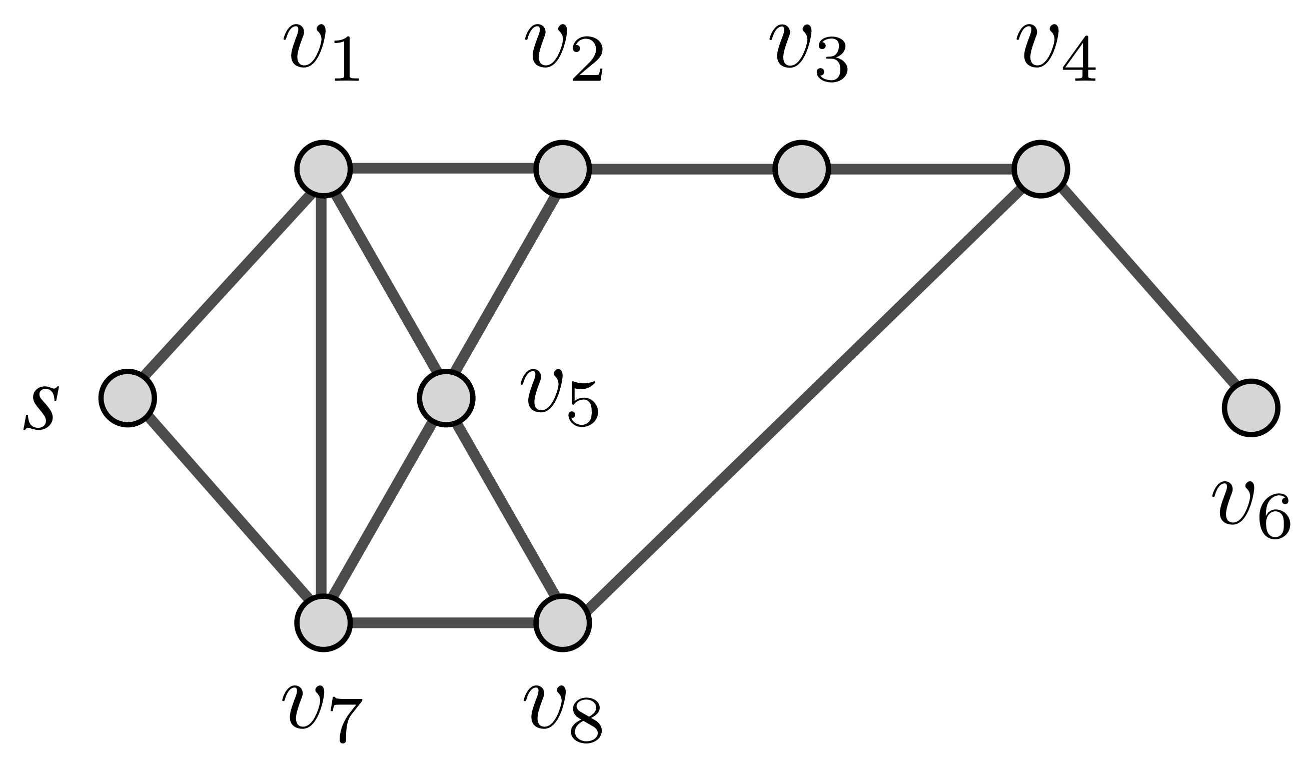
3. STOP



Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
   }
3. STOP



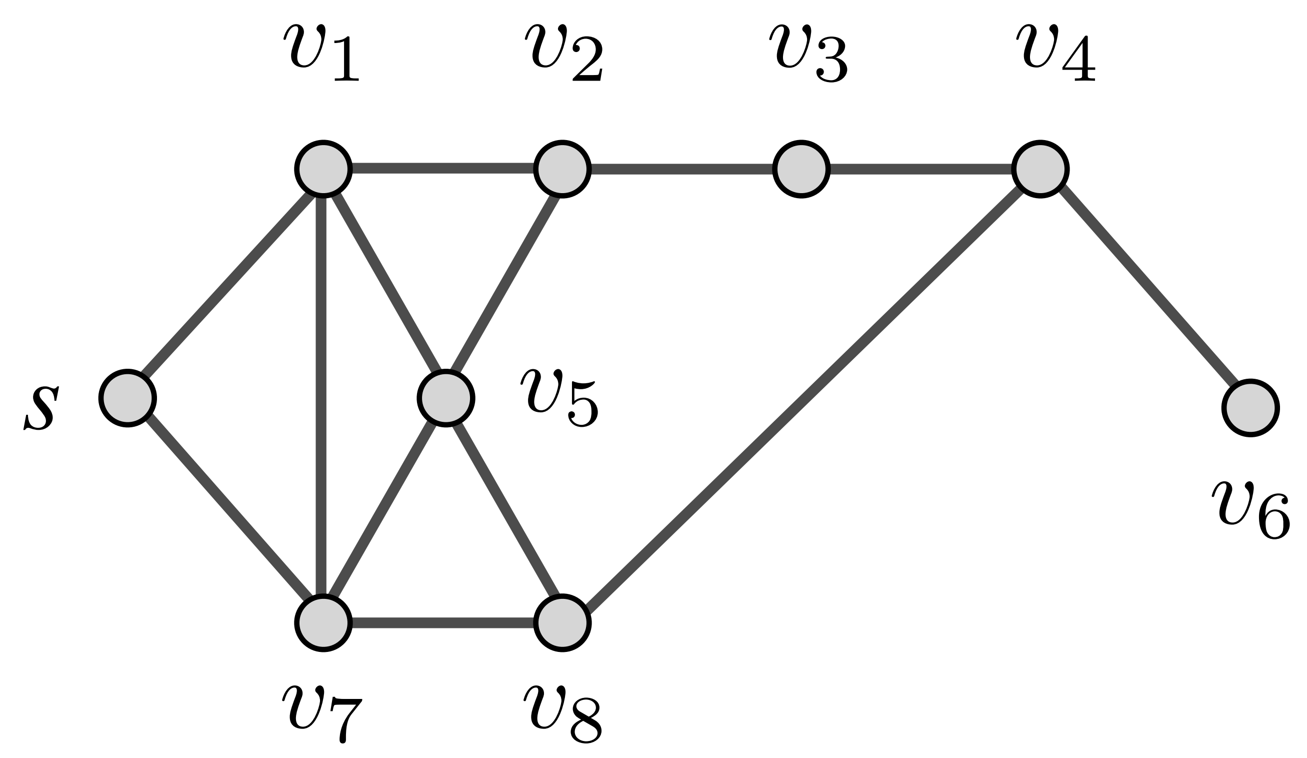
Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }
   - }
3. STOP



s

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

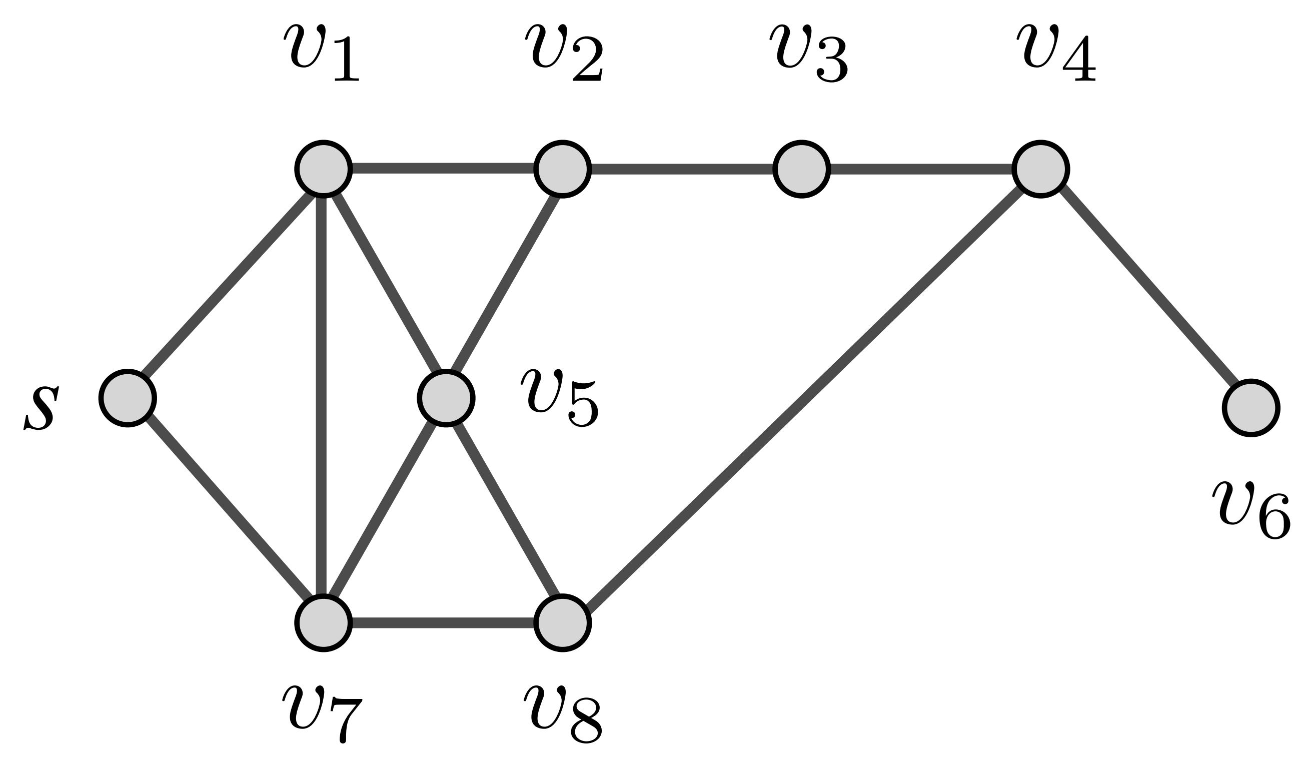
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

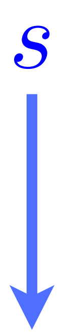
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈ E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP



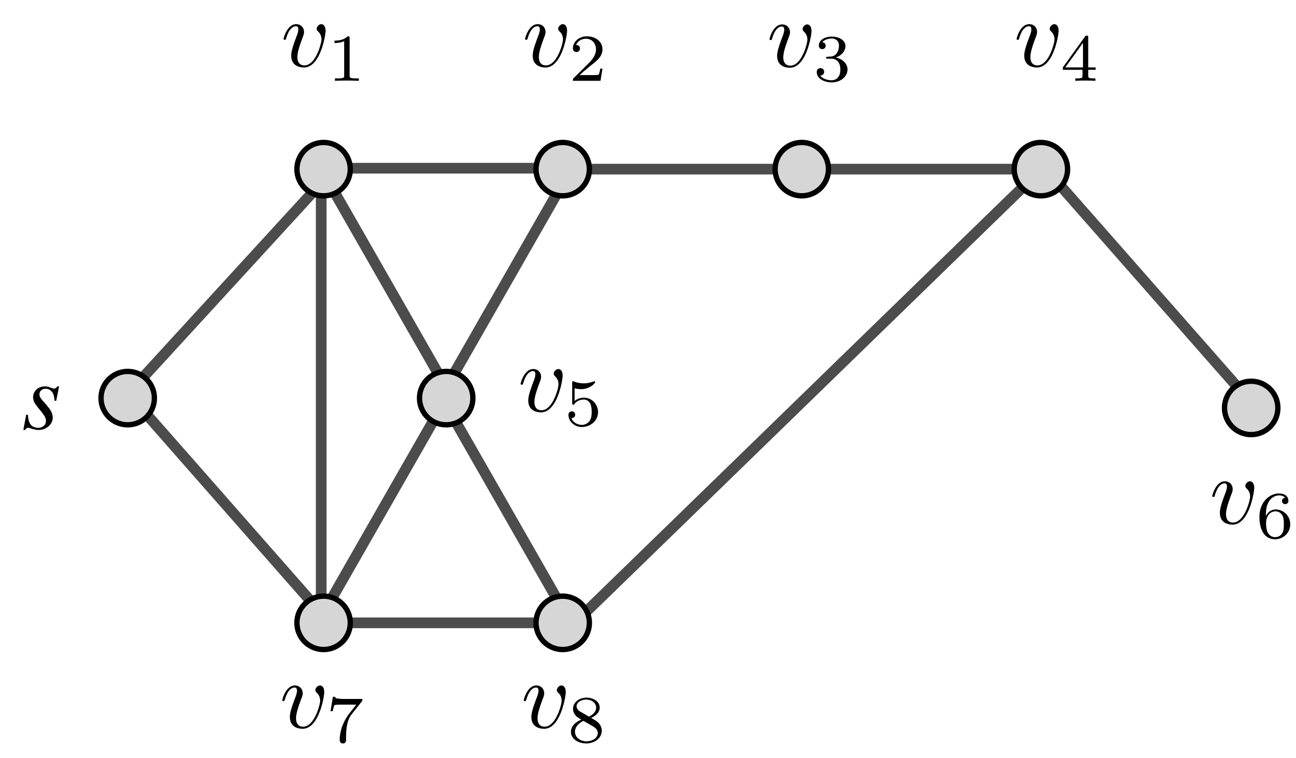


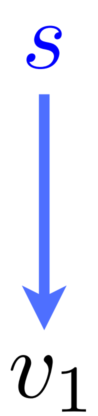
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v ∈ R
   - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

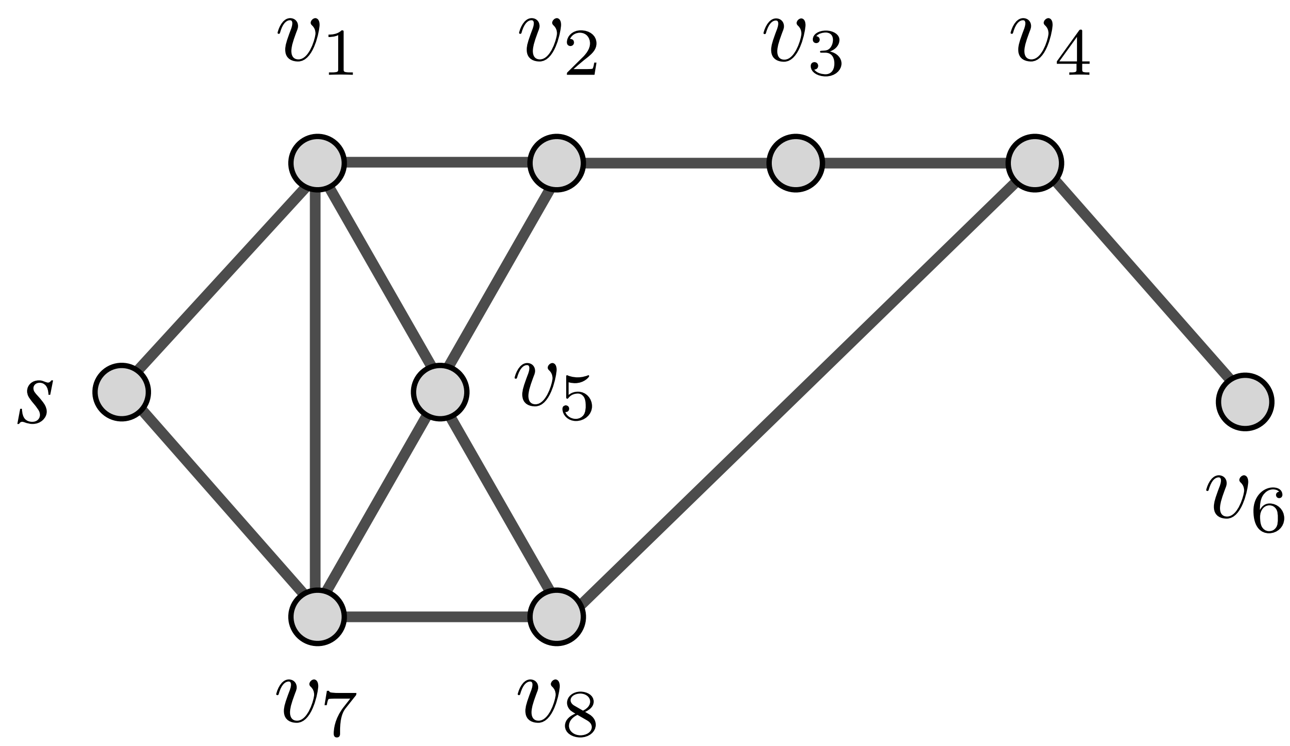
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

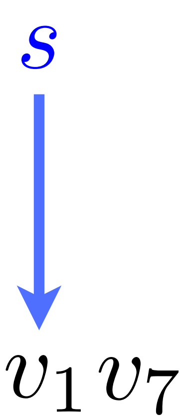
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

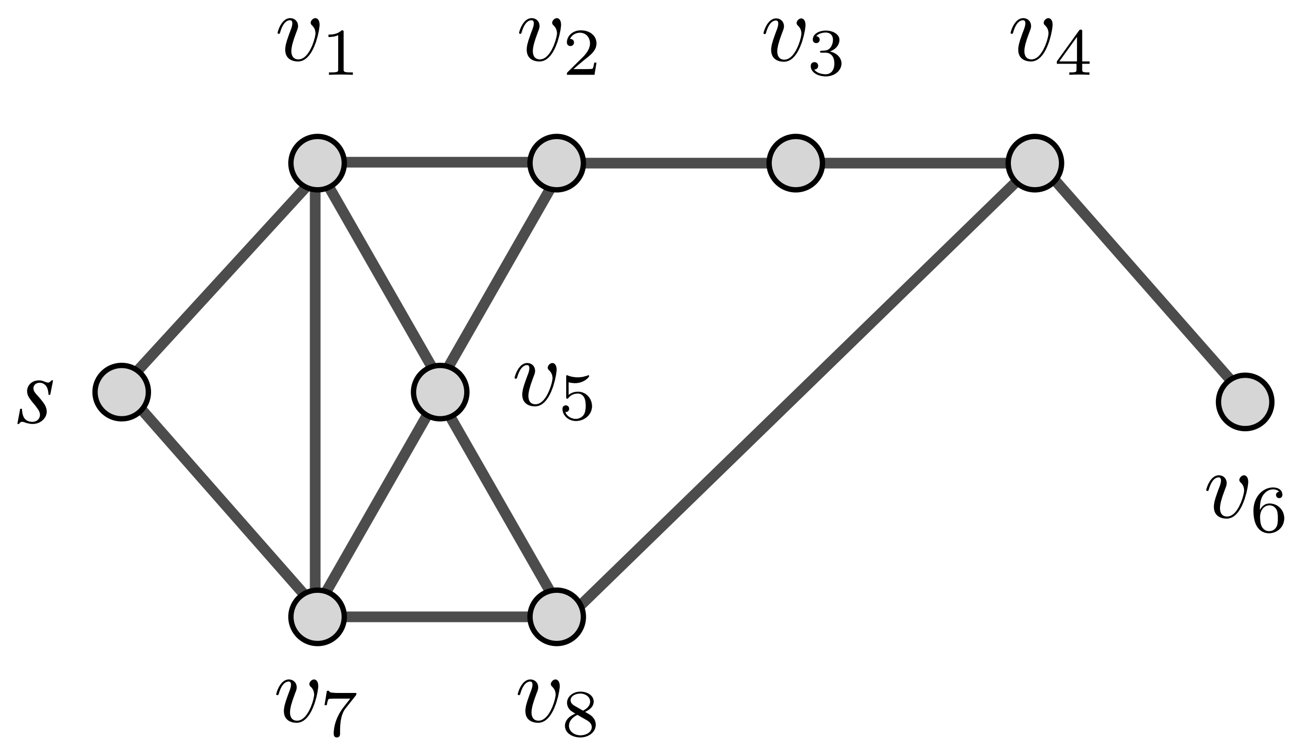
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

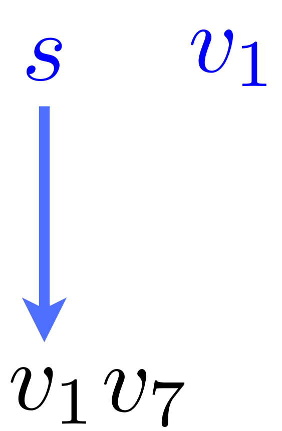
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





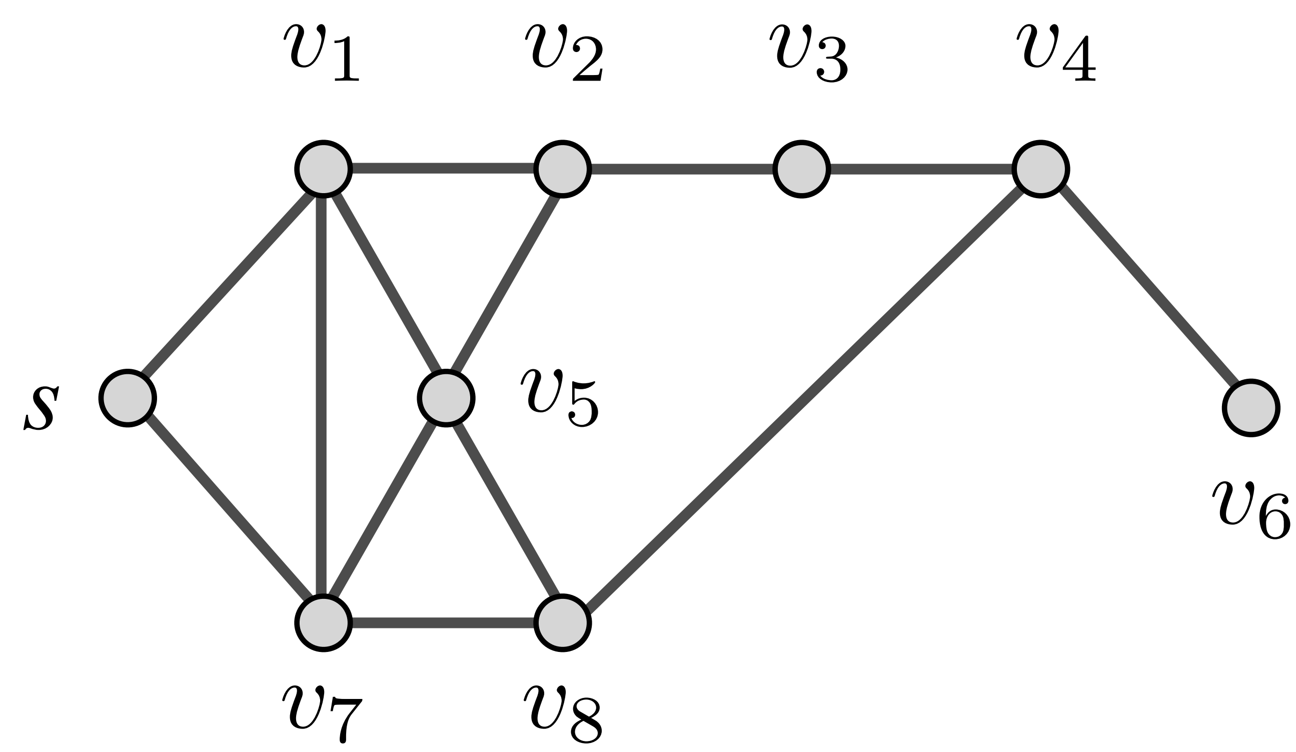
Algorithmus 3.7

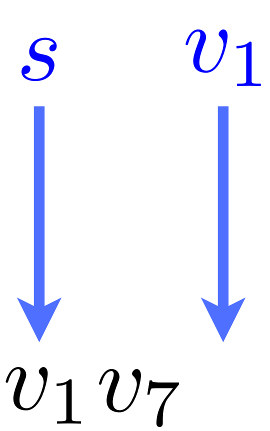
INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    2.1. Wähle $v \in R$
    2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

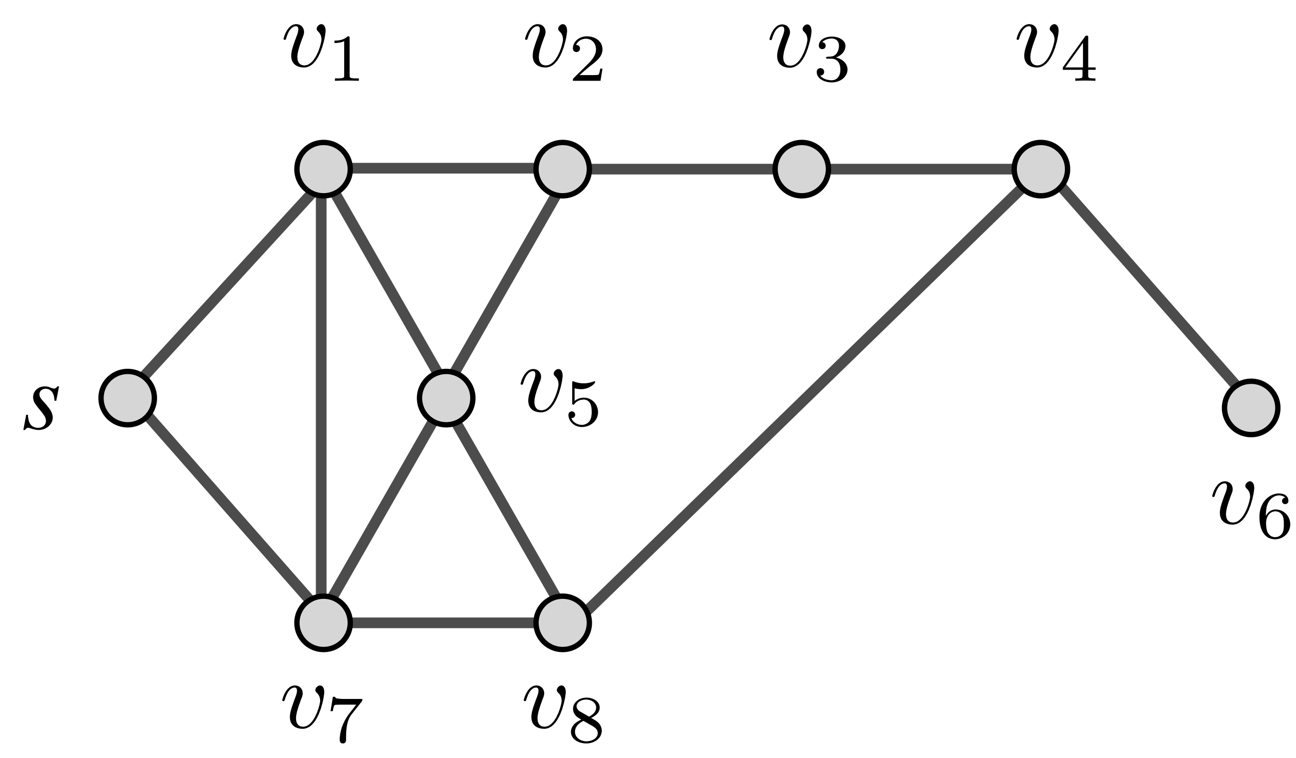
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



$s \qquad v_1$

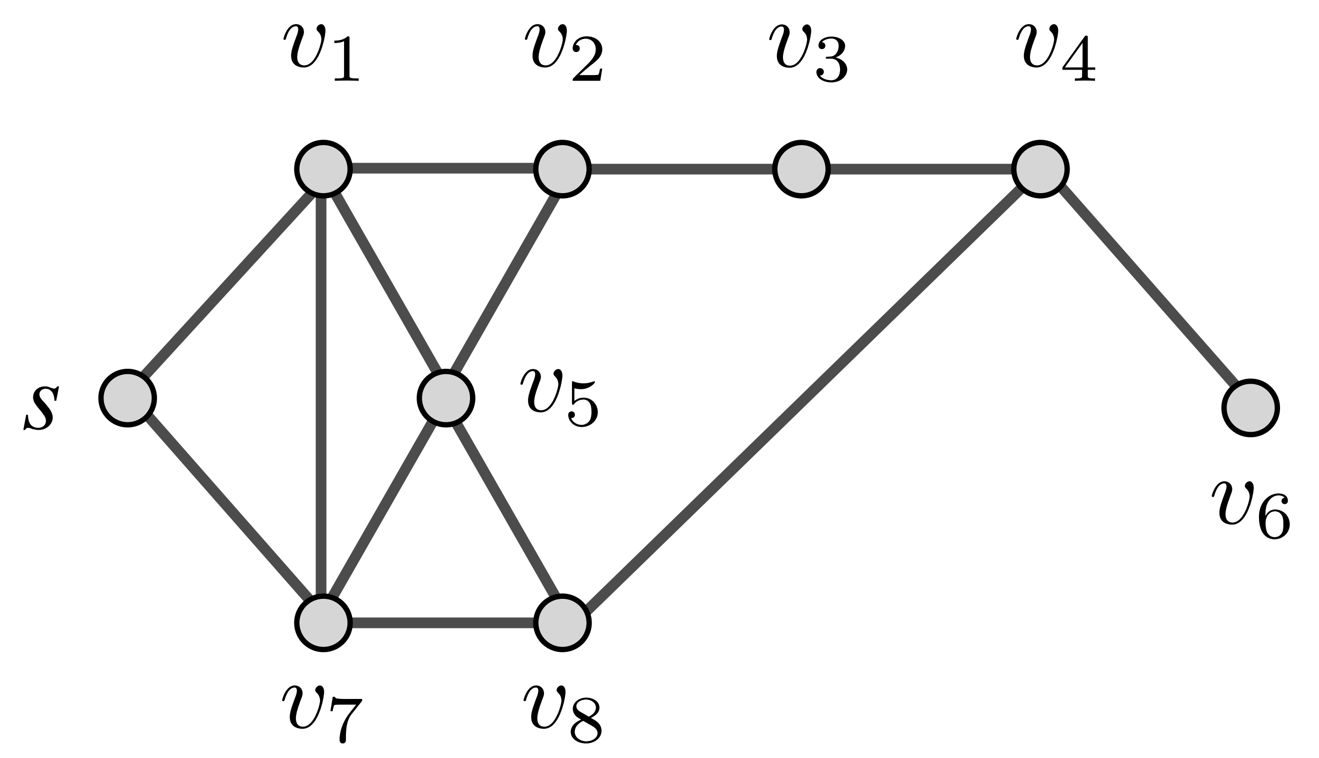
$\downarrow \qquad \downarrow$

$v_1\ v_7\ s$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP



$s \downarrow$ $v_1\ v_7$

$v_1 \downarrow$ $s\ v_7$

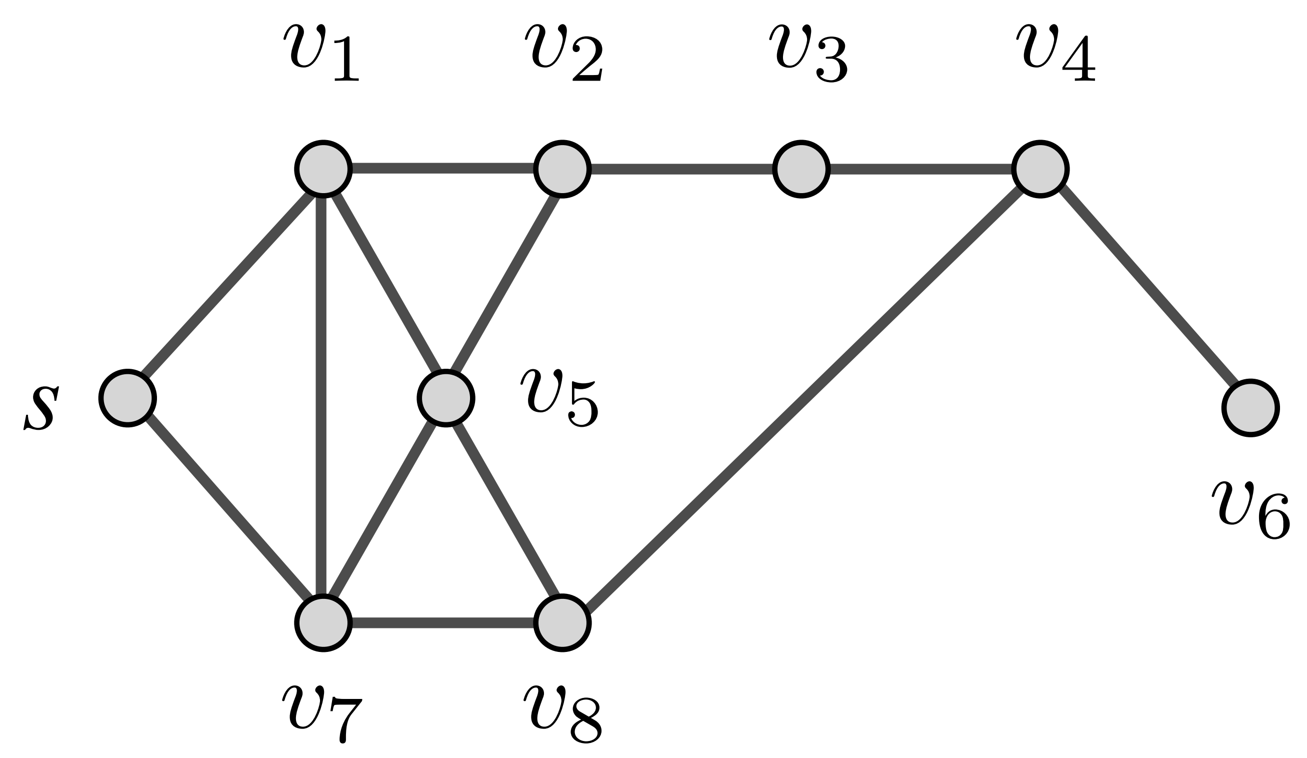
Algorithmus 3.7

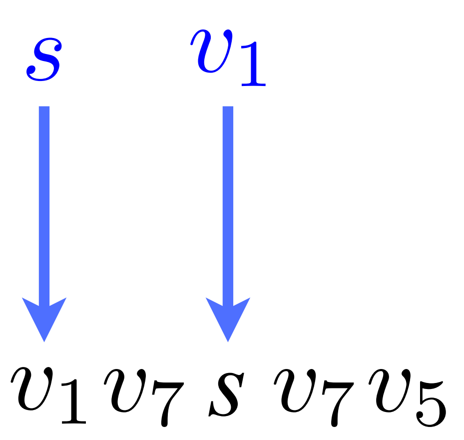
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

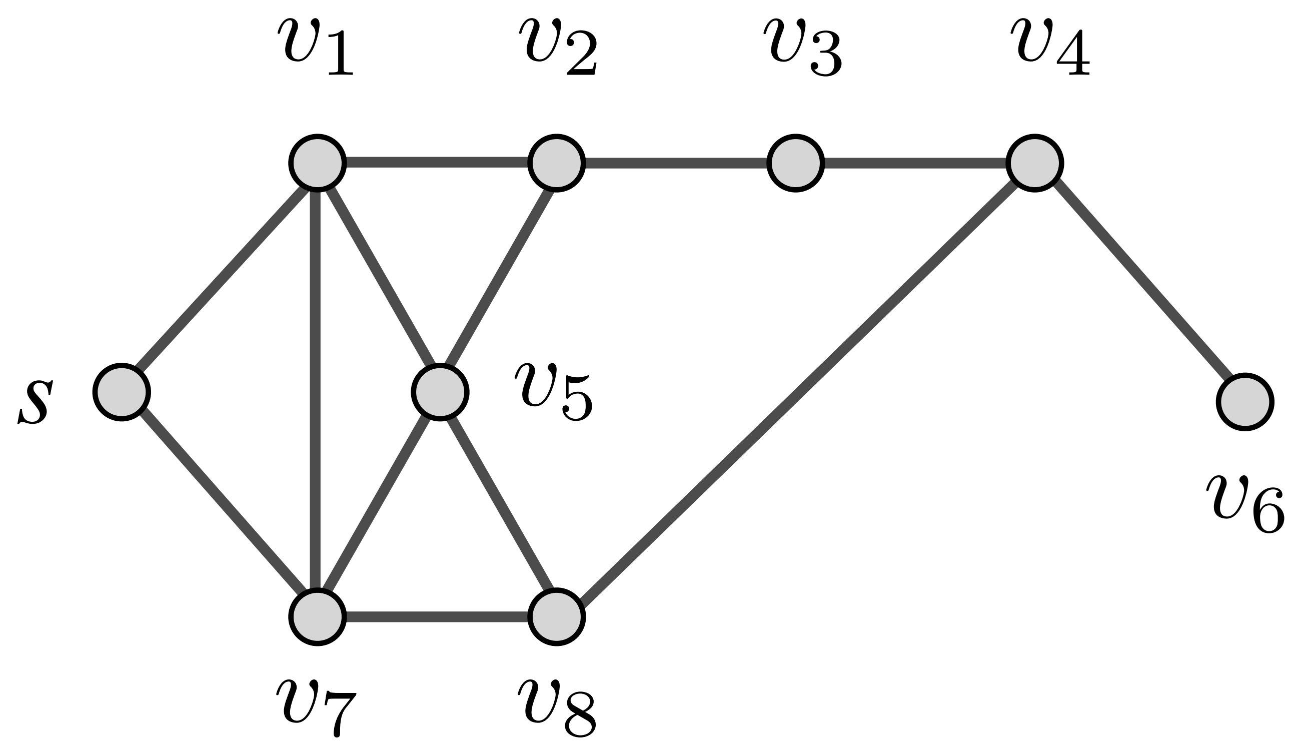
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



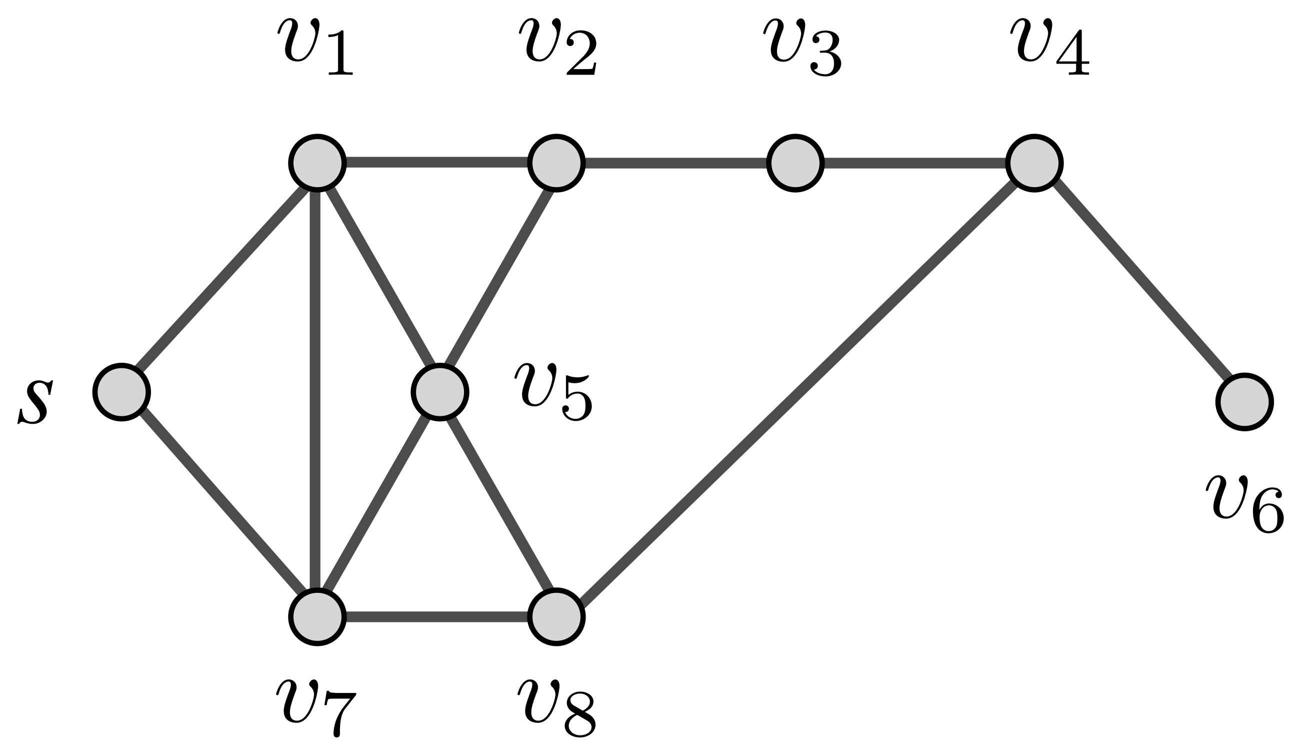
$s \downarrow$ $v_1\, v_7$

$v_1 \downarrow$ $s\, v_7\, v_5\, v_2$

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   2.1. Wähle $v \in R$
   2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
      2.2.1. $R:=R\setminus\{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP



s v_1 v_2

$v_1\ v_7\ s\ v_7\ v_5\ v_2$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

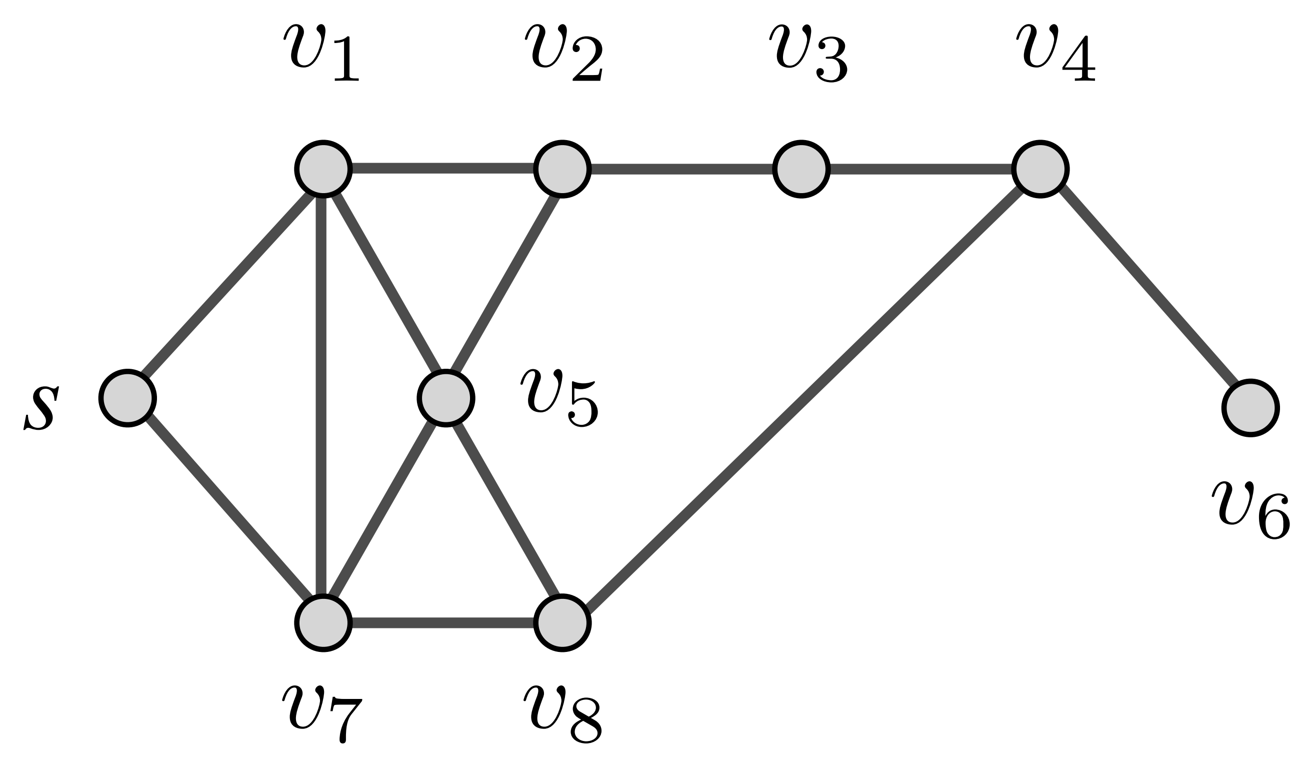
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP



s $\quad v_1 \quad\quad v_2$

$v_1\, v_7\;\; s\;\; v_7\, v_5\, v_2$

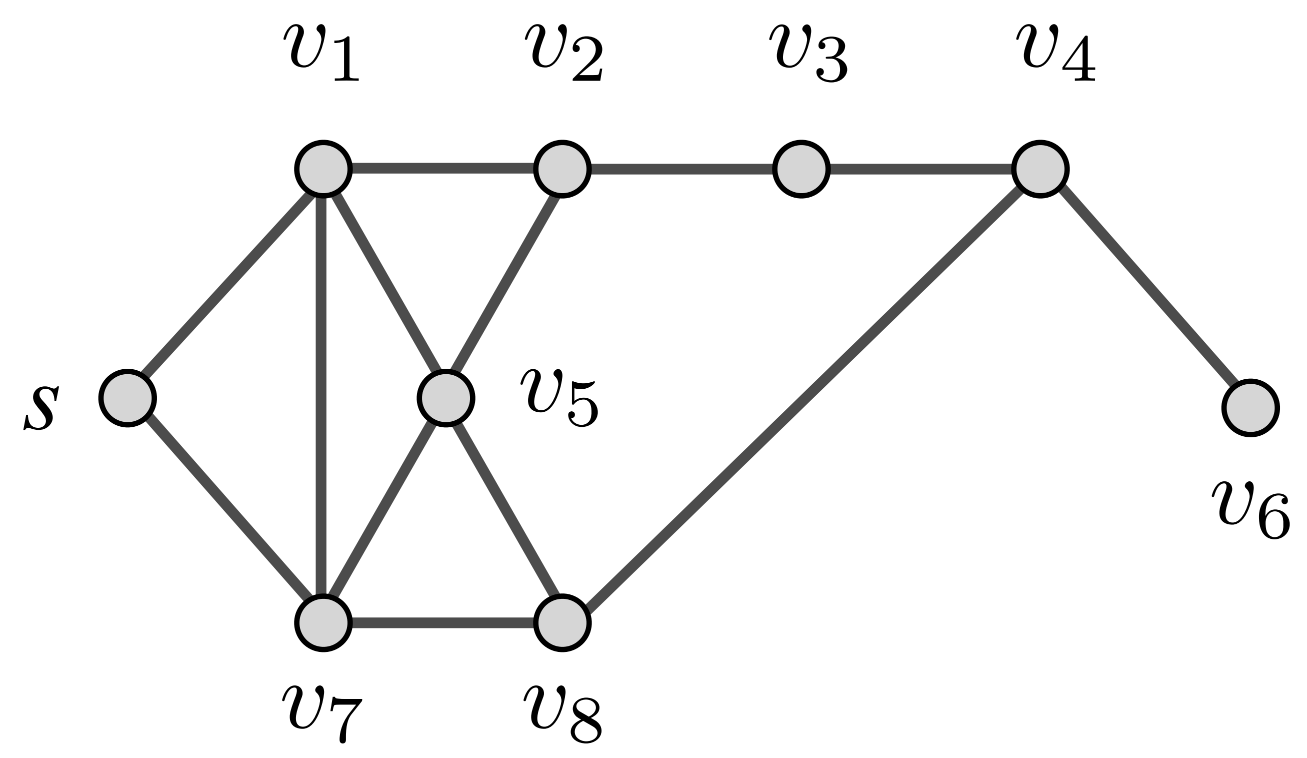
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

   }
3. STOP



s v_1 v_2

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1$

Algorithmus 3.7

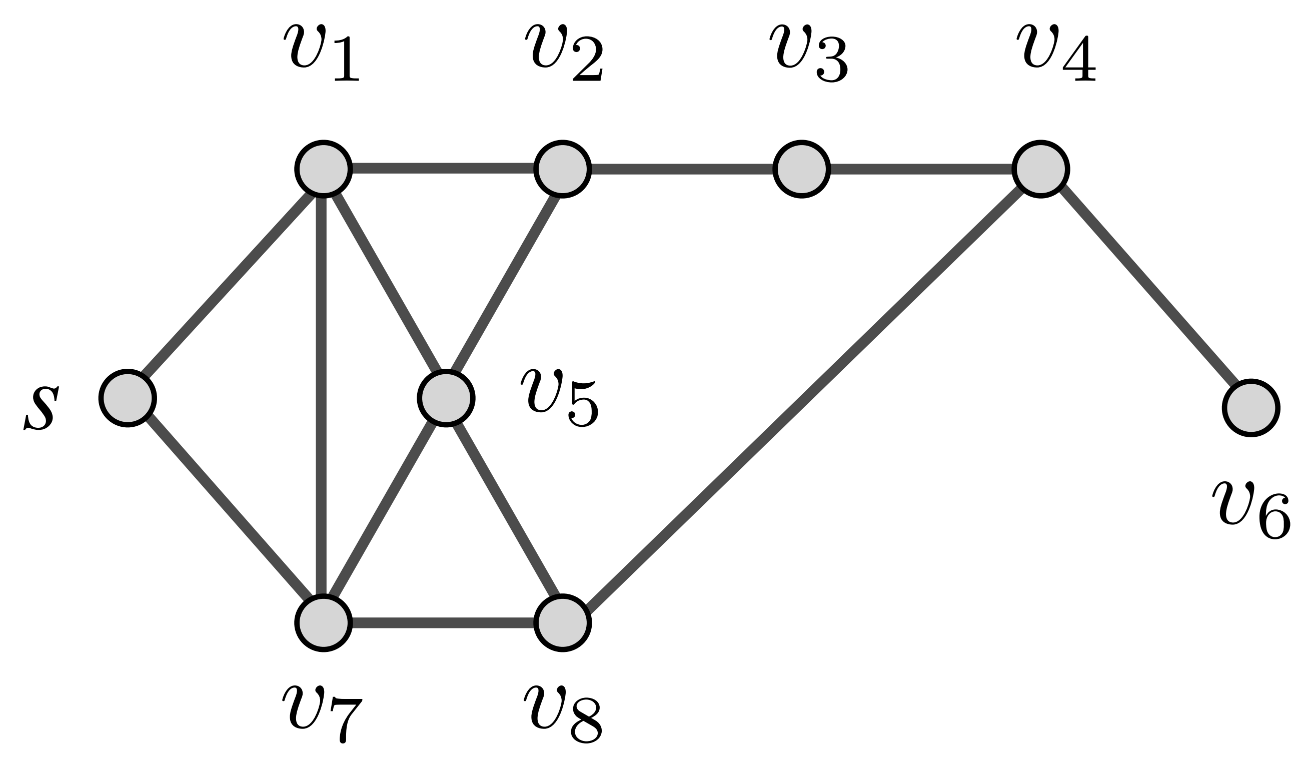
INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP



s v_1 v_2

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {

    2.1. Wähle $v \in R$

    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

        2.2.1. $R:=R\setminus\{v\}$

    2.3. ELSE {

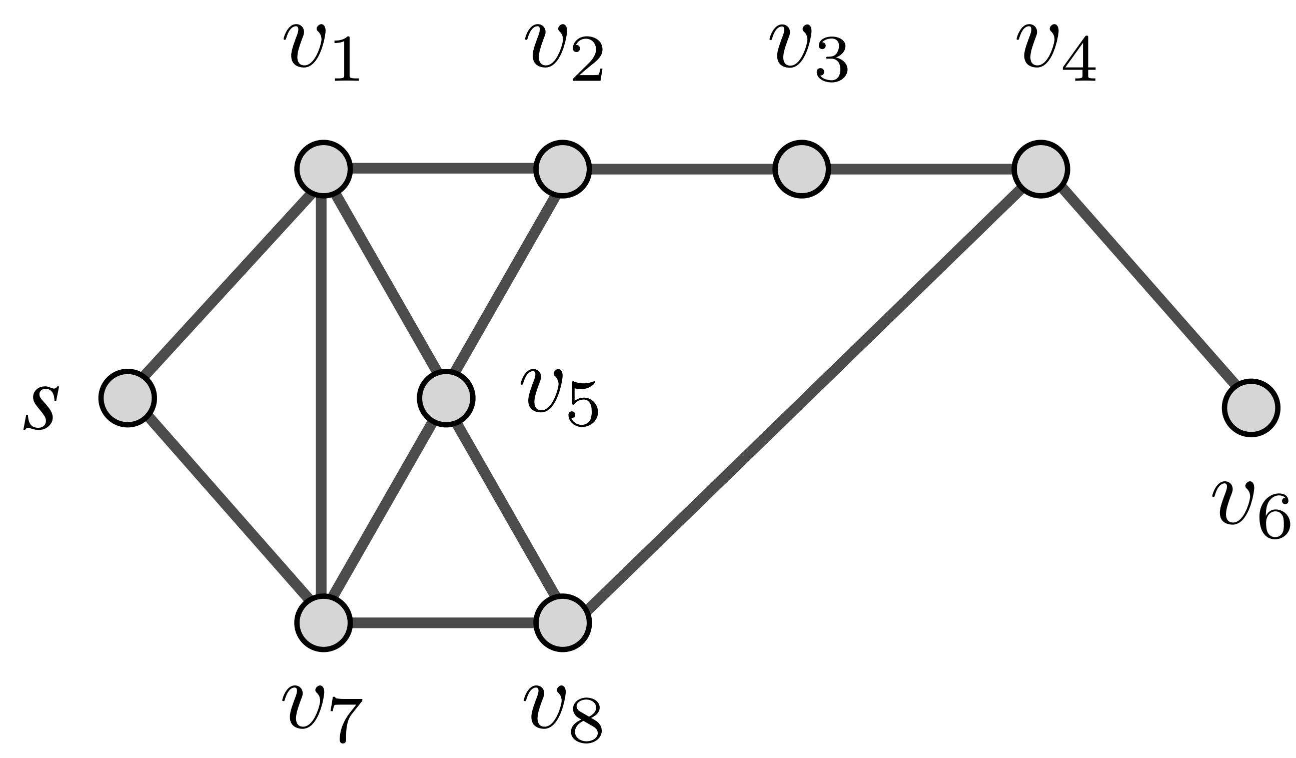
        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$

        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }

3. STOP



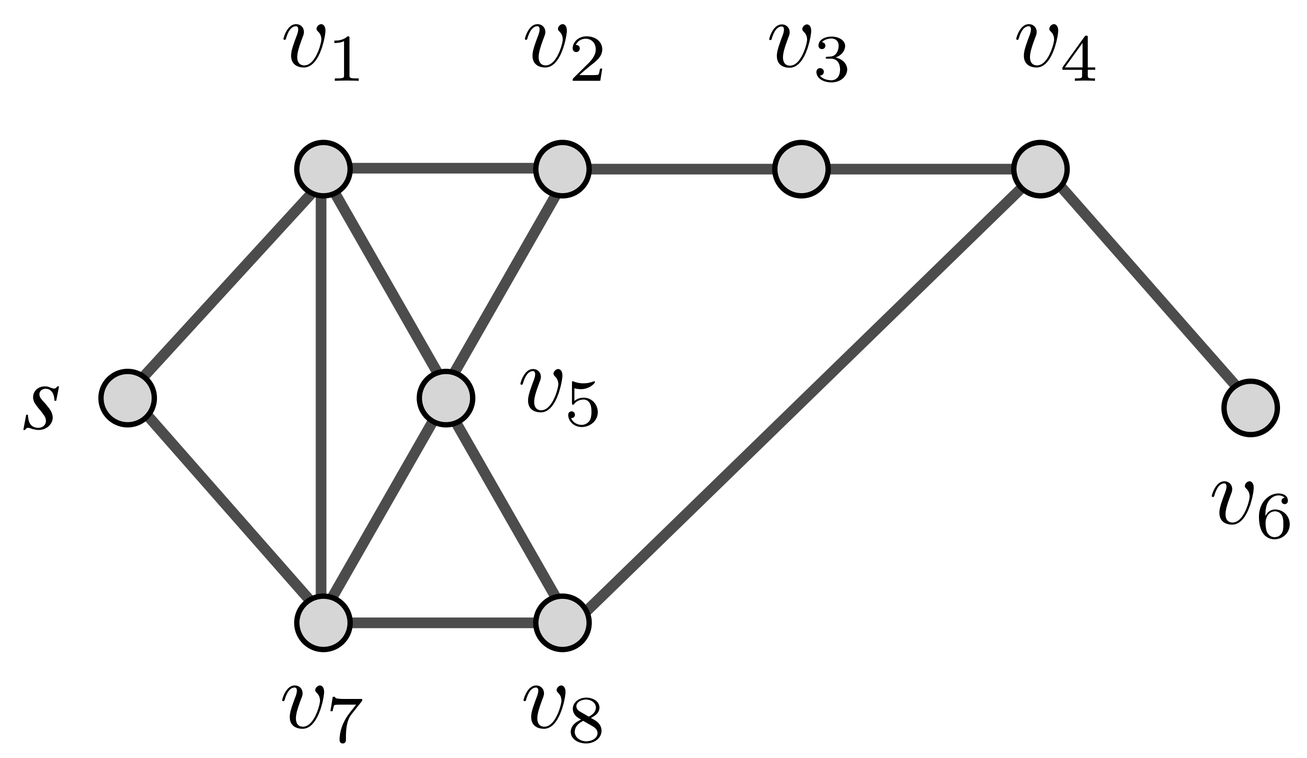
s v_1 v_2

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈ E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3$

$v_1 \; v_7 \; s \; v_7 \; v_5 \; v_2 \; v_1 \; v_5 \; v_3$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

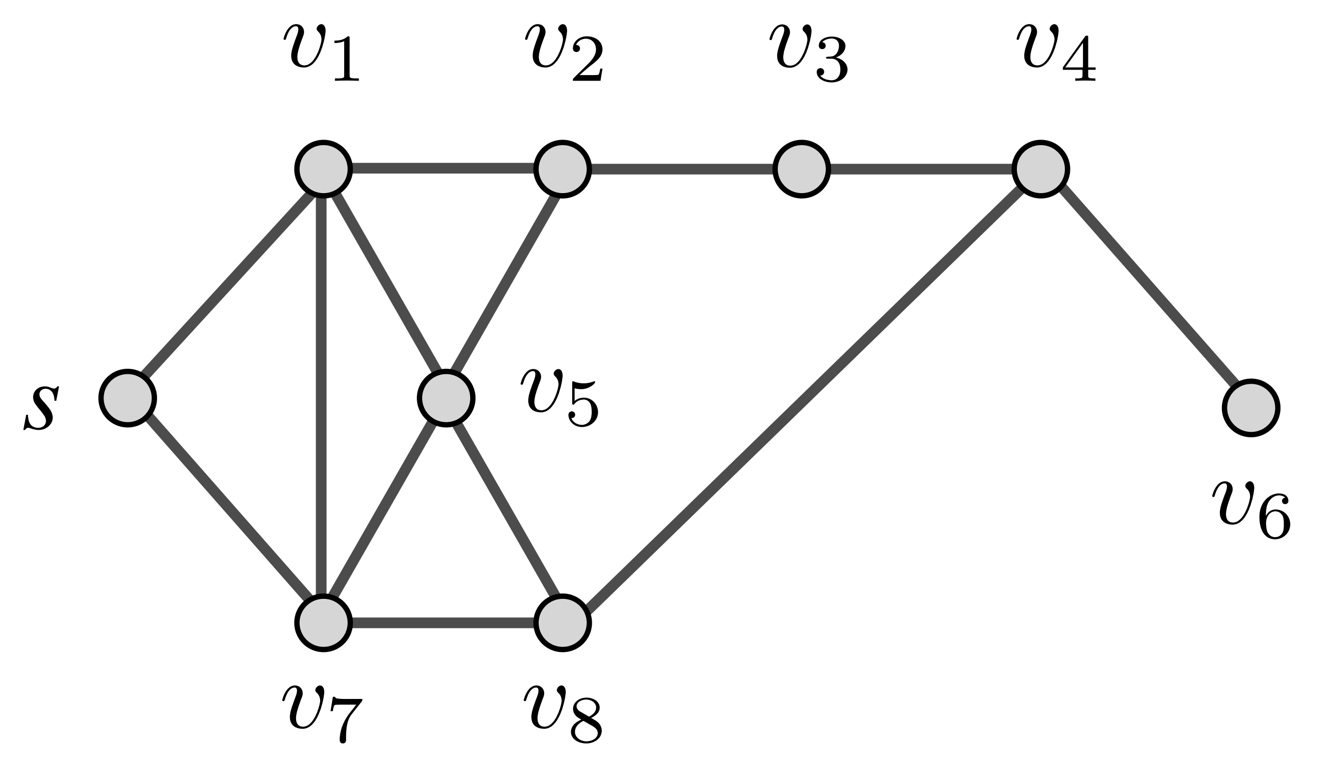
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



s v_1 v_2 v_3

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

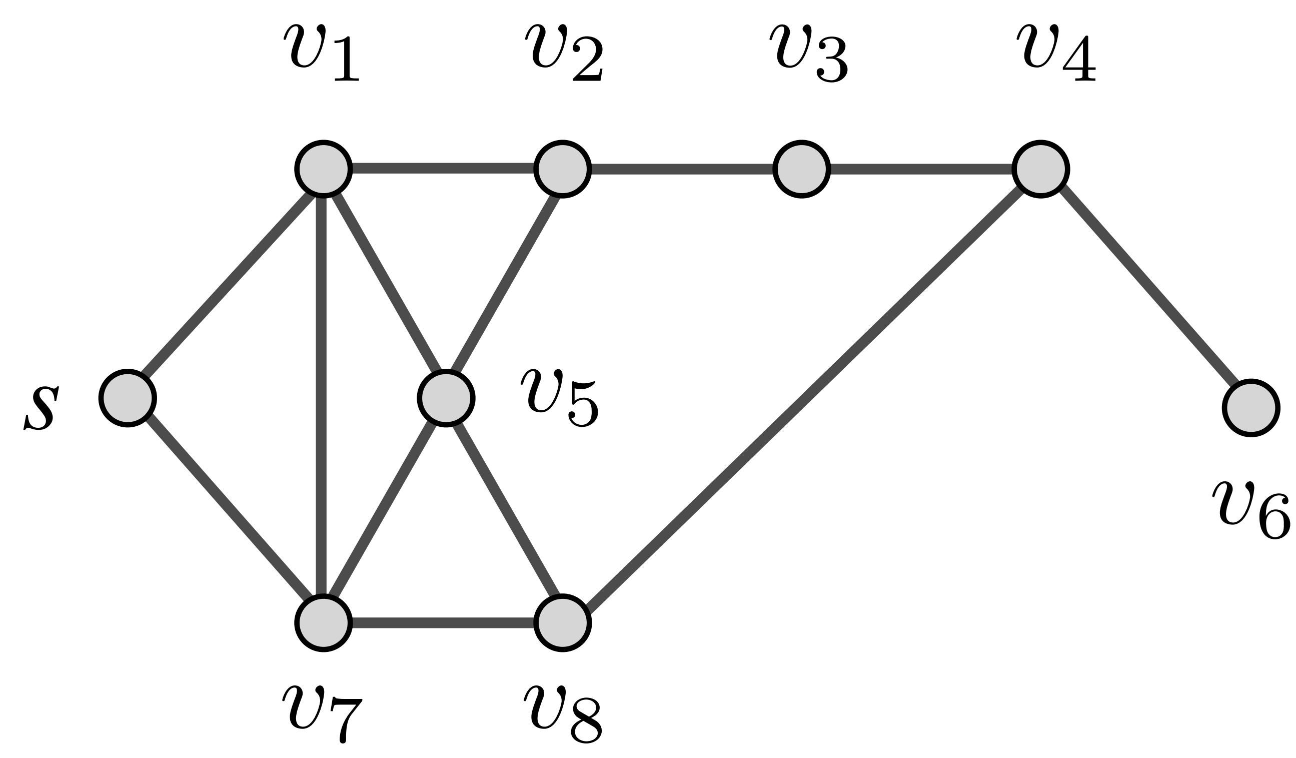
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3$

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2$

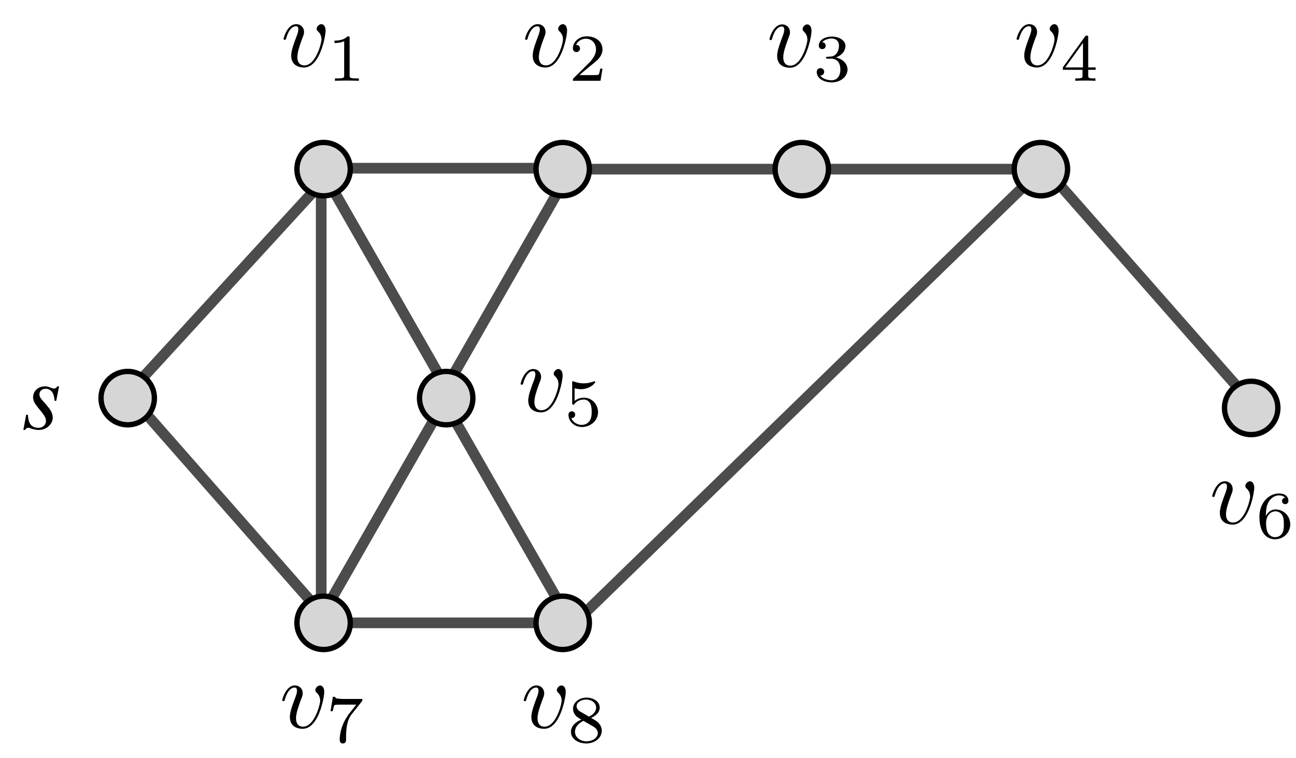
Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        - }
    - }
3. STOP



$s \downarrow$ $\quad v_1 \downarrow$ $\quad v_2 \downarrow$ $\quad v_3 \downarrow$

$v_1 \; v_7 \; s \; v_7 \; v_5 \; v_2 \; v_1 \; v_5 \; v_3 \; v_2 \; v_4$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

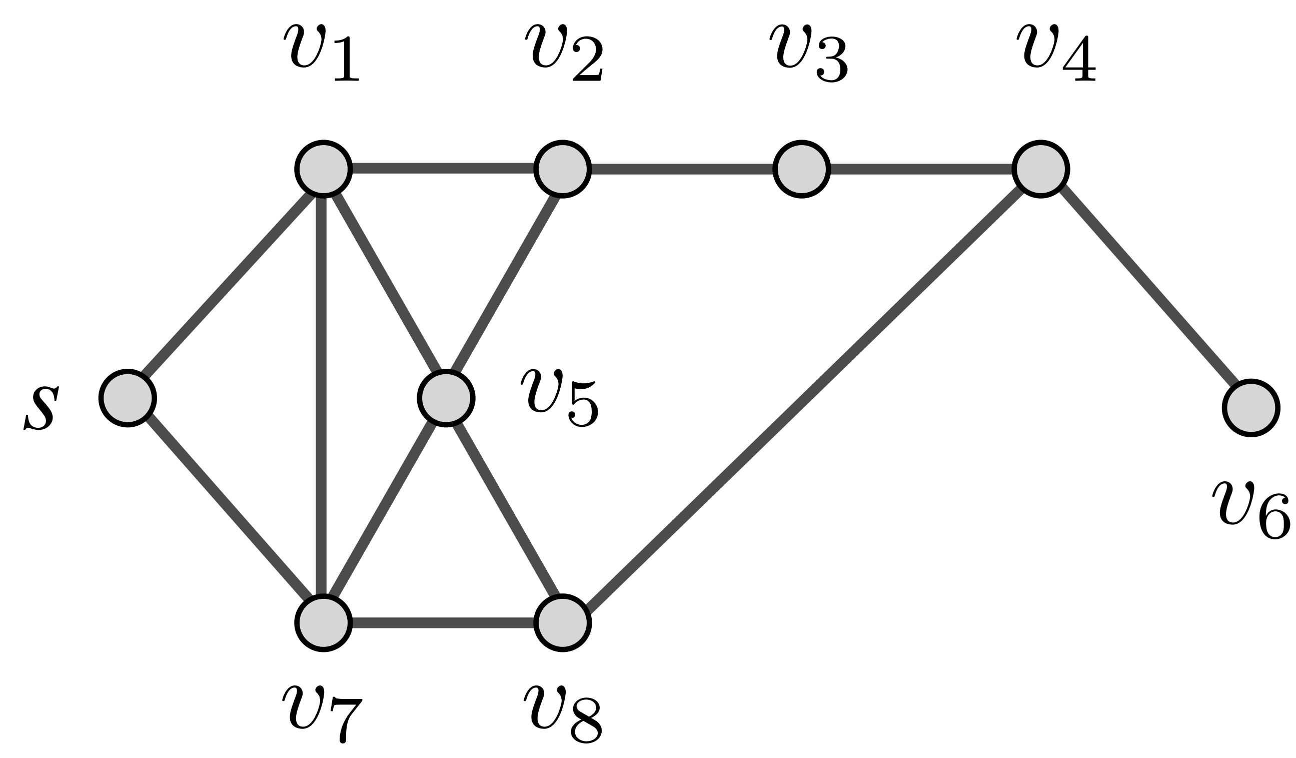
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

       }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4$

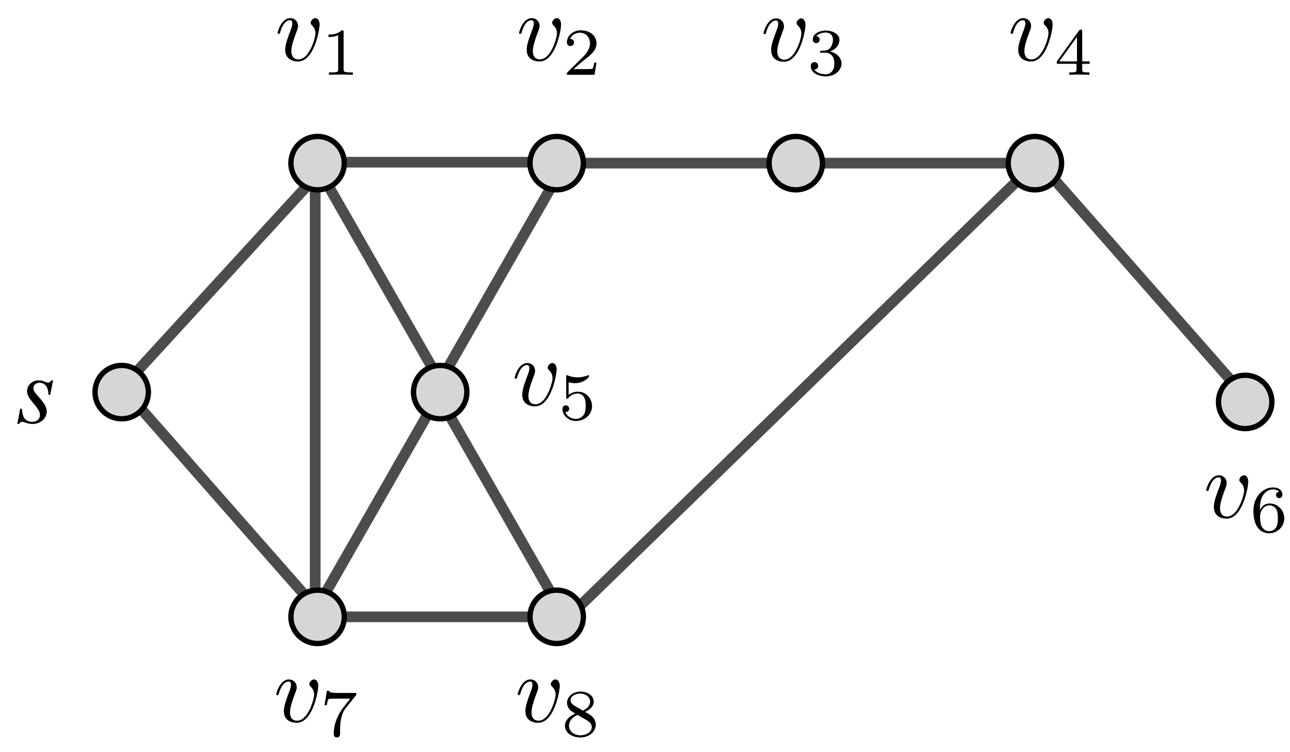
$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4$

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }

   }
3. STOP



s v_1 v_2 v_3 v_4

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

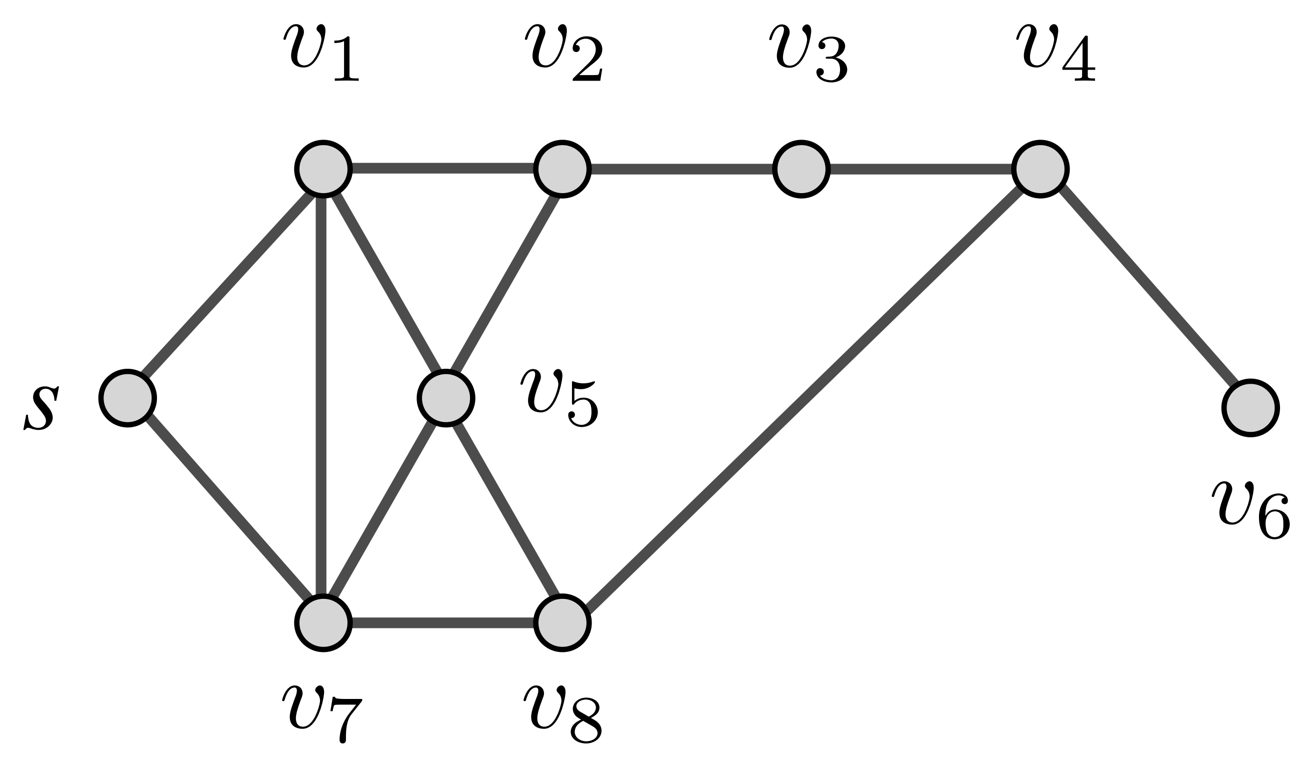
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3$

Algorithmus 3.7

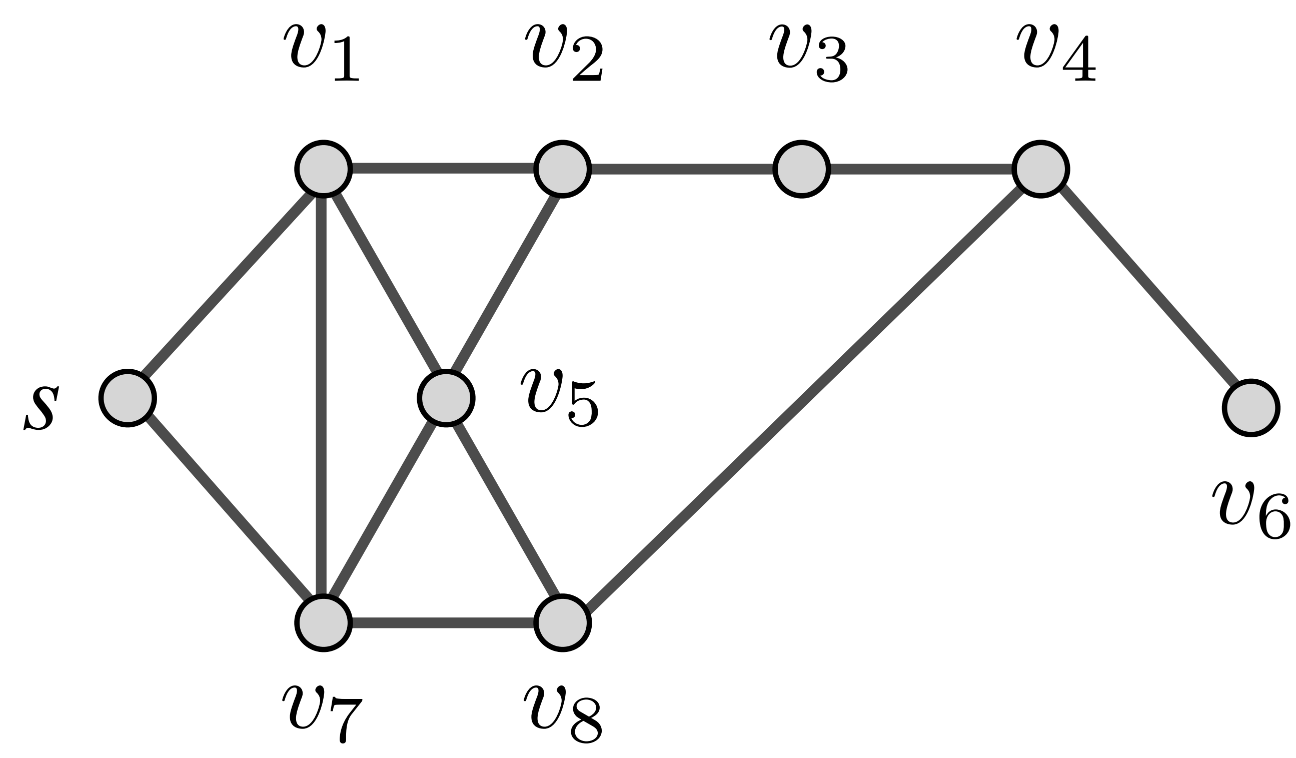
INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist, Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



s v_1 v_2 v_3 v_4

v_1 v_7 s v_7 v_5 v_2 v_1 v_5 v_3 v_2 v_4 v_3 v_8

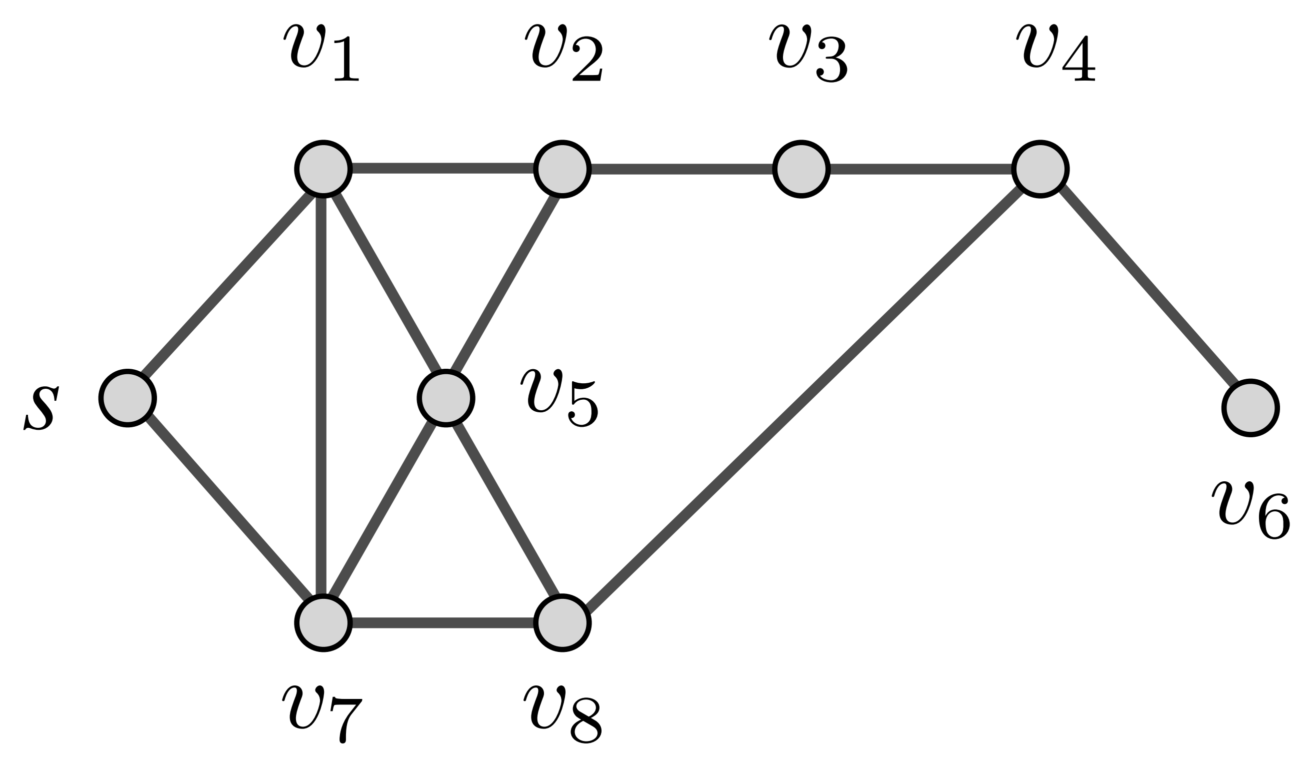
Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈ R
   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
      2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
      }
   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4$

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

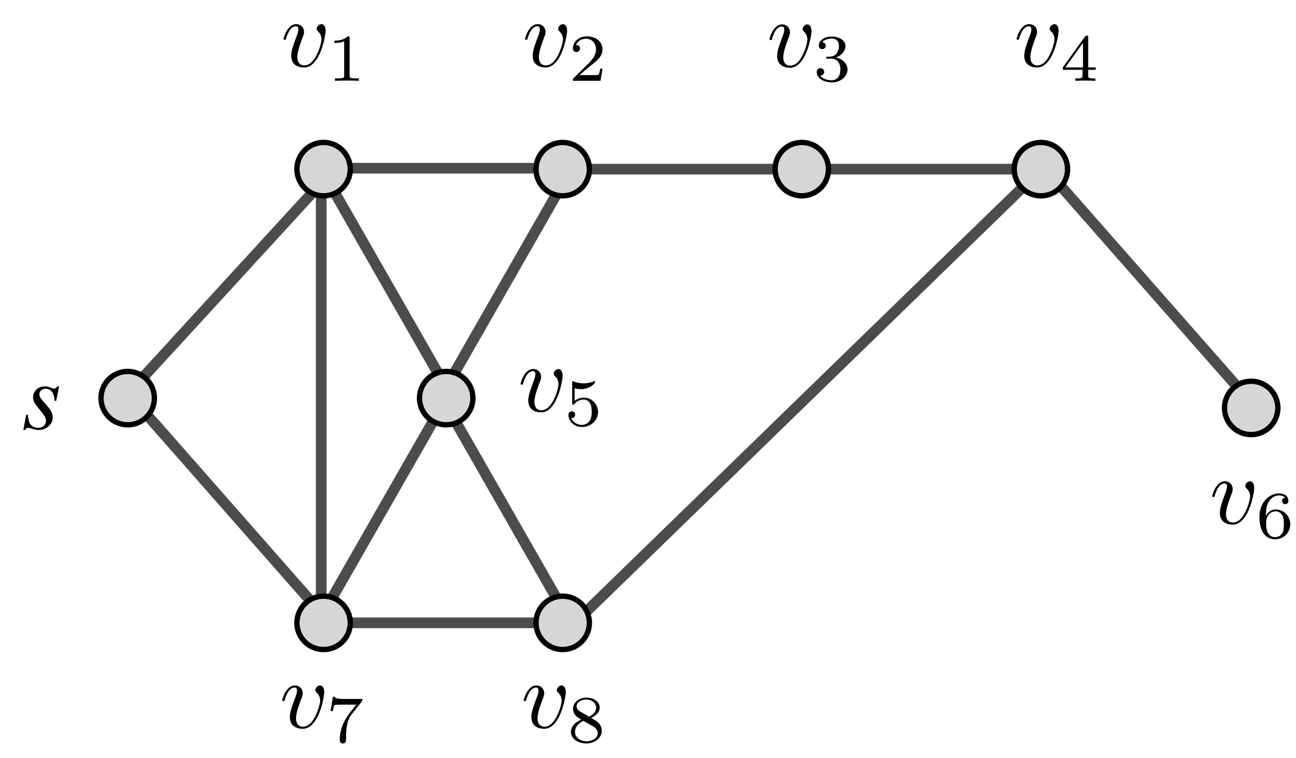
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6$

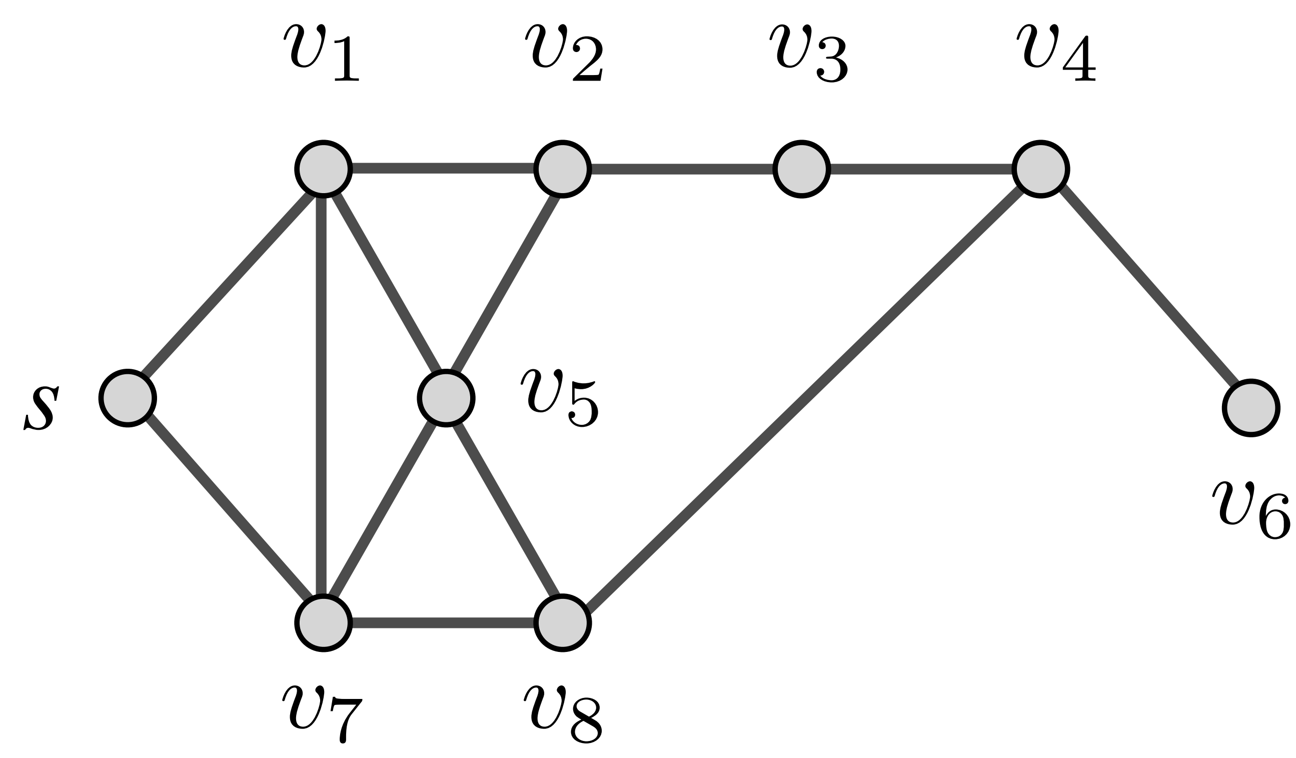
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

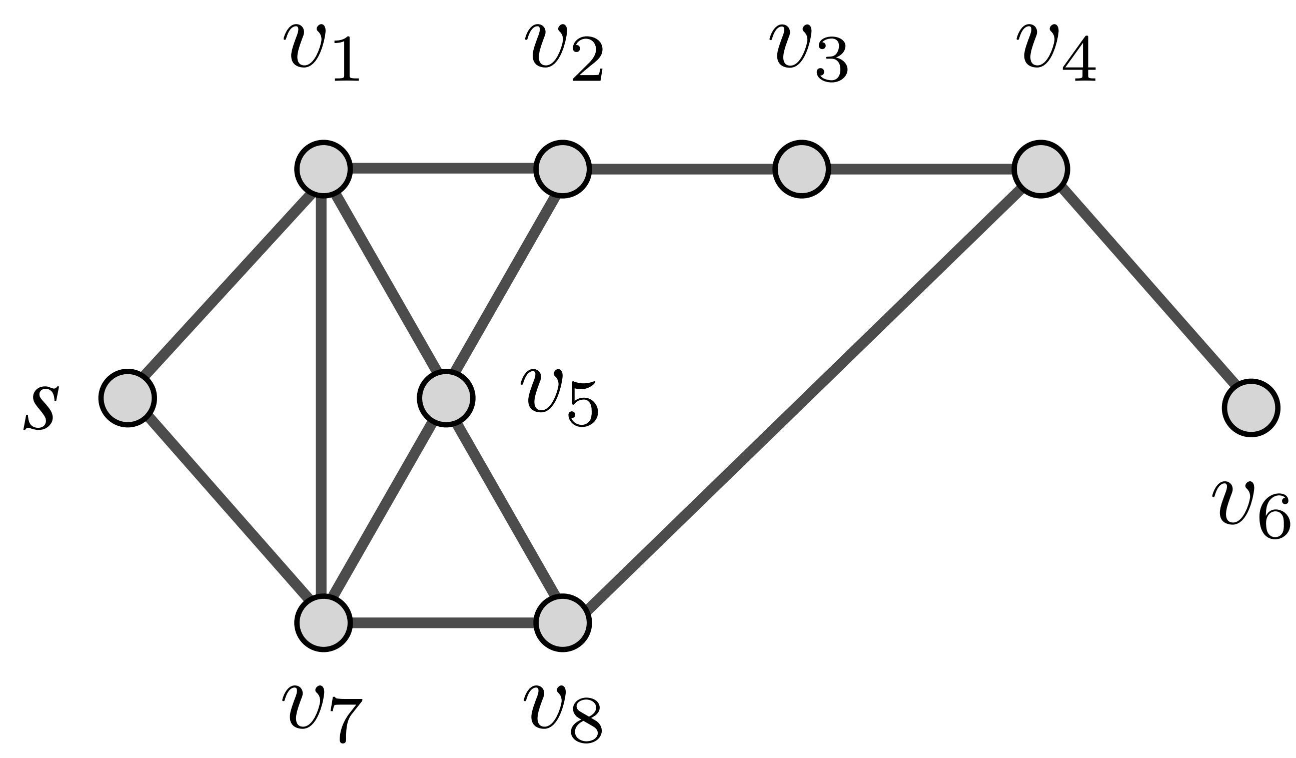
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

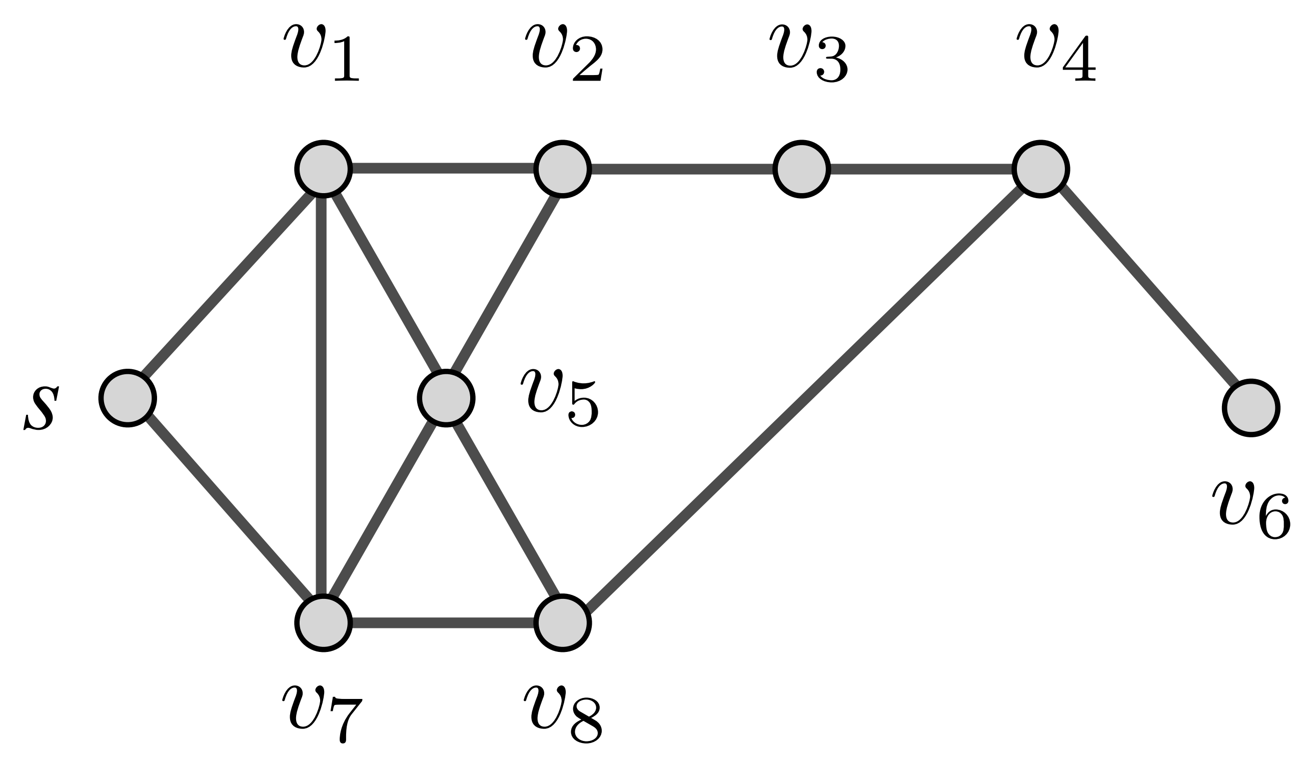
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2$

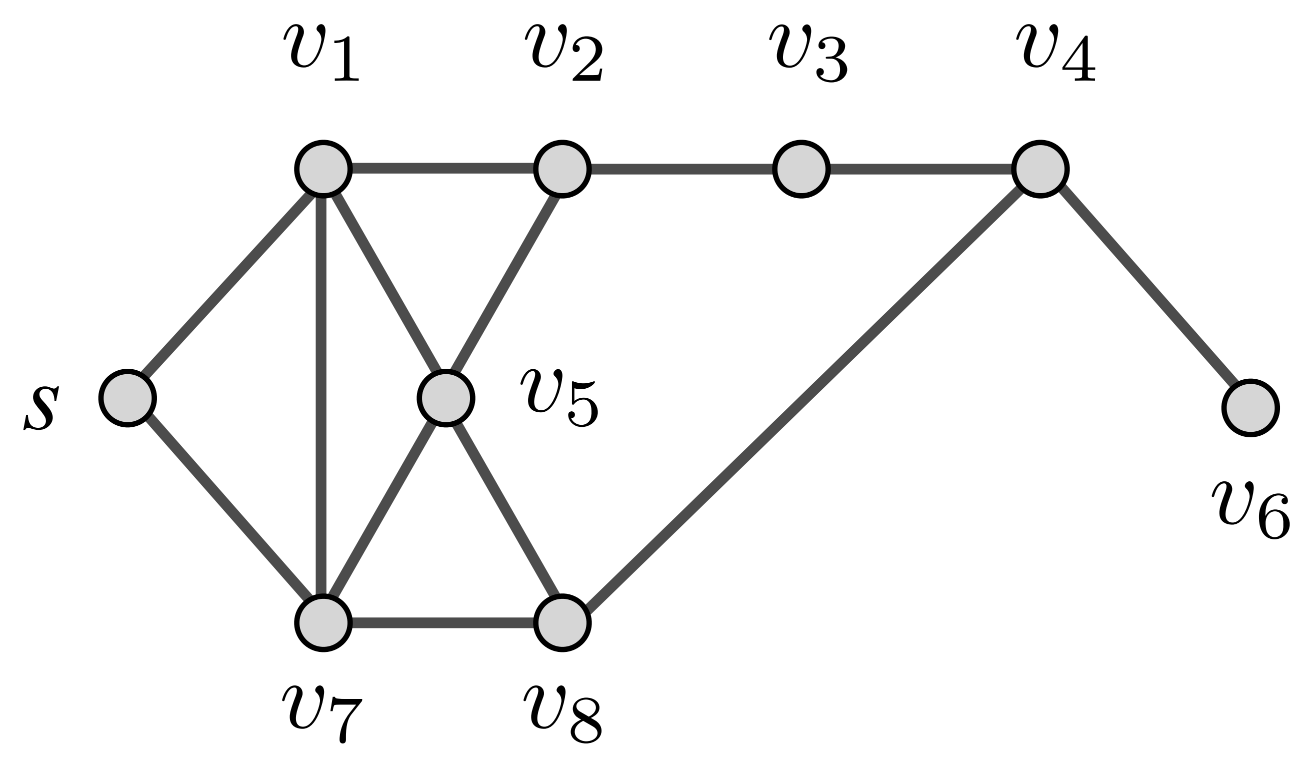
Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    2.1. Wähle $v \in R$
    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        2.2.1. $R:=R \setminus \{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7$

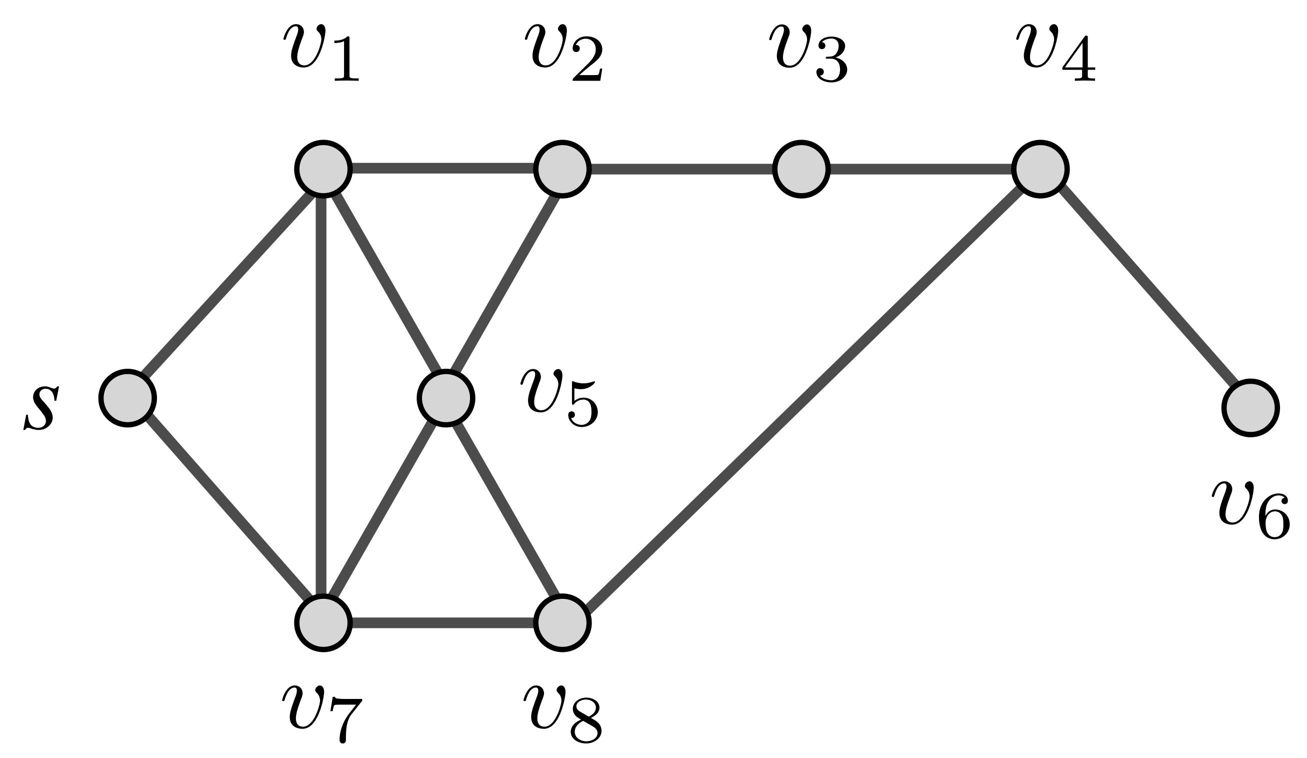
Algorithmus 3.7

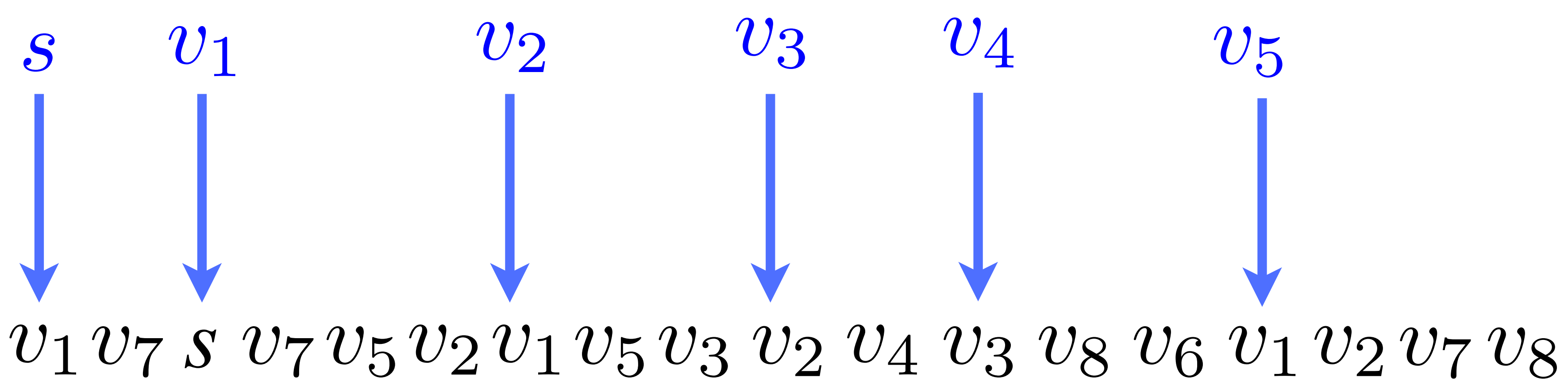
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R\neq\emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

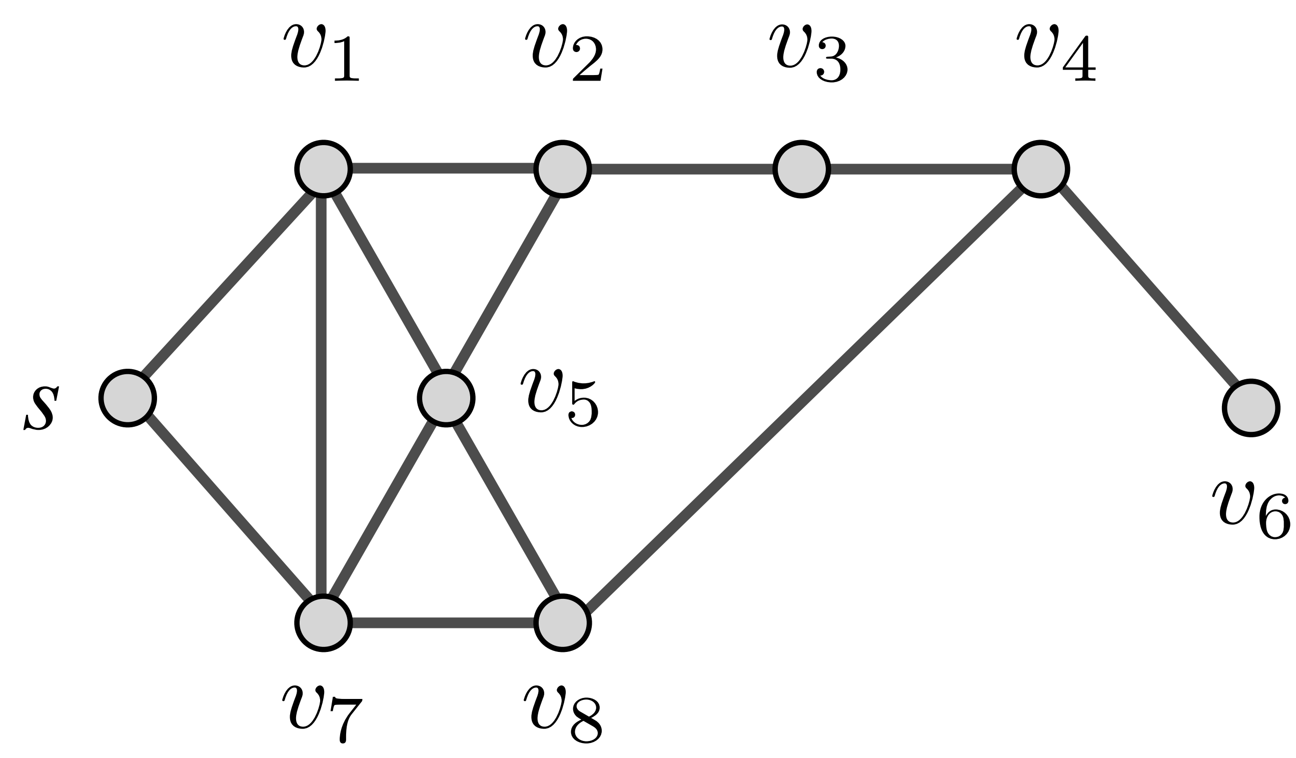
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP



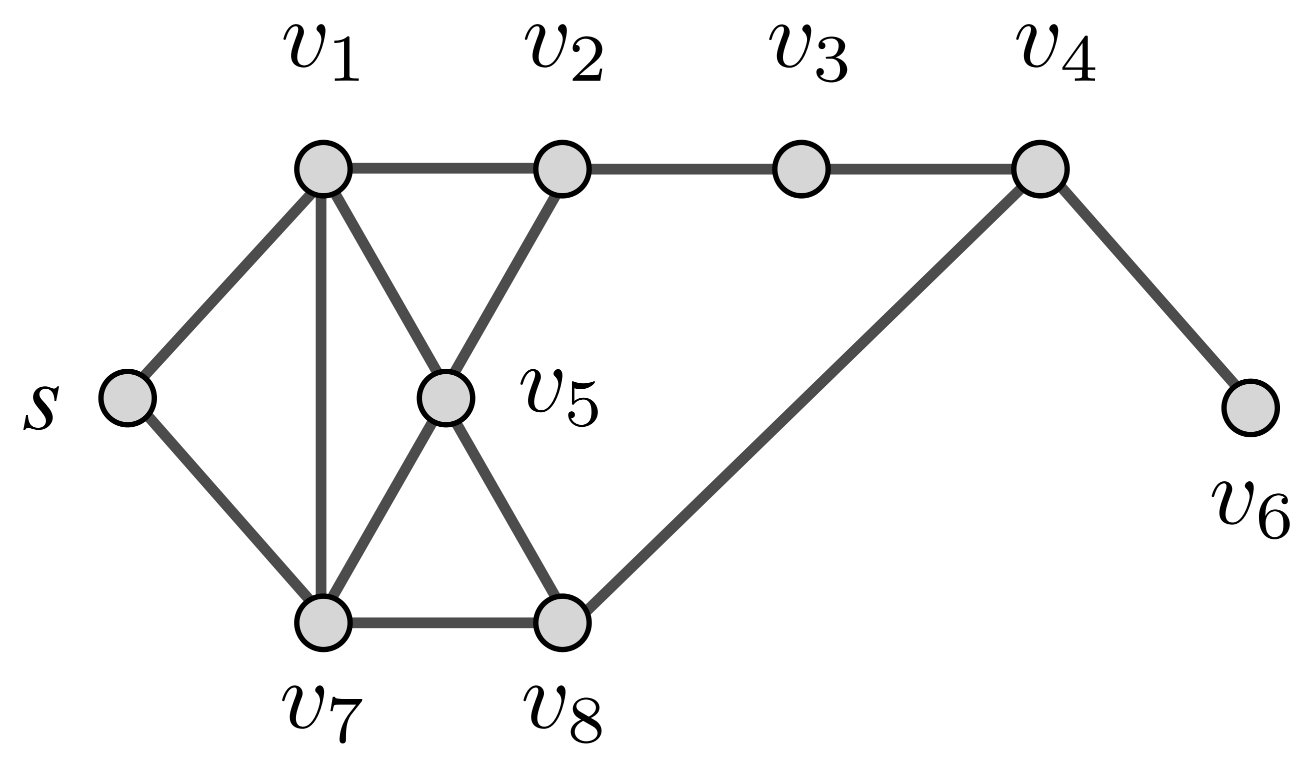
$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6$

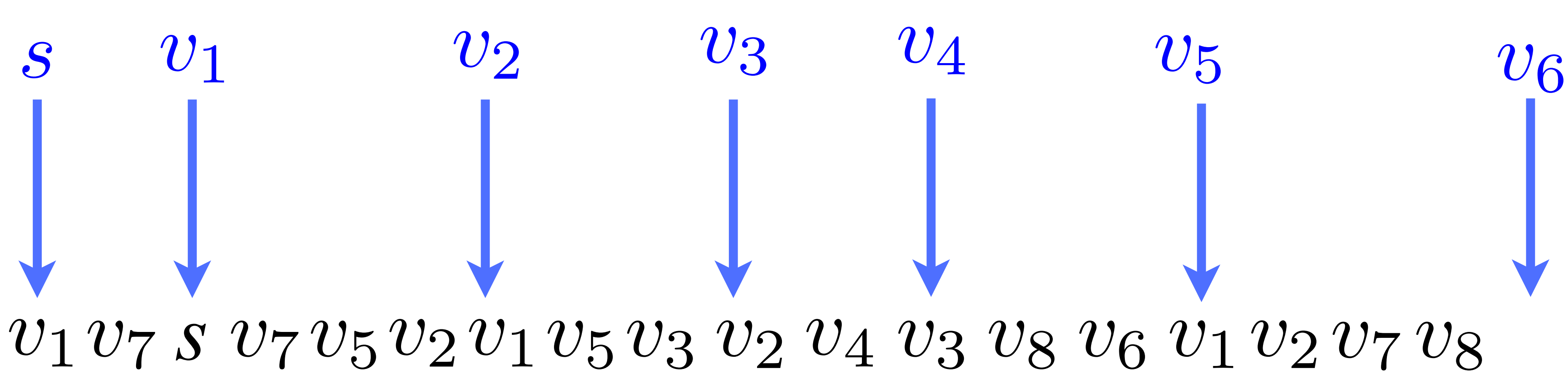
$v_1\, v_7\, s\;\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\;\, v_2\;\, v_4\;\, v_3\;\, v_8\;\, v_6\;\, v_1\, v_2\, v_7\, v_8$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈R
   2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
      2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
      }
   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

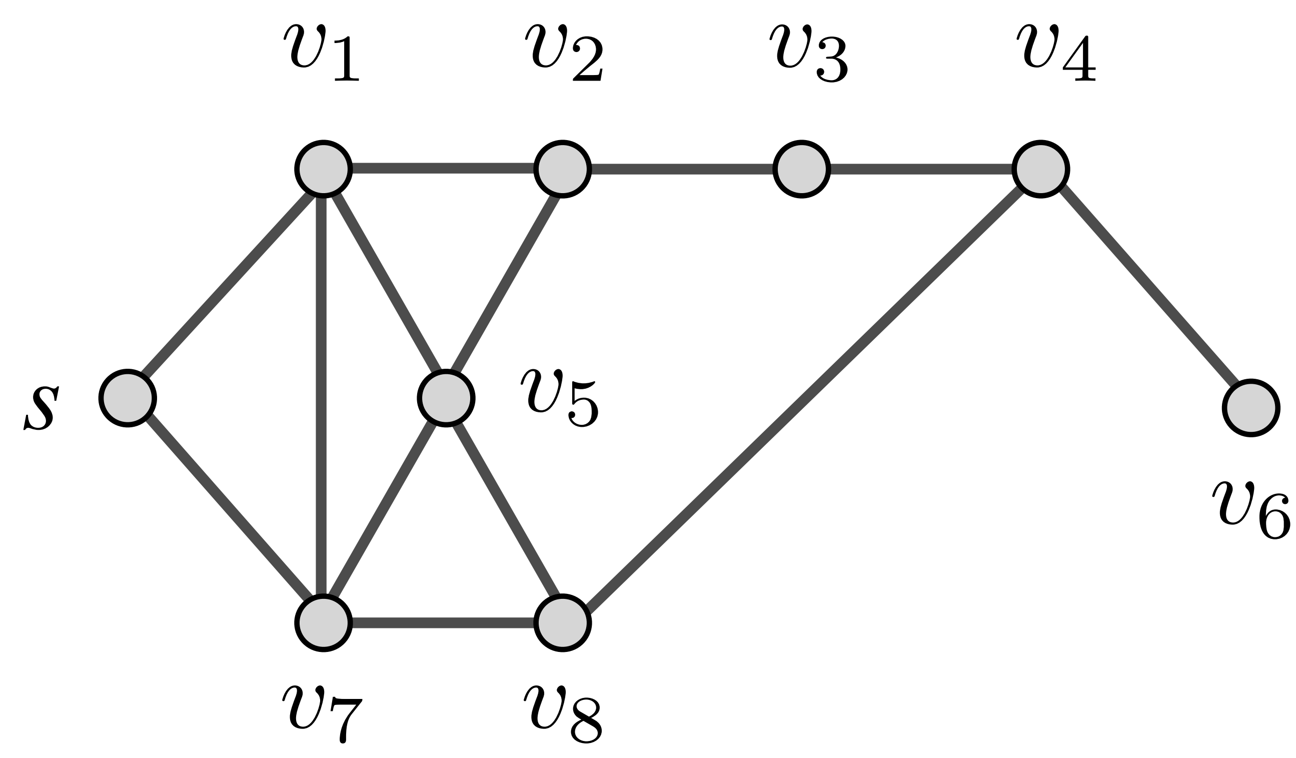
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2



s $\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4$

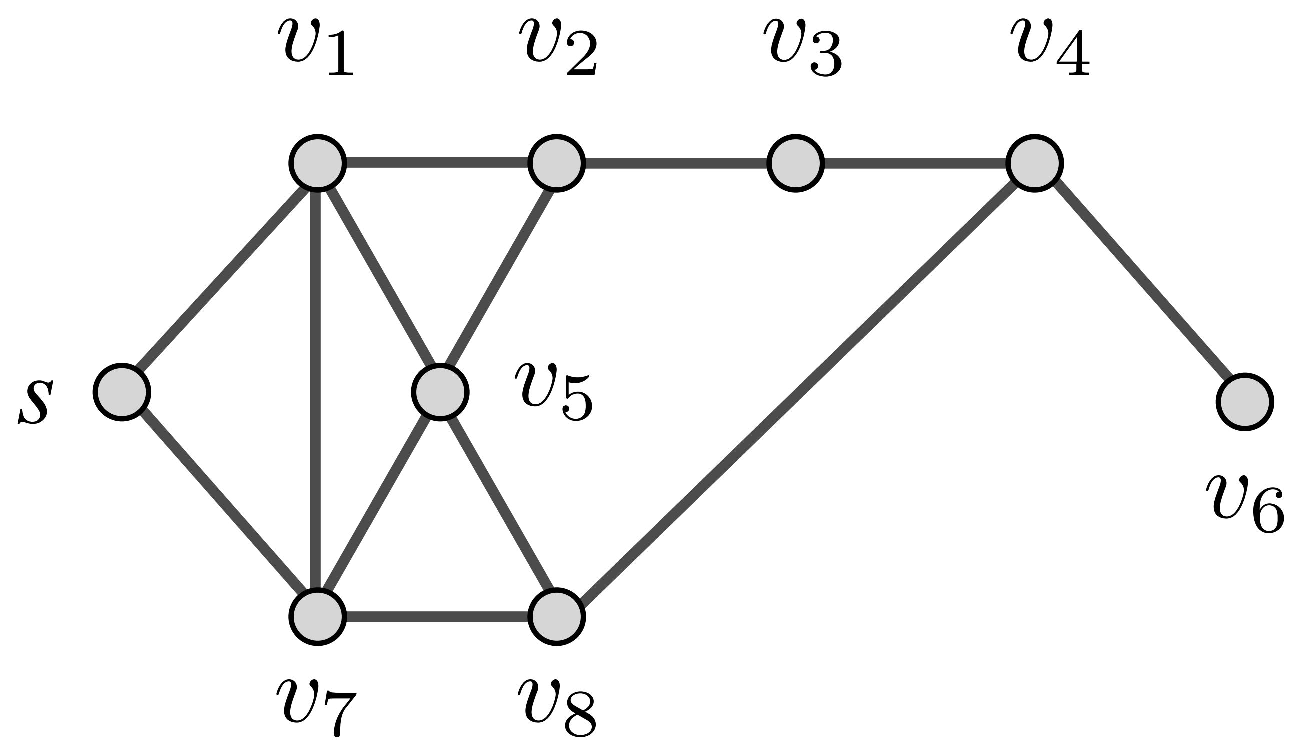
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



s v_1 v_2 v_3 v_4 v_5 $v_6\,v_7$

$v_1\,v_7\,s\,v_7\,v_5\,v_2\,v_1\,v_5\,v_3\,v_2\,v_4\,v_3\,v_8\,v_6\,v_1\,v_2\,v_7\,v_8\,v_4$

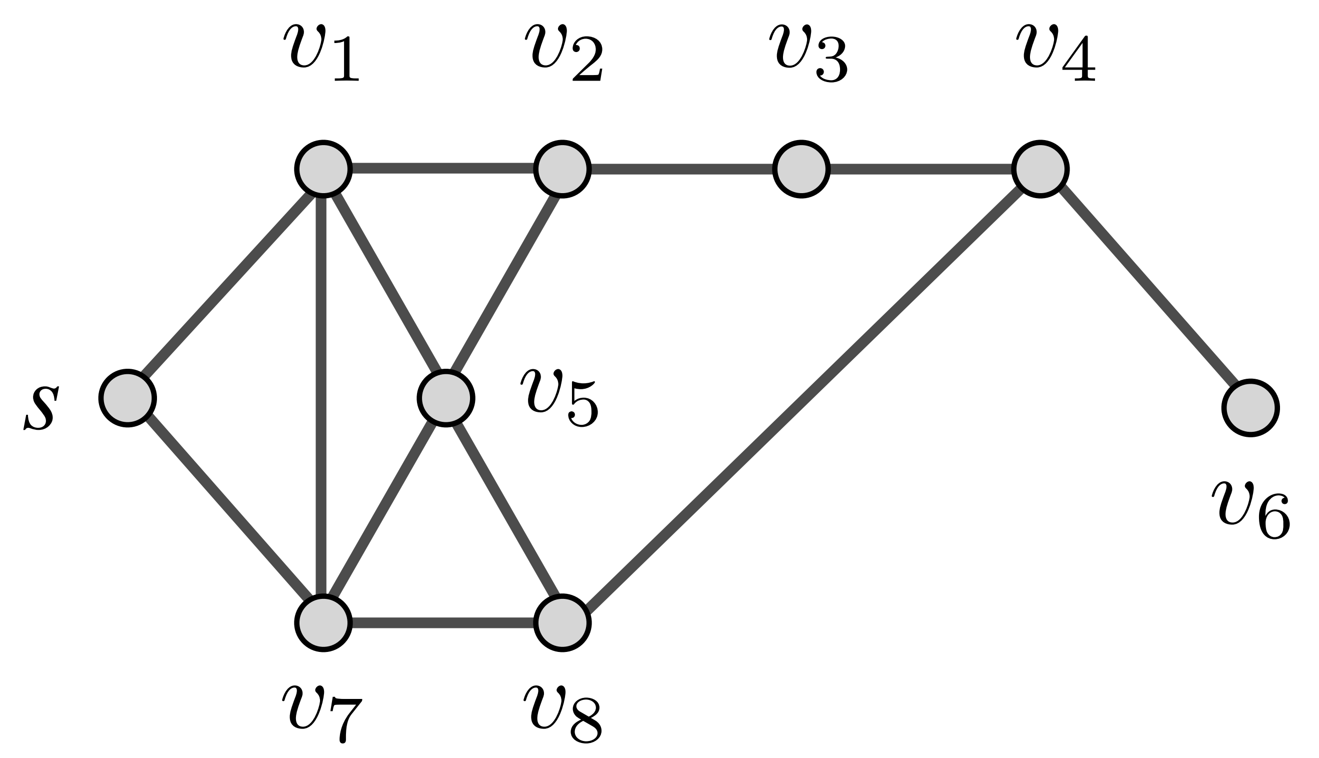
Algorithmus 3.7

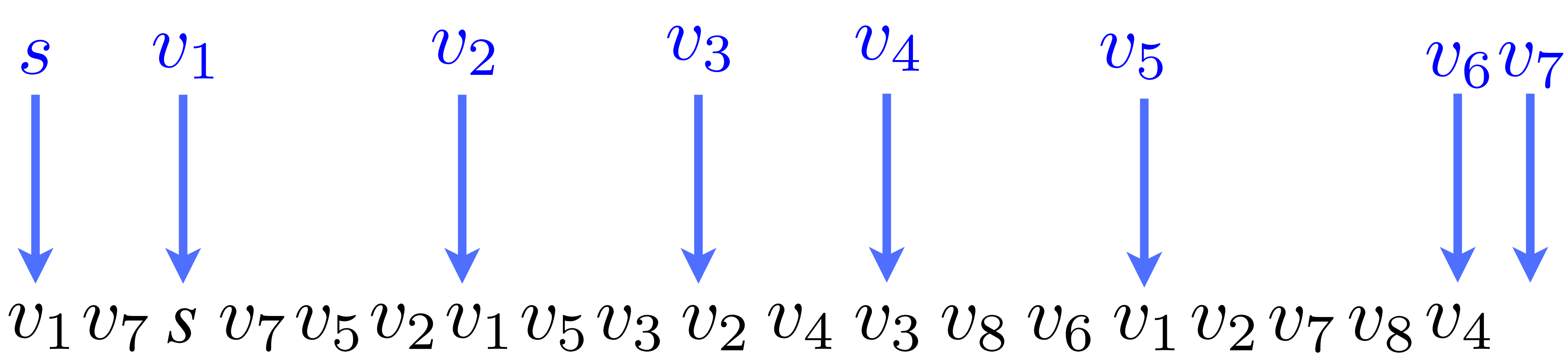
INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
           2.2.1. R:=R\{v}
      2.3. ELSE {
           2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
           2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
           }
    }
3.  STOP
```

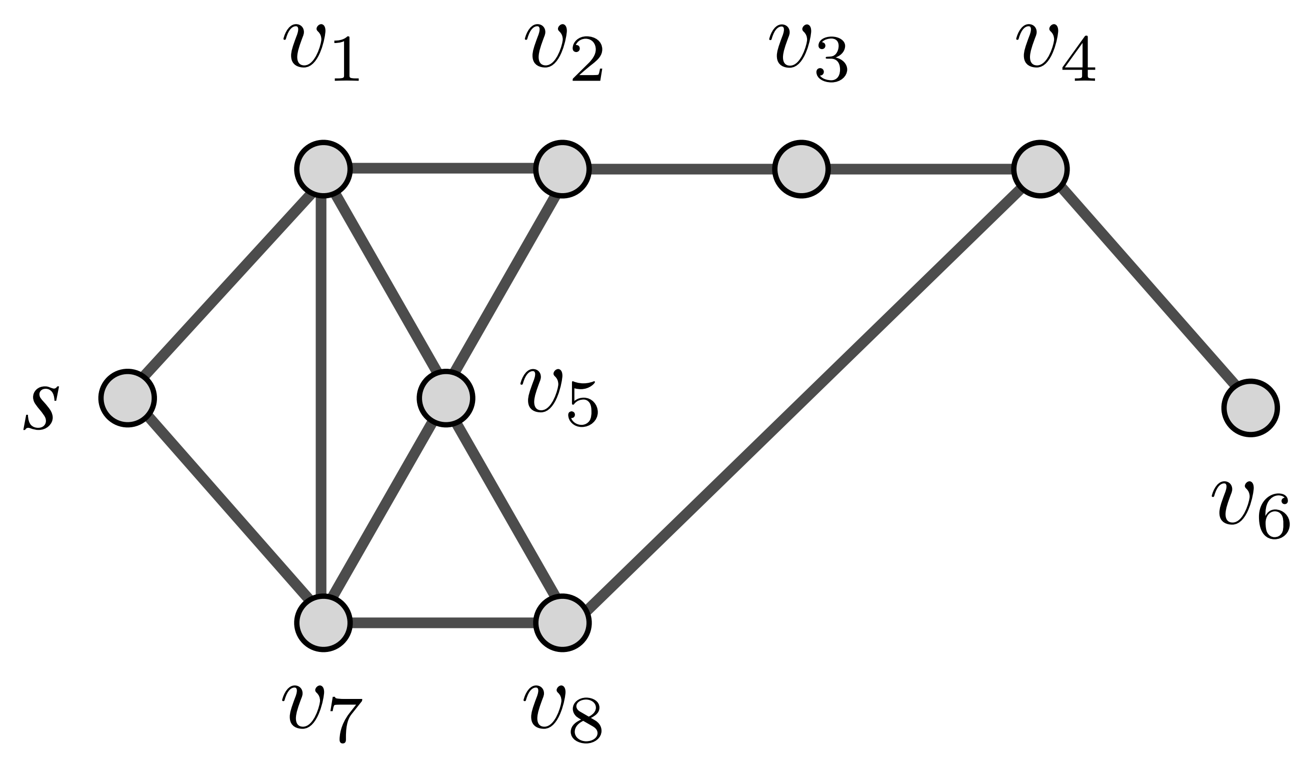





Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }
   - }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \; v_7$

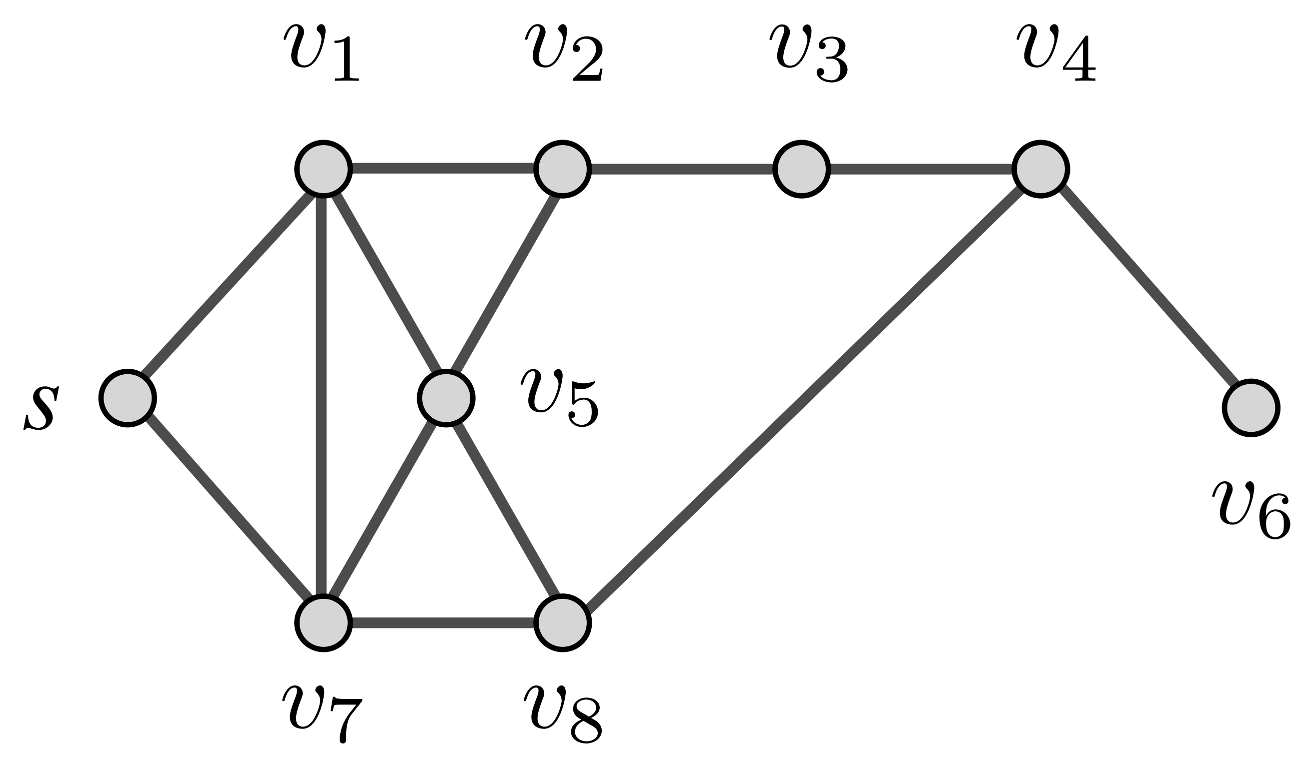
$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }
   - }
3. STOP

2



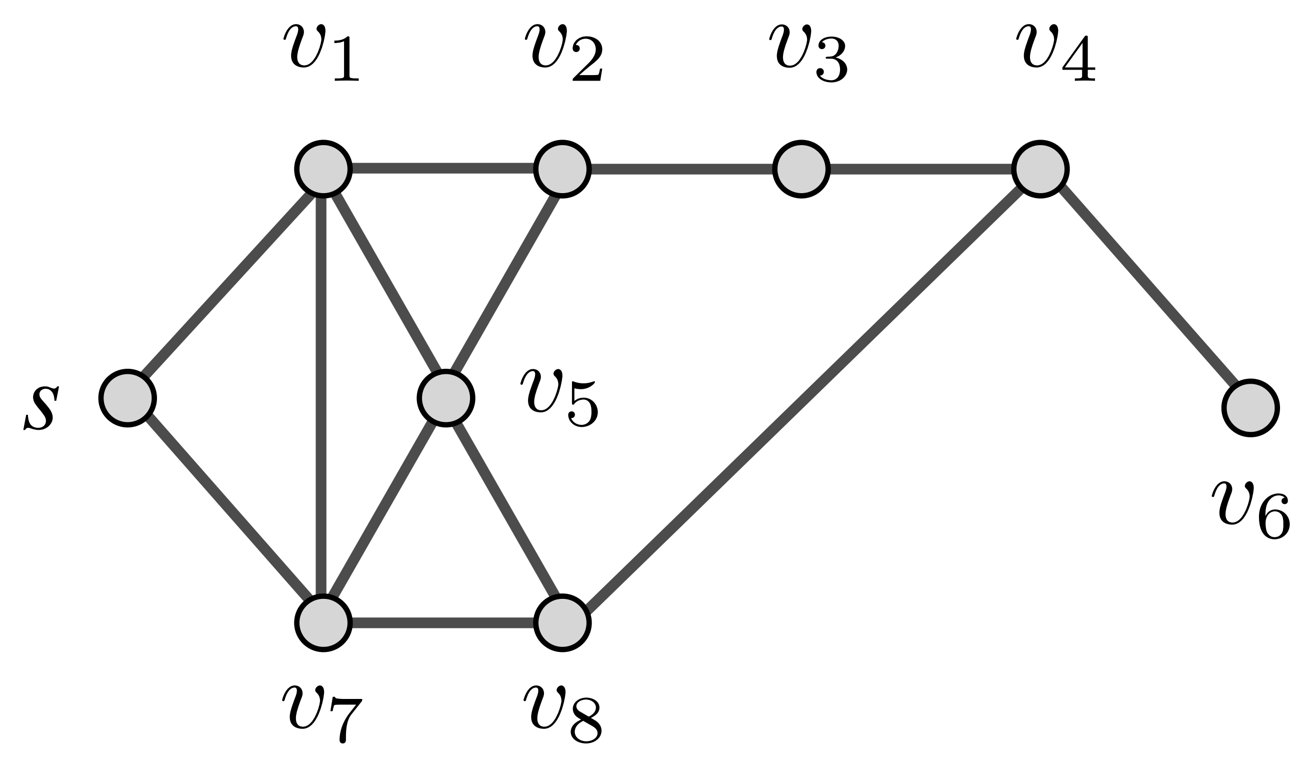
$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7$

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1$

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   2.1. Wähle $v \in R$
   2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
      2.2.1. $R:=R \setminus \{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \; v_7$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5$

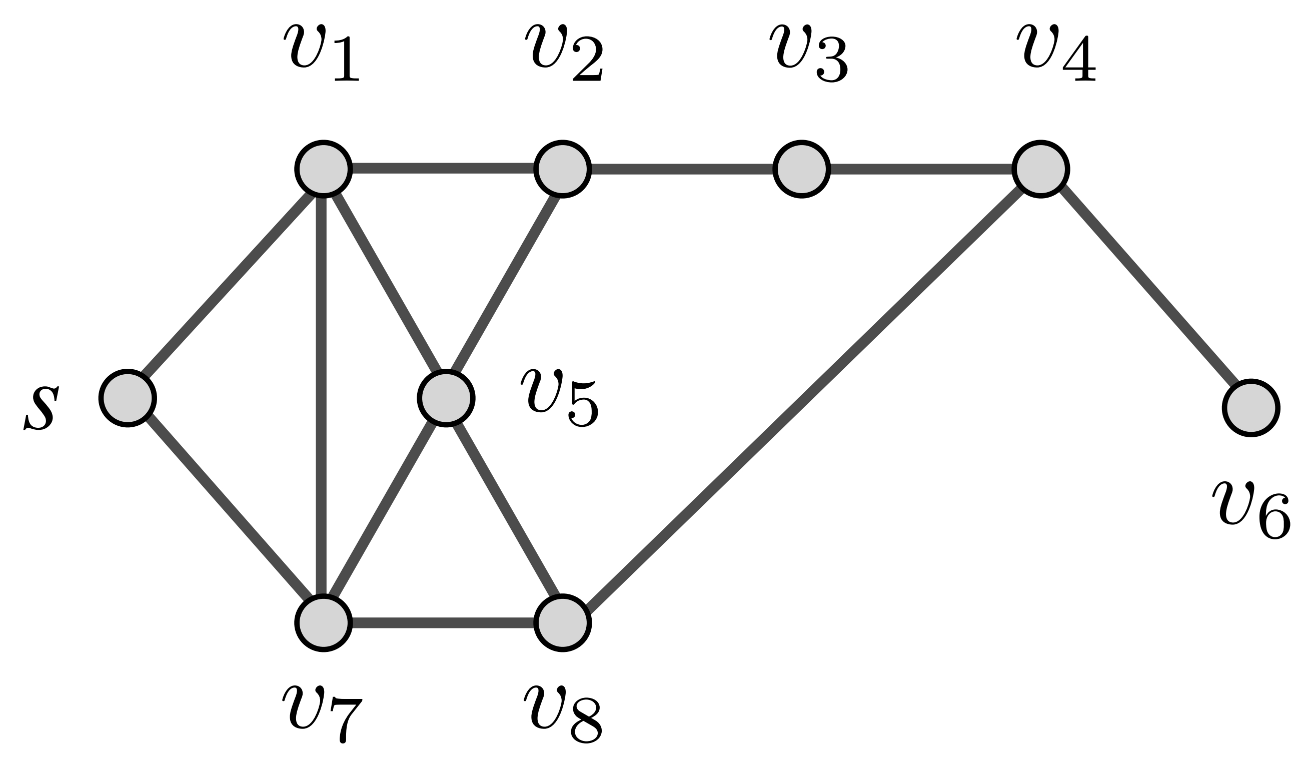
Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }
   - }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8$

Algorithmus 3.7

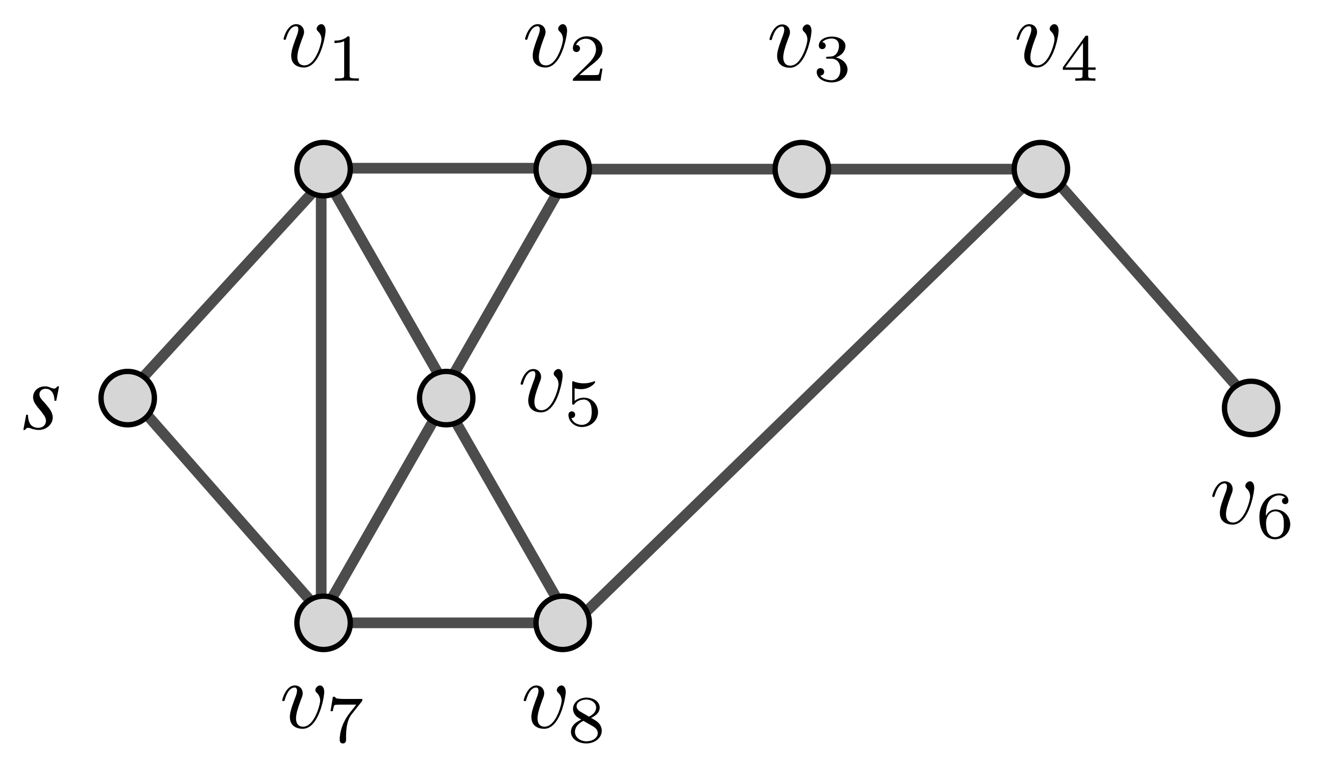
INPUT: Graph $G = (V,E)$, Knoten s

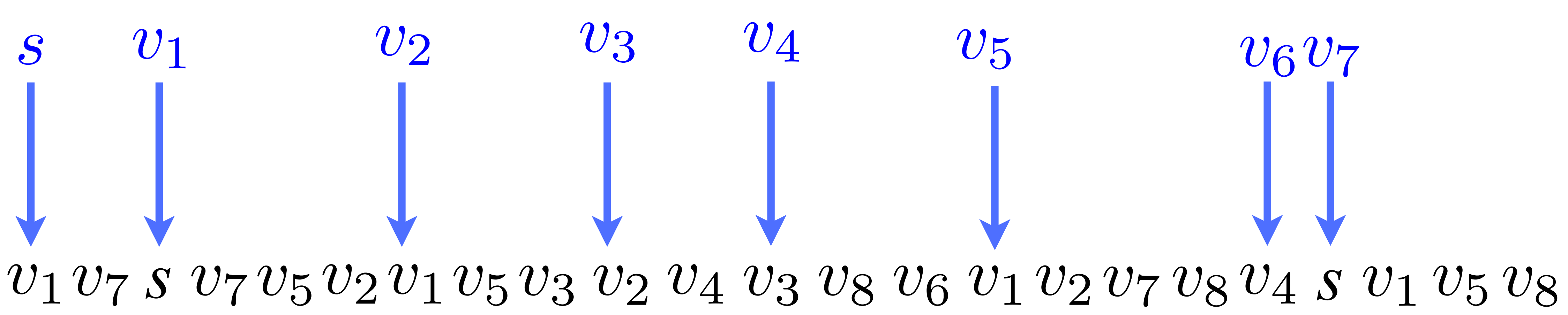
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }

   }
3. STOP



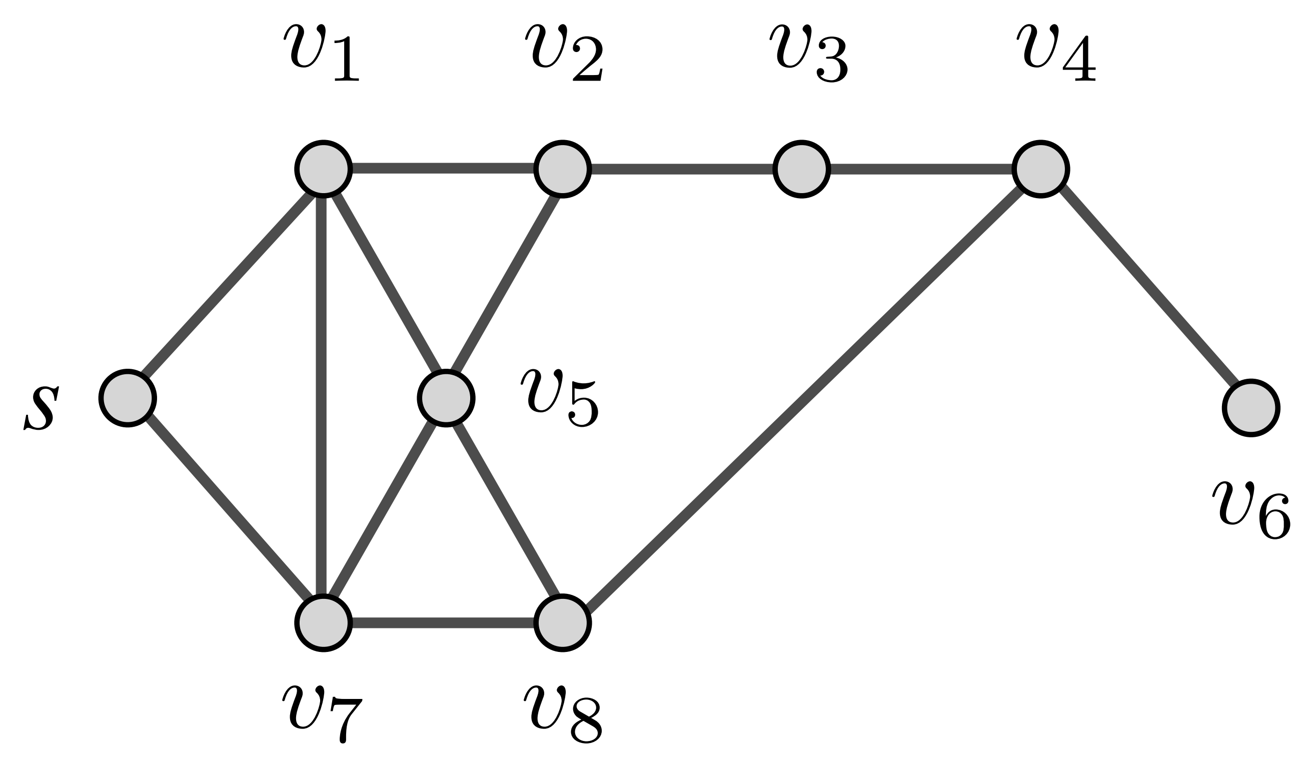


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v ∈ R
   - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w} ∈ E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

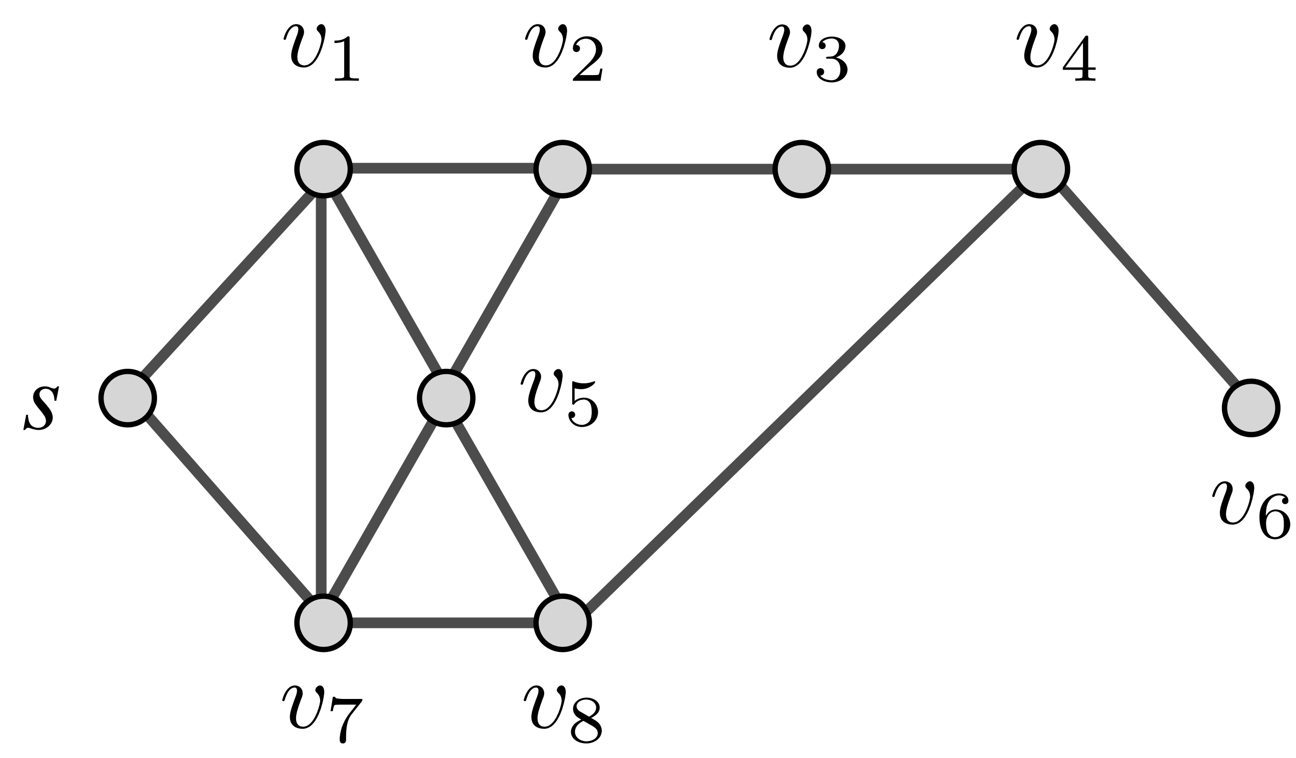
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP



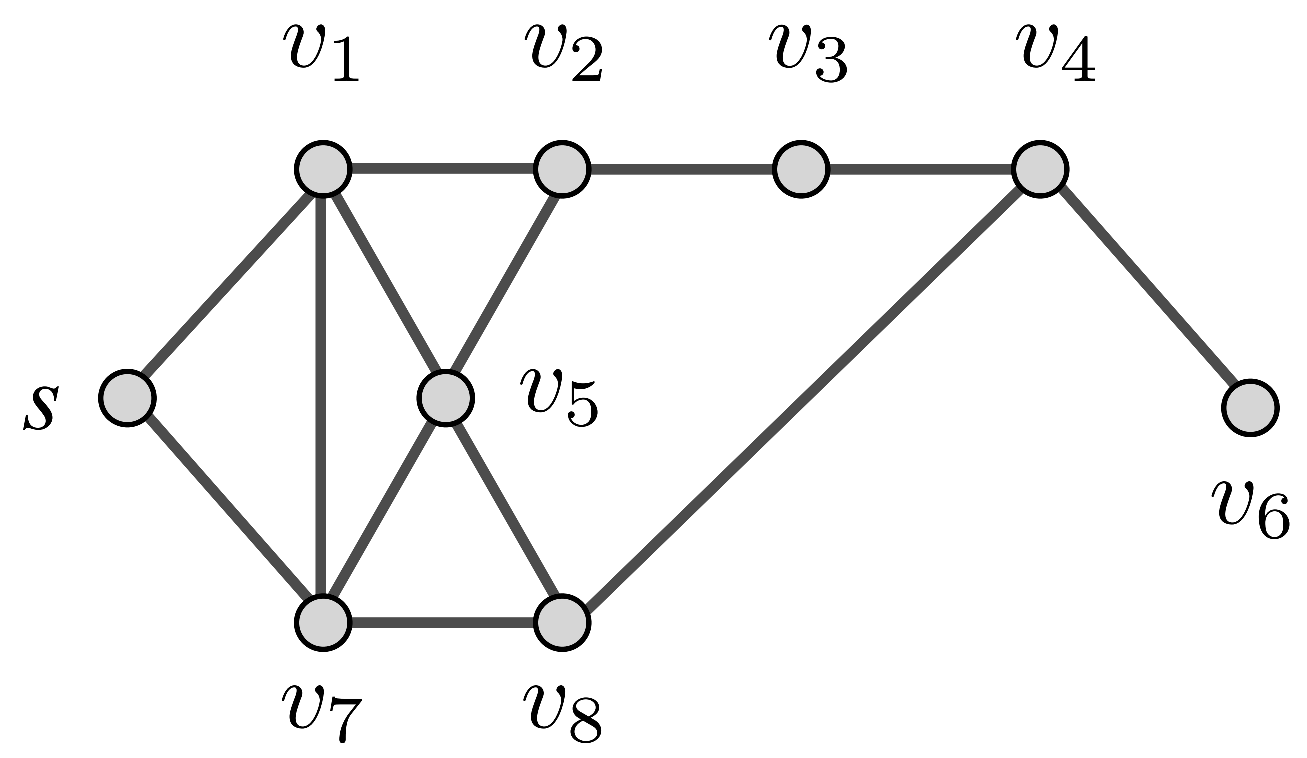
s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
           2.2.1. R:=R\{v}
      2.3. ELSE {
           2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
           2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
           }
    }
3.  STOP
```

s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

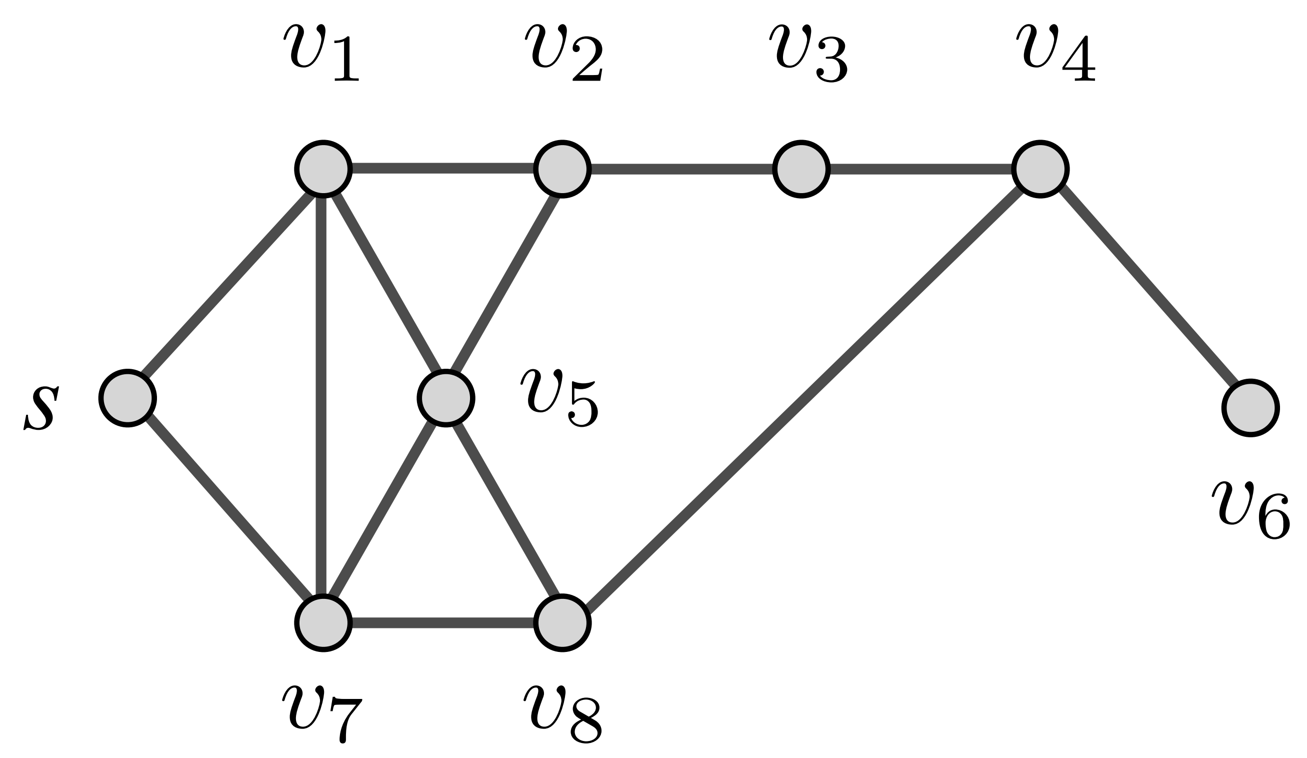
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

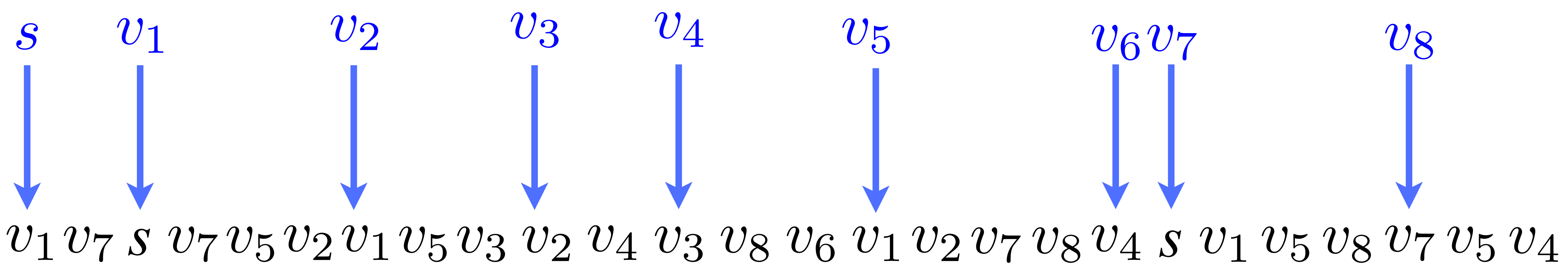
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

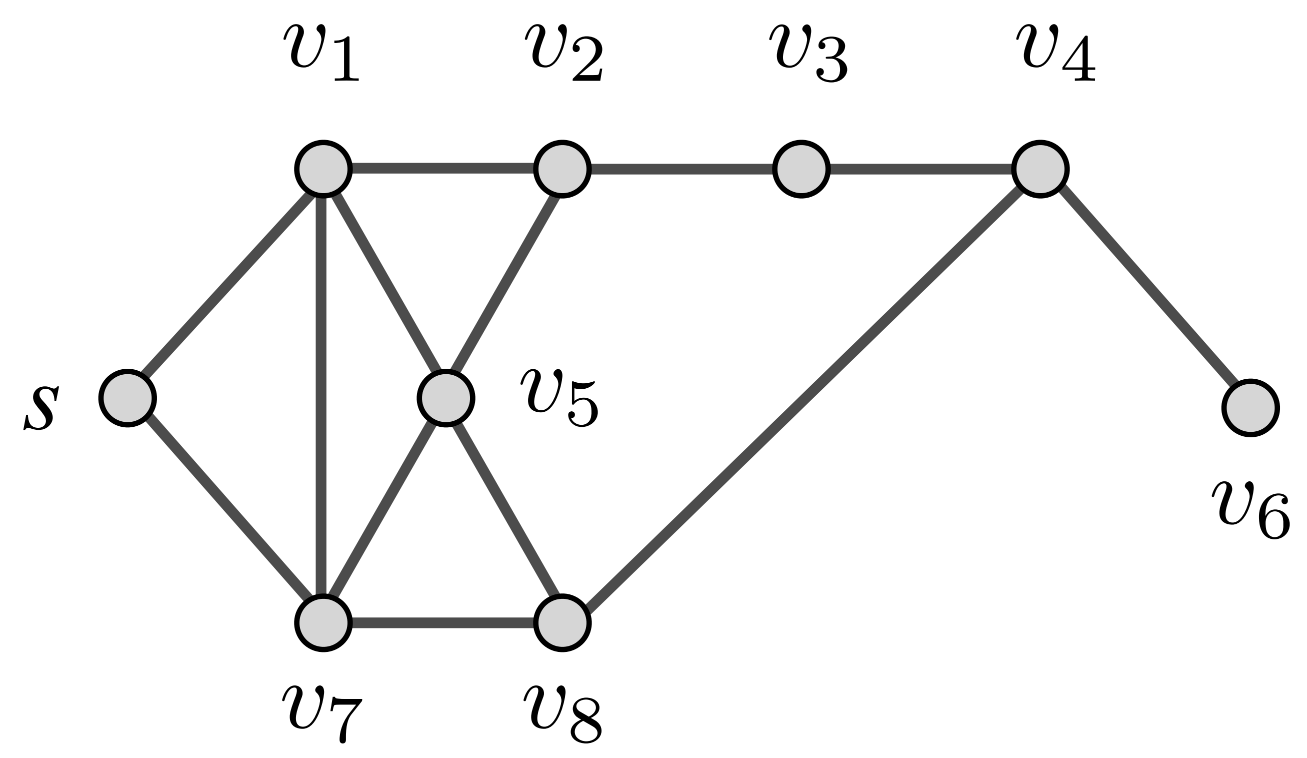
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

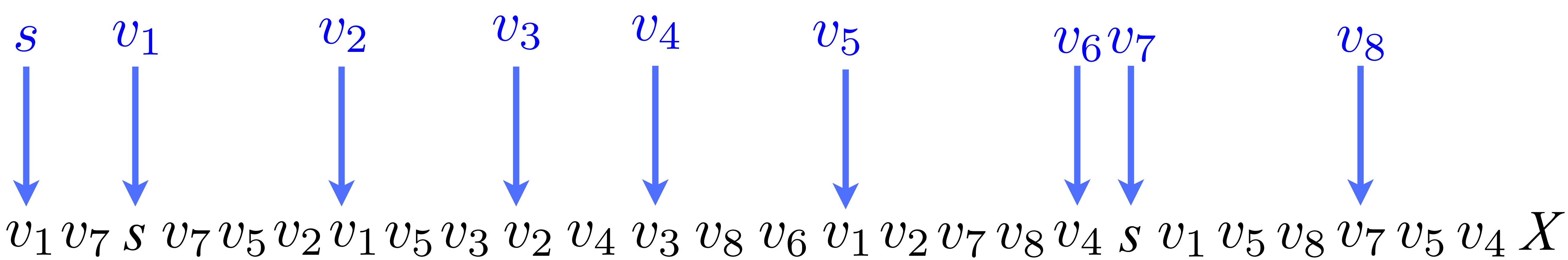
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

2. WHILE ($R \neq \emptyset$) DO {

    2.1. Wähle $v \in R$

    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

        2.2.1. $R:=R \setminus \{v\}$

    2.3. ELSE {

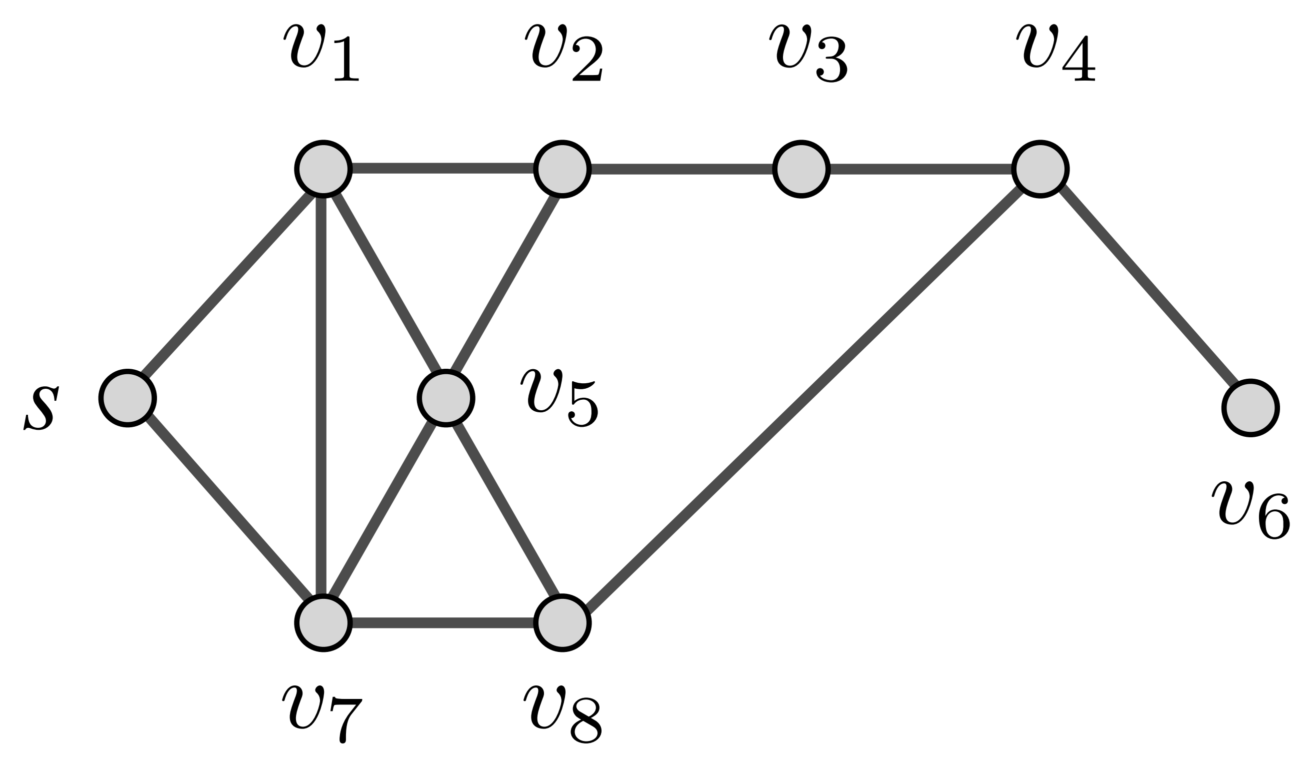
        2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$

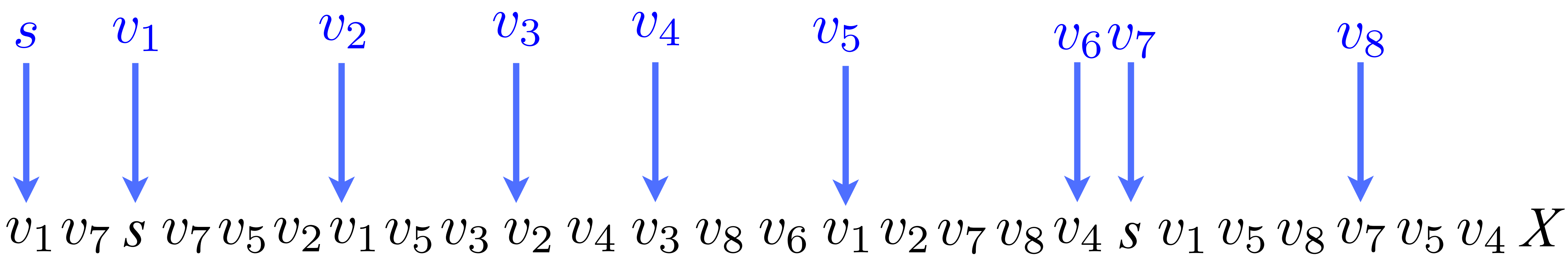
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

}

3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

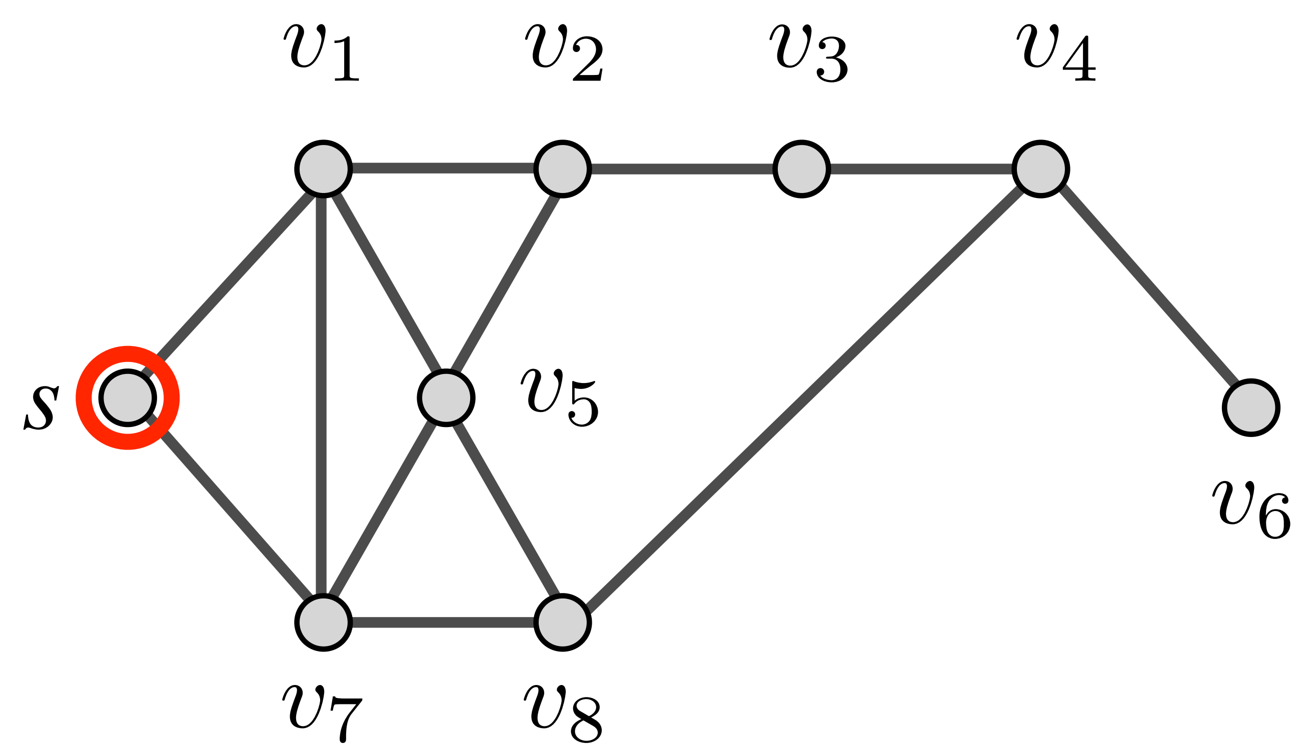
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5\, v_8\, v_7\, v_5\, v_4\, X$

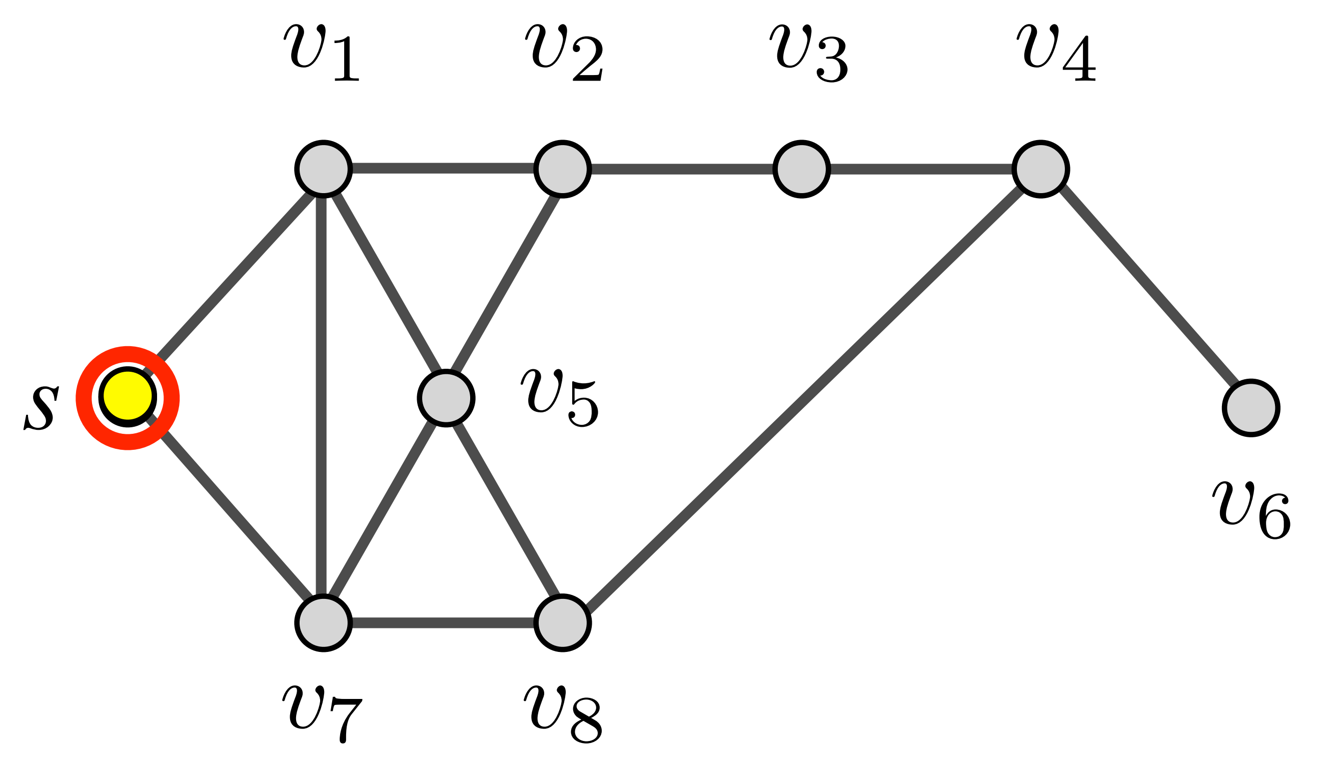
Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP

2



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

Algorithmus 3.7

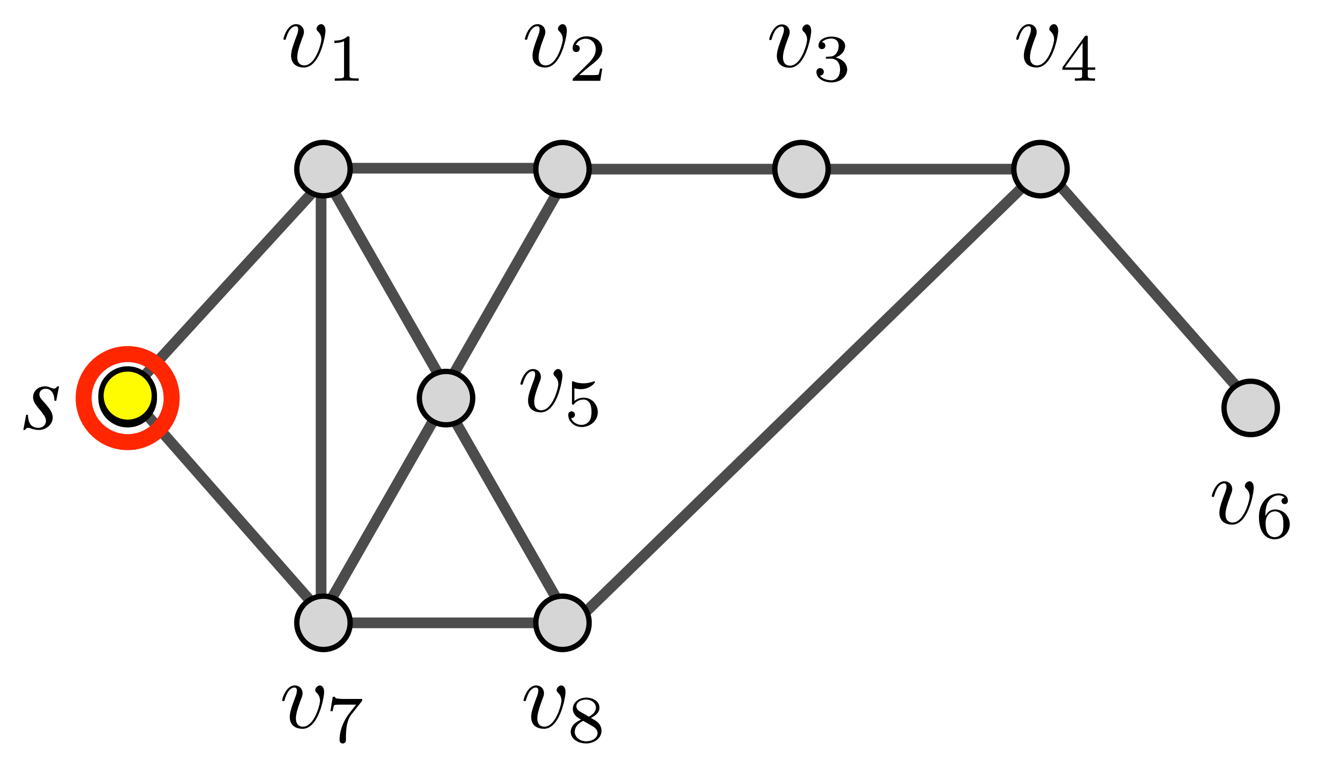
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

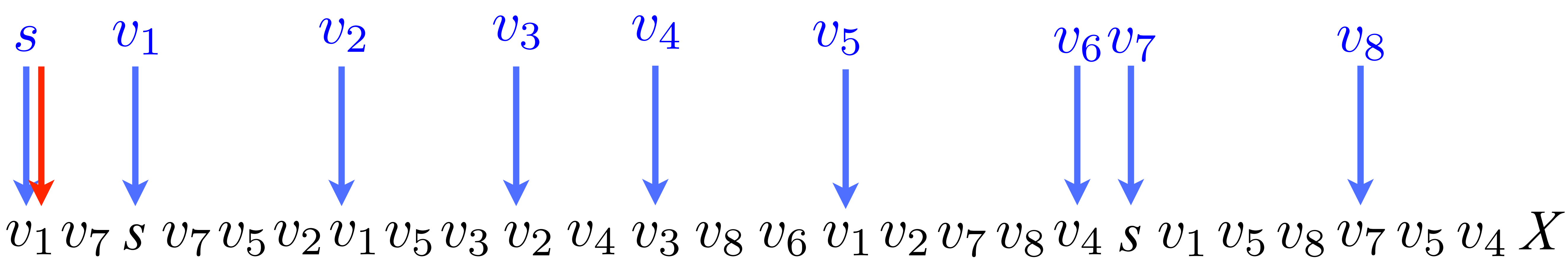
Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

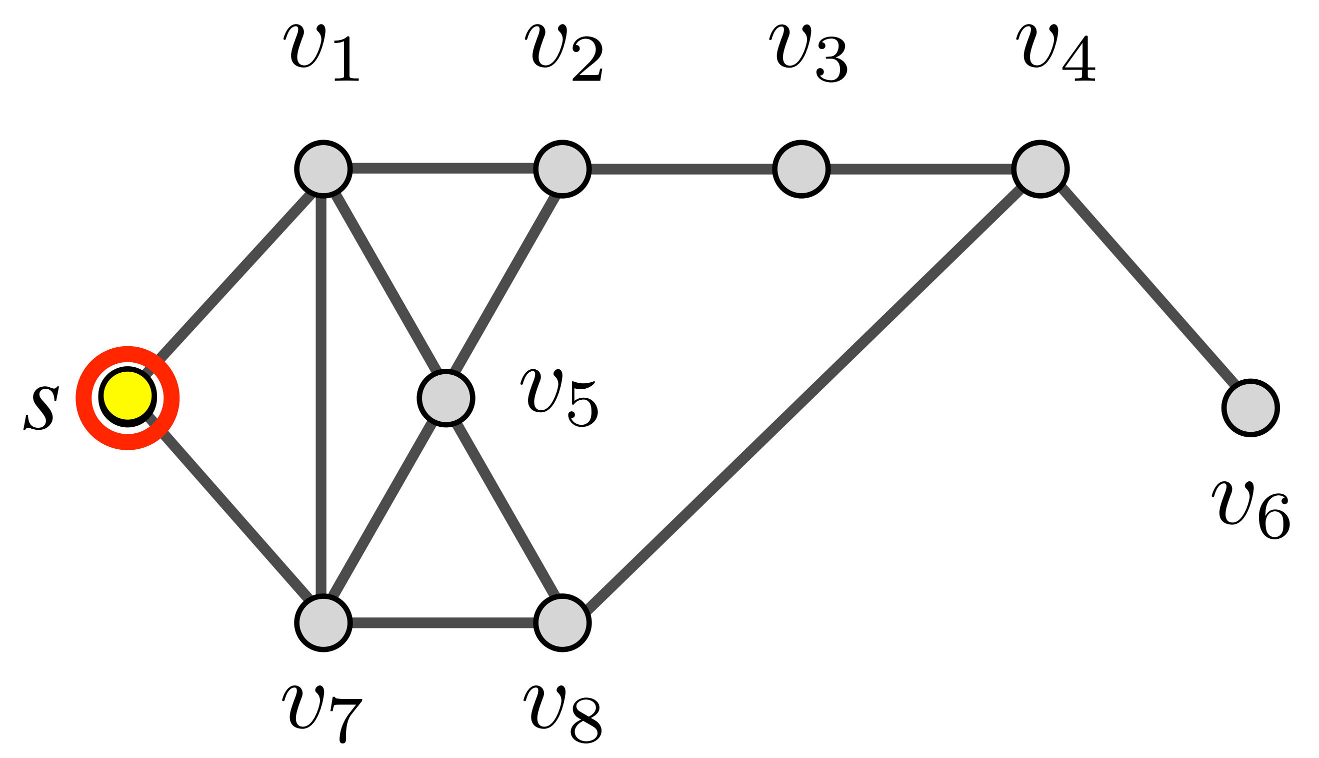
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP

2



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

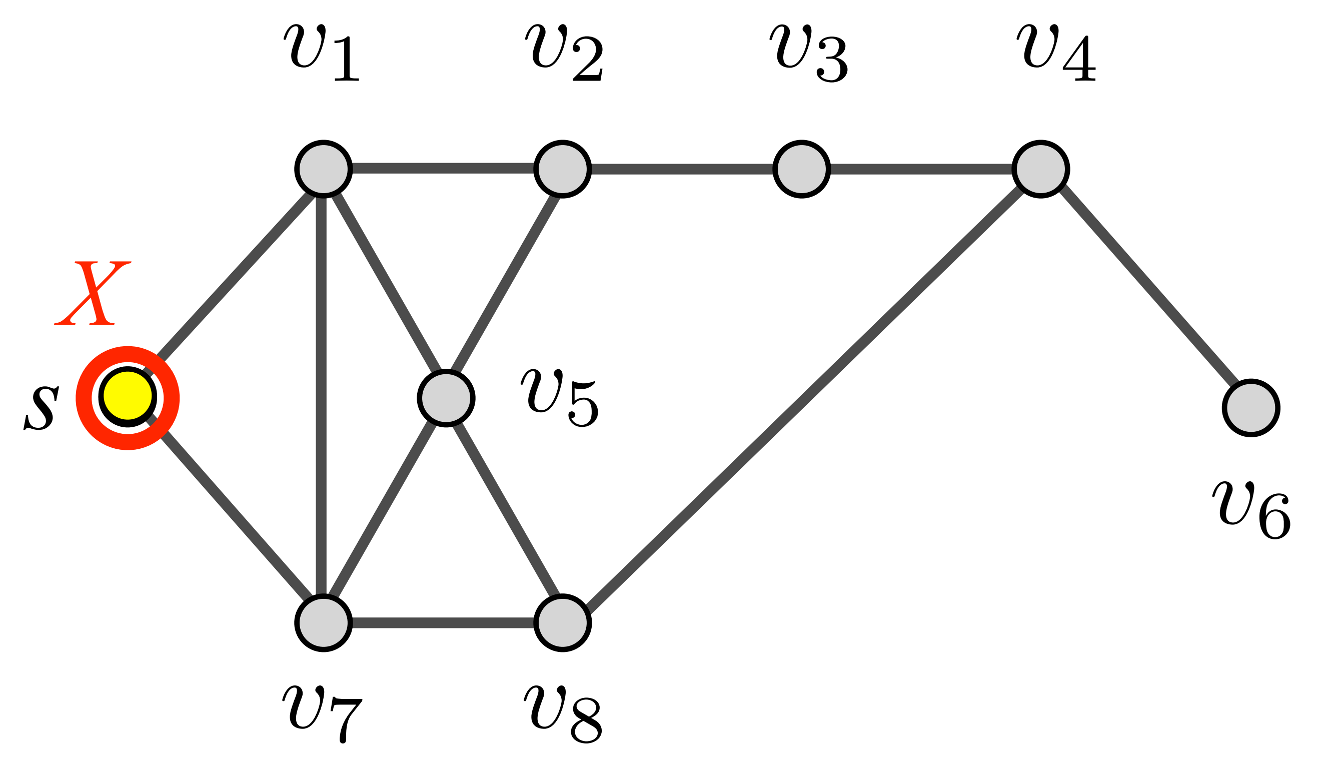
Algorithmus 3.7

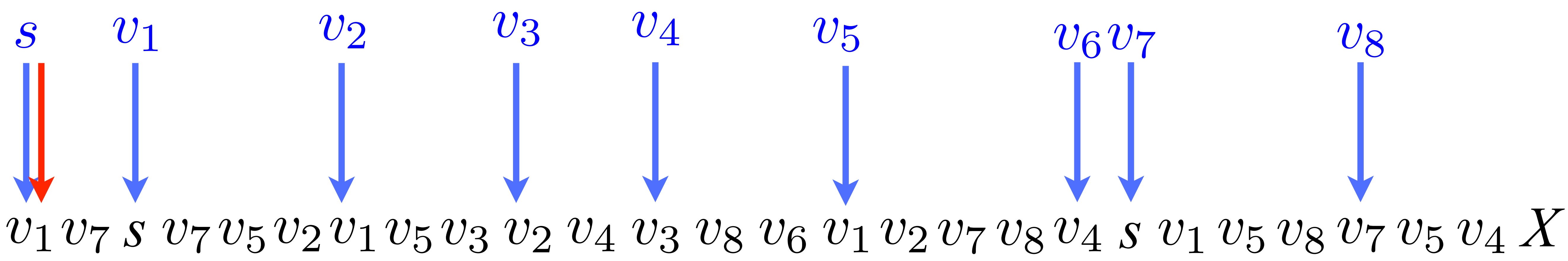
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
          2.2.1. R:=R\{v}
      2.3. ELSE {
          2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
          2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
          }
    }
3.  STOP
```



Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

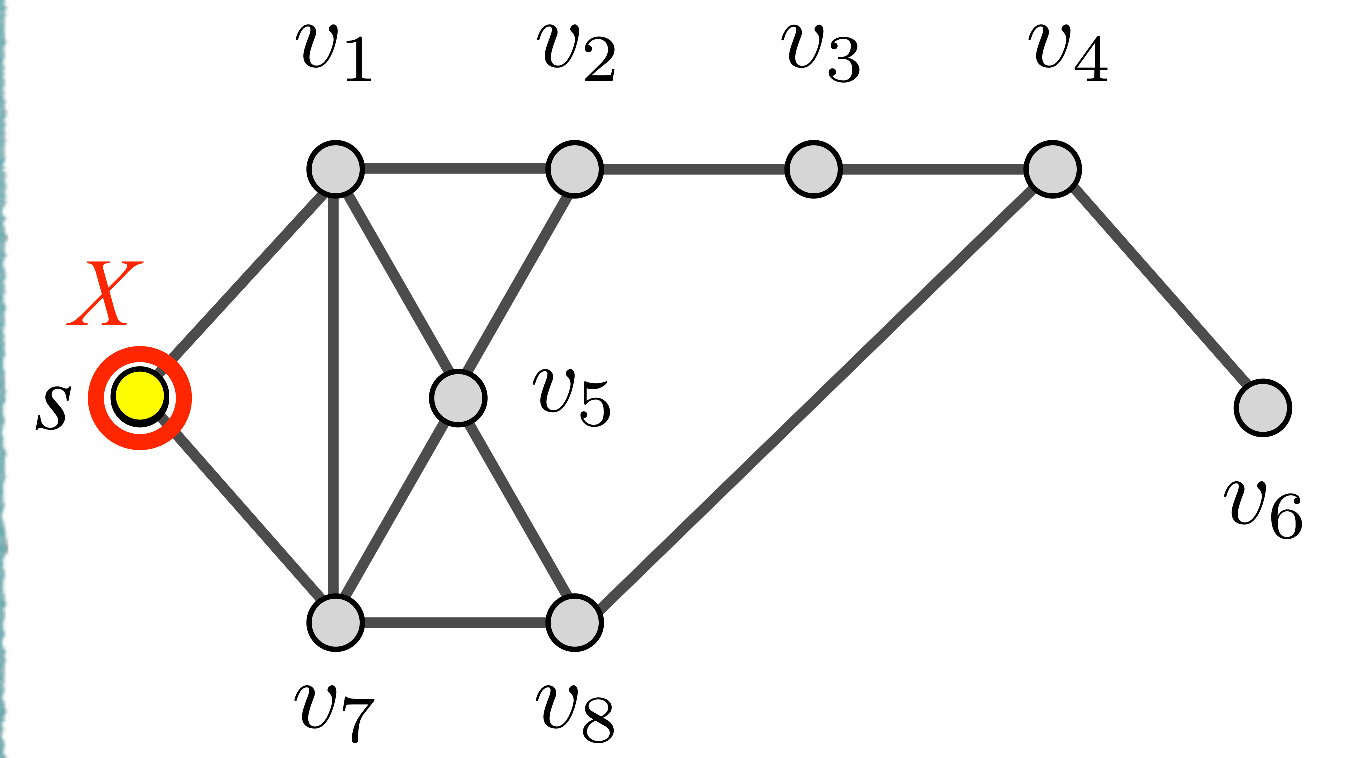
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

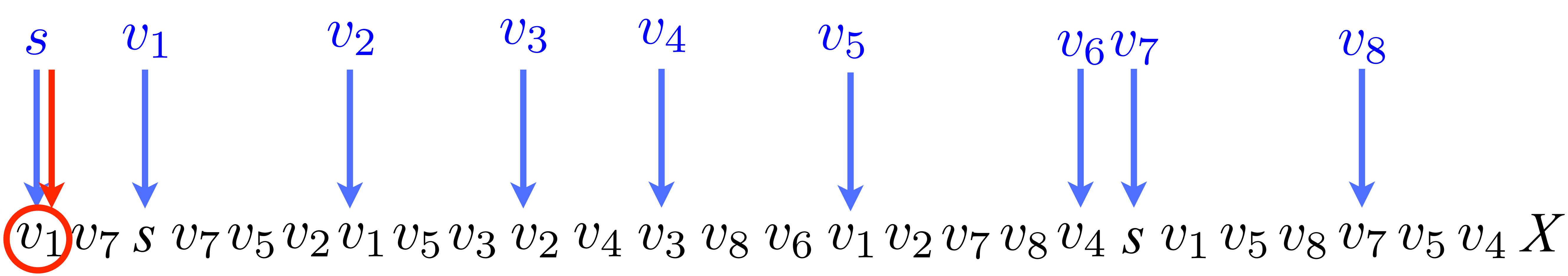
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





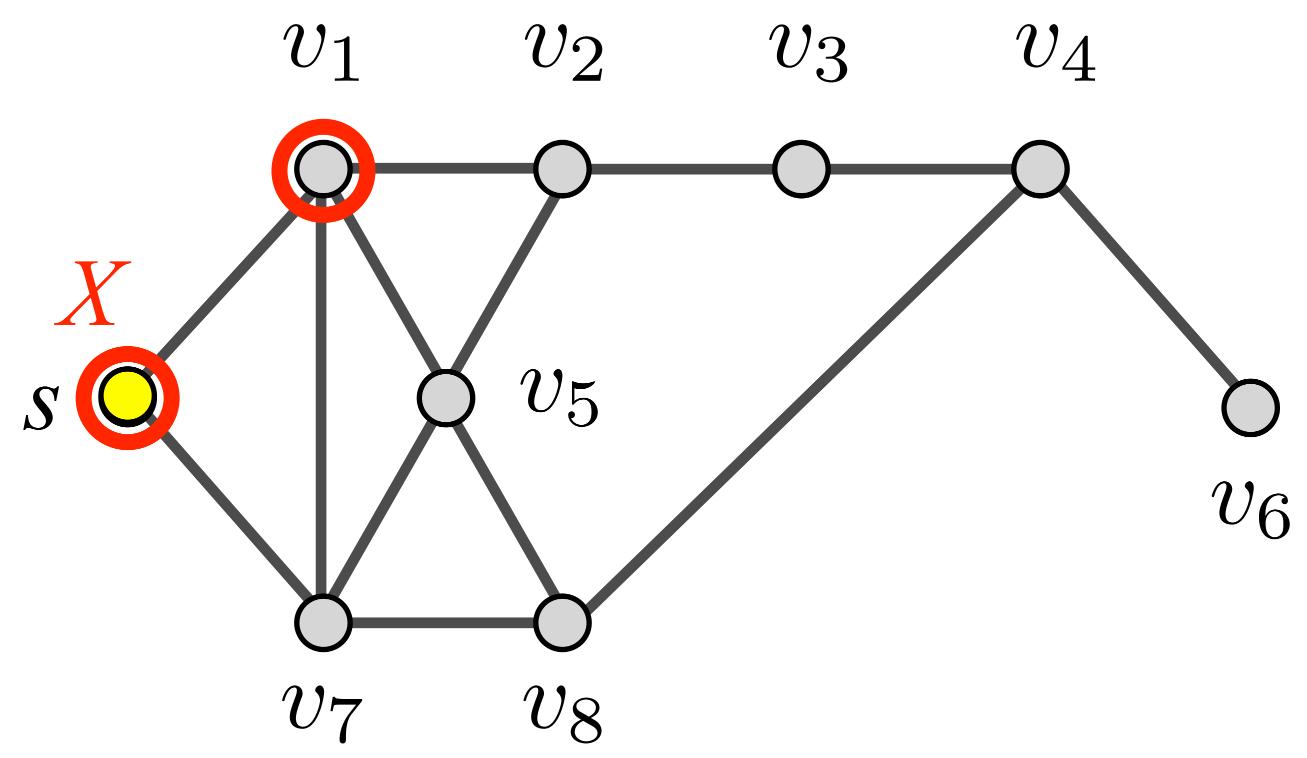
Algorithmus 3.7

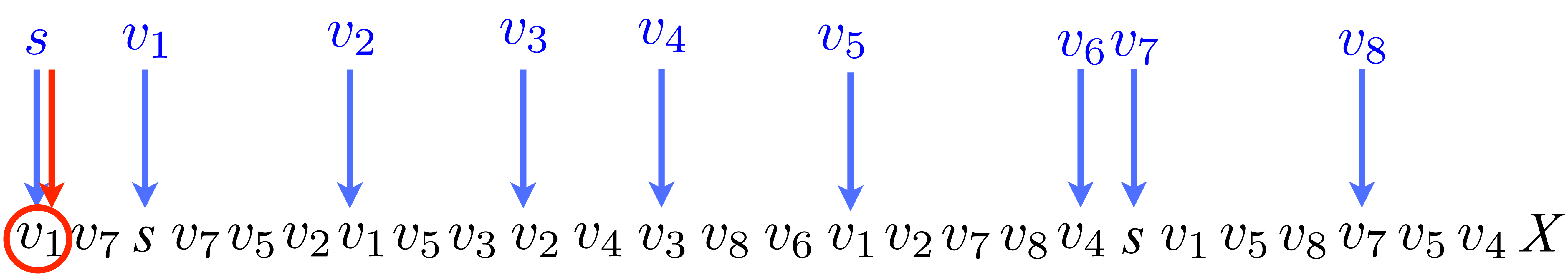
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v ∈ R

    2.2. IF (es gibt kein w ∈ V\Y mit e={v,w}∈E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

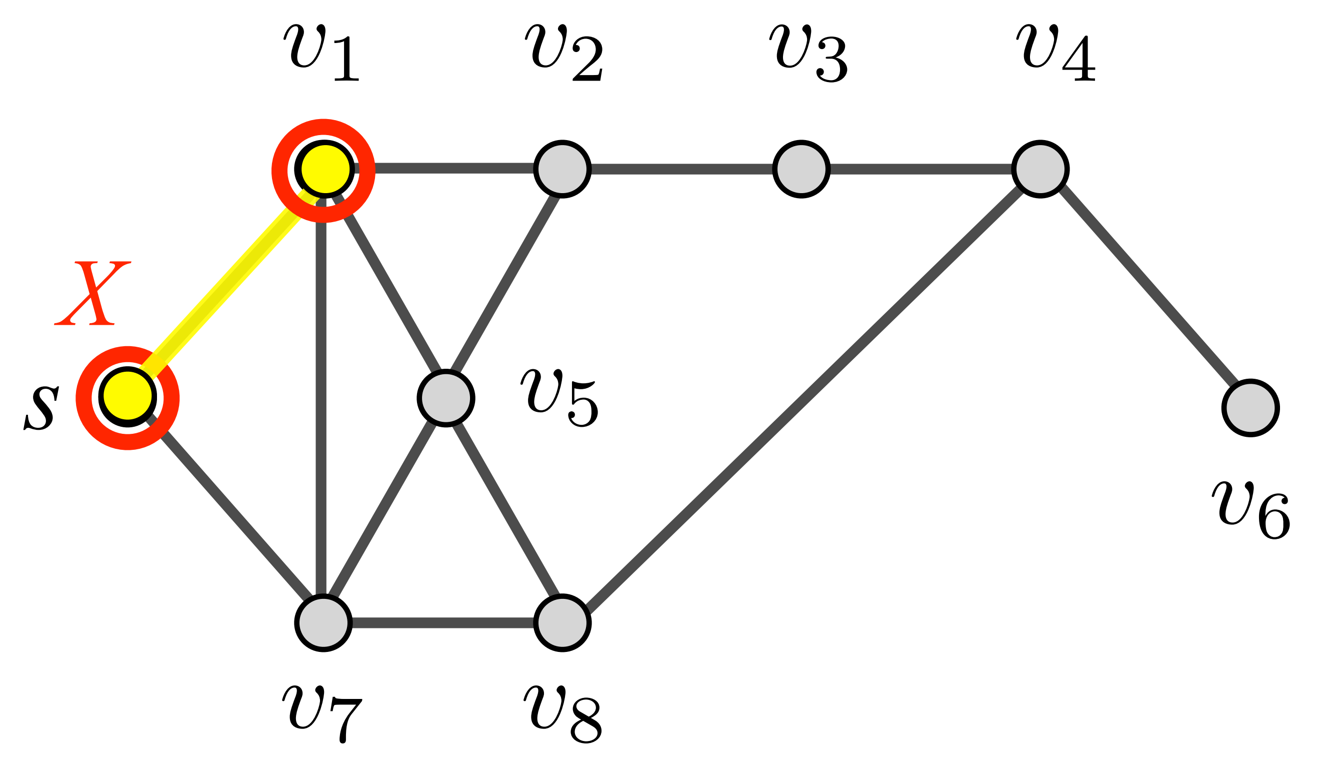
        2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E

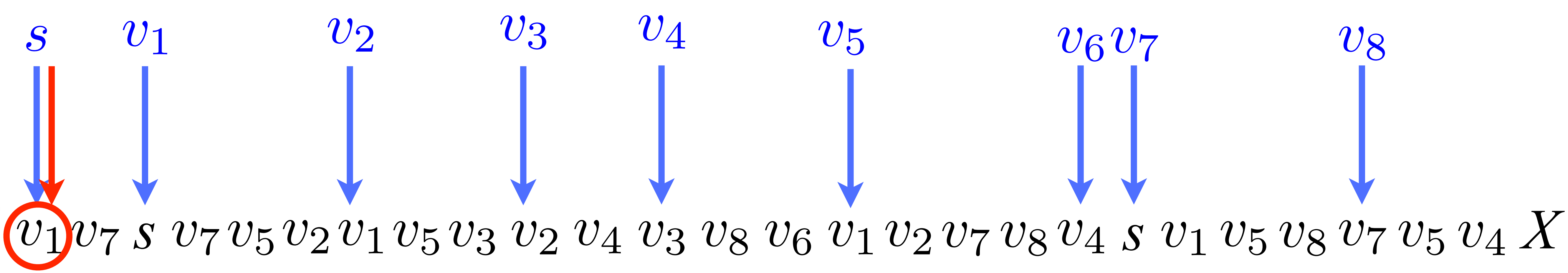
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }

3. STOP

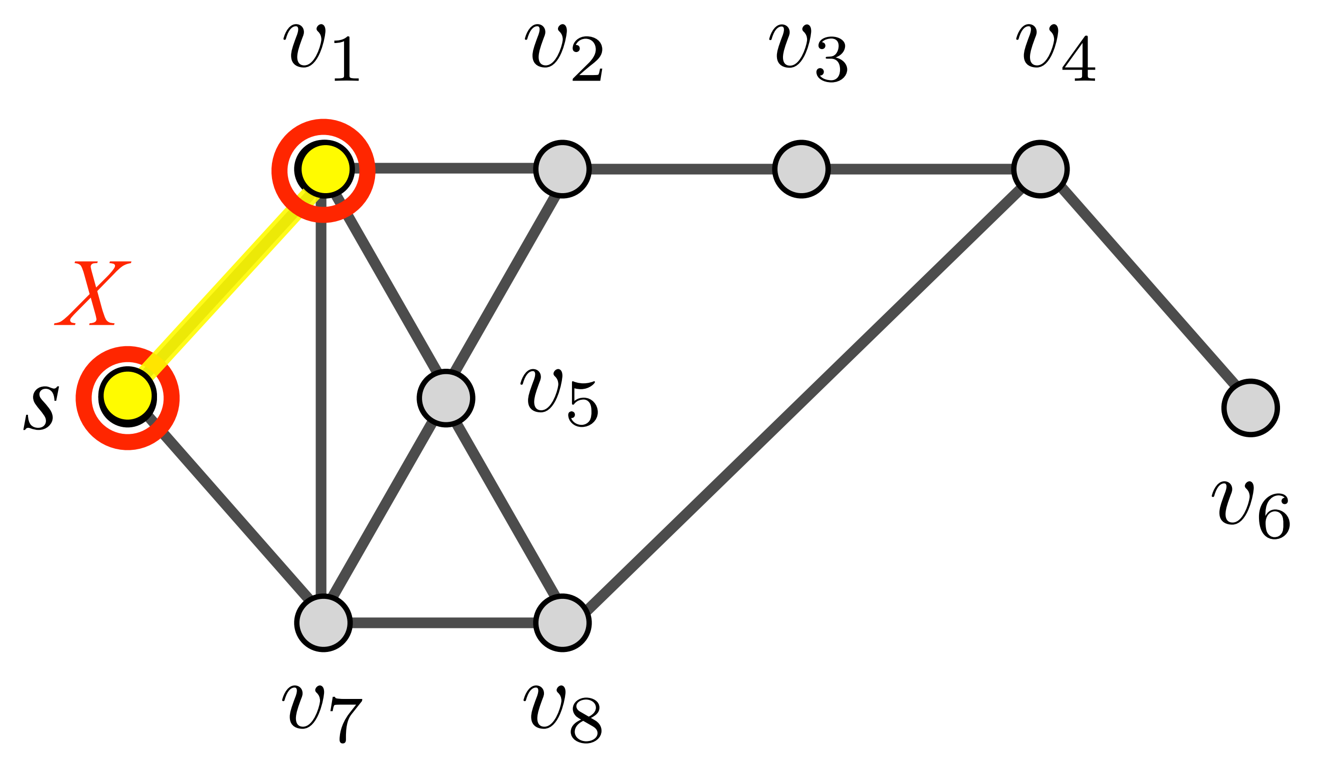




Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈ V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP



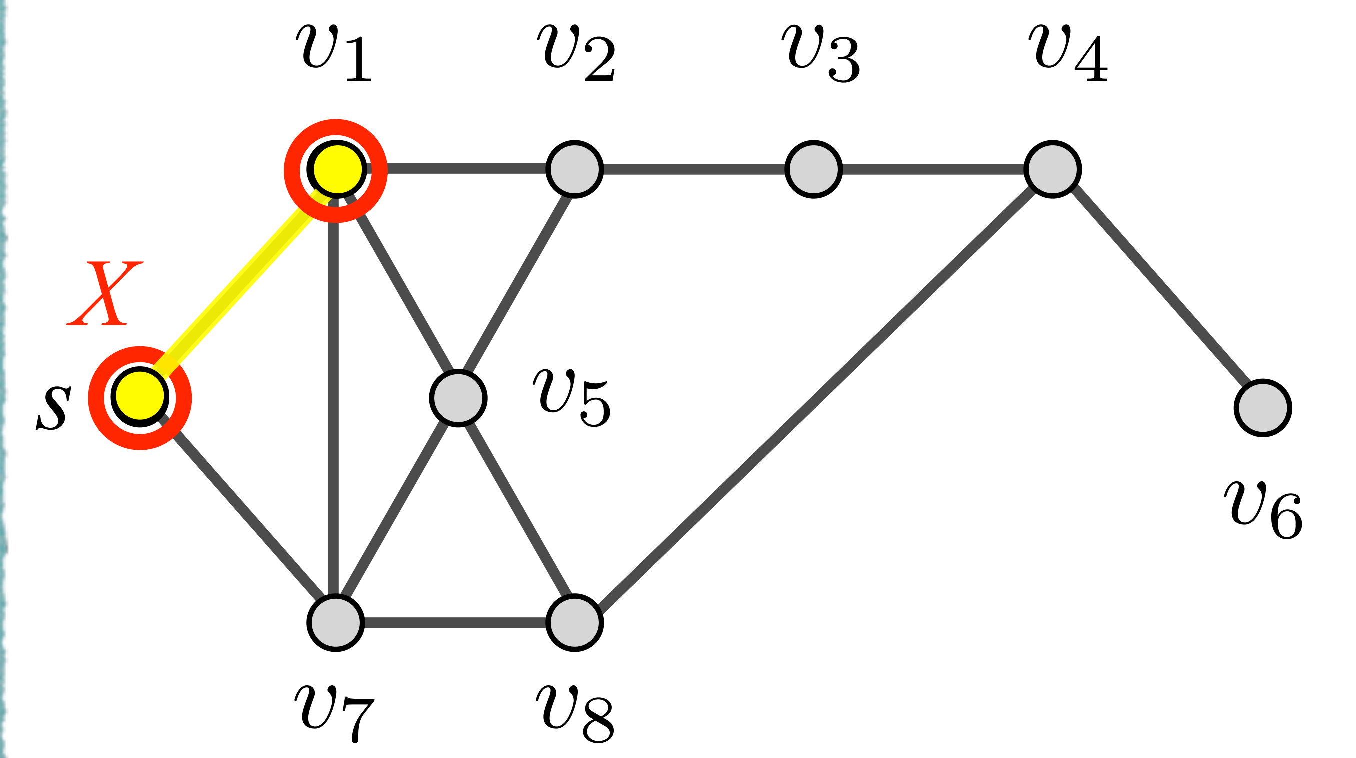
$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \, v_7 \quad v_8$

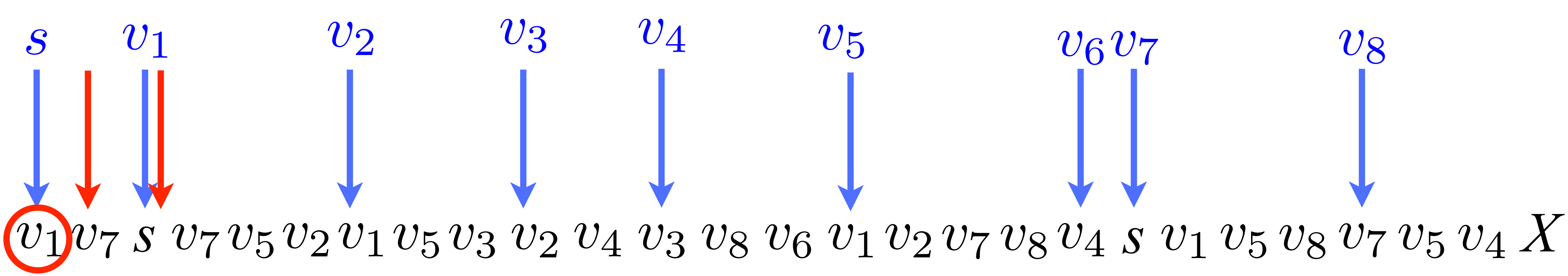
$v_1 \, v_7 \, s \, v_7 \, v_5 \, v_2 \, v_1 \, v_5 \, v_3 \, v_2 \, v_4 \, v_3 \, v_8 \, v_6 \, v_1 \, v_2 \, v_7 \, v_8 \, v_4 \, s \, v_1 \, v_5 \, v_8 \, v_7 \, v_5 \, v_4 \, X$

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v ∈ R
    2.2. IF (es gibt kein w ∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

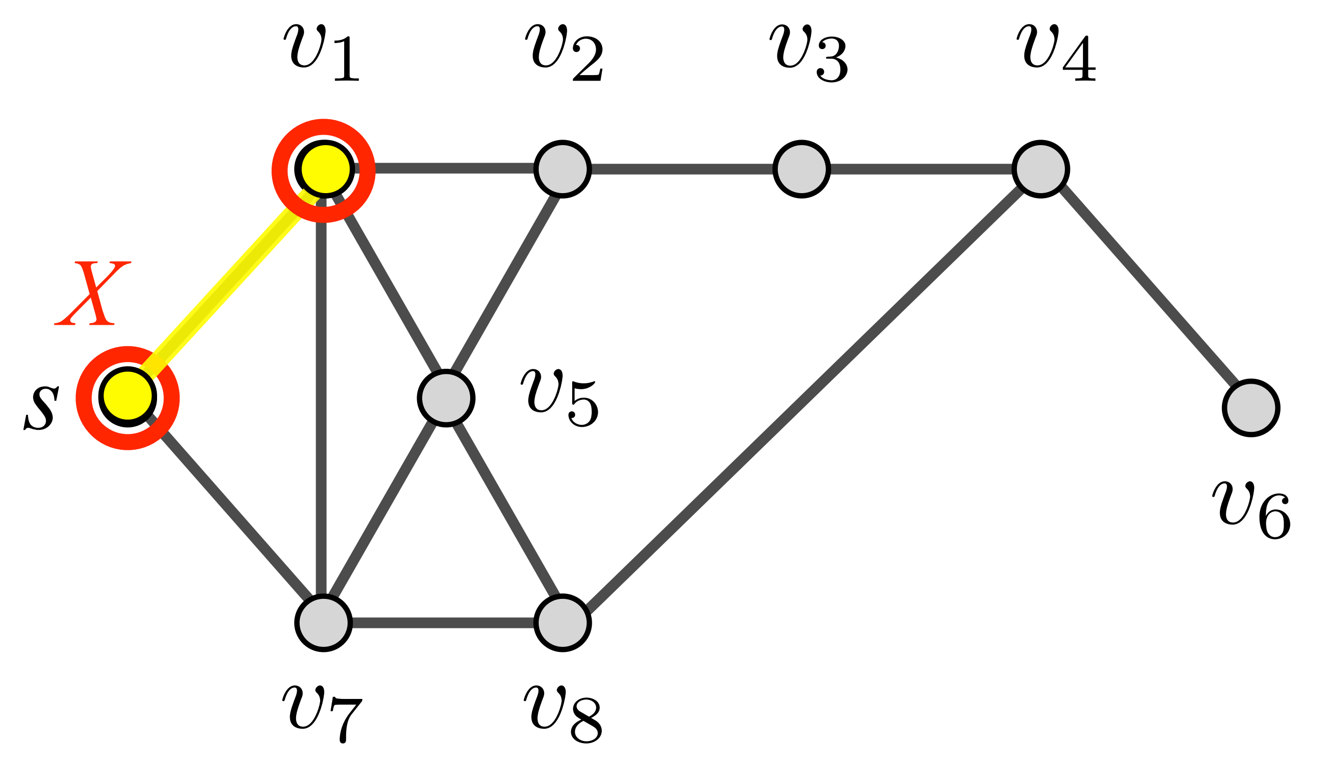
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7 \quad v_8$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5\, v_8\, v_7\, v_5\, v_4\, X$

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

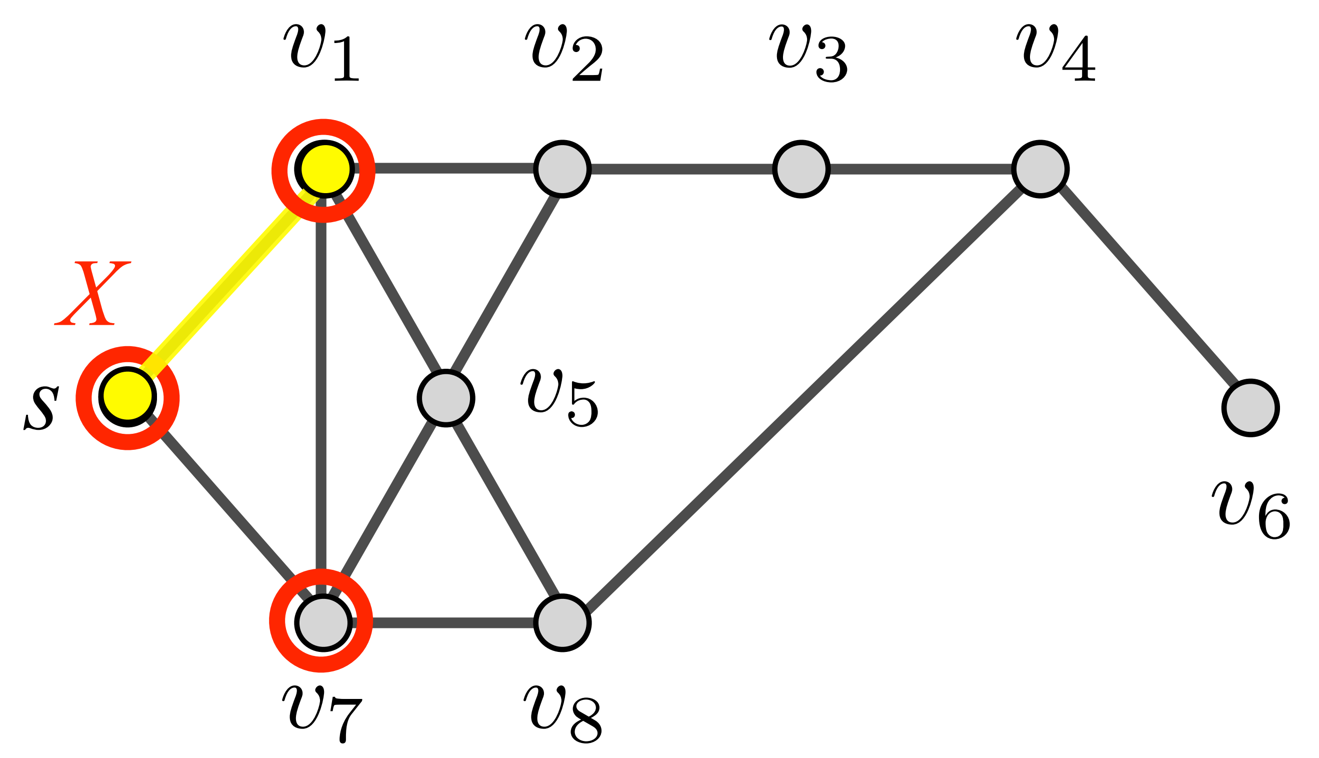
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

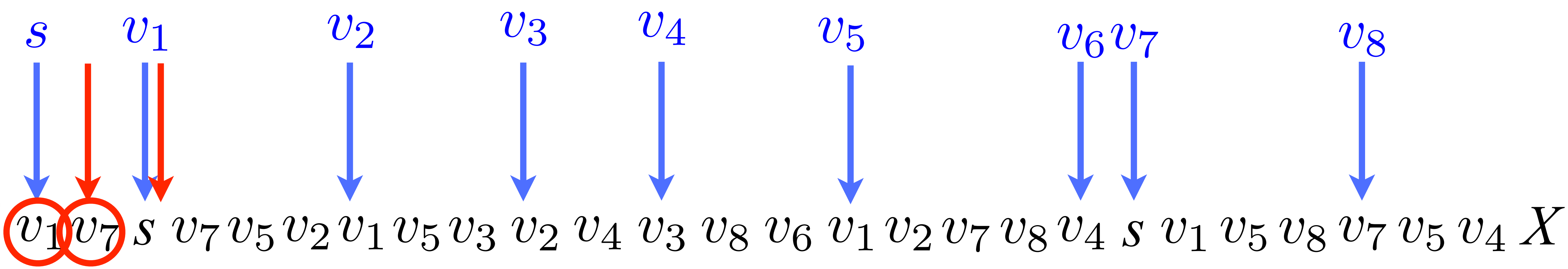
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





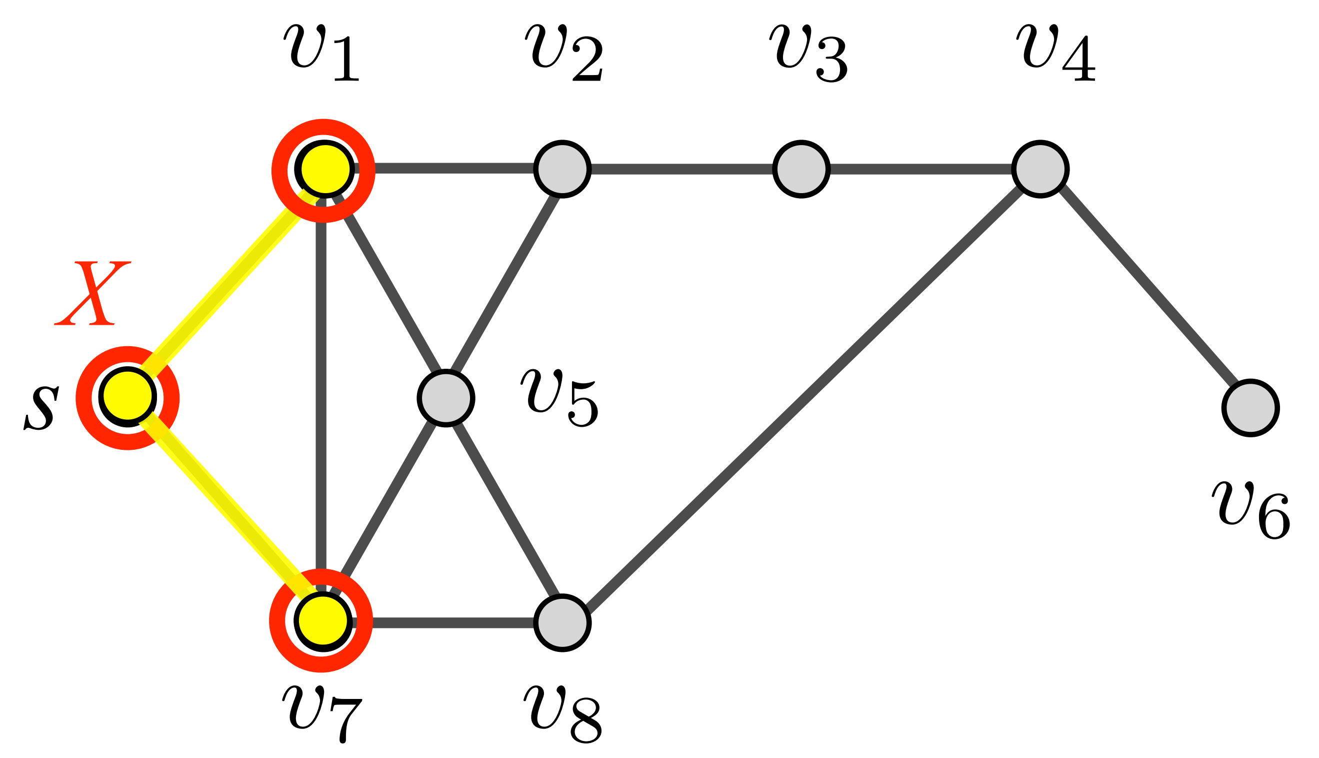
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

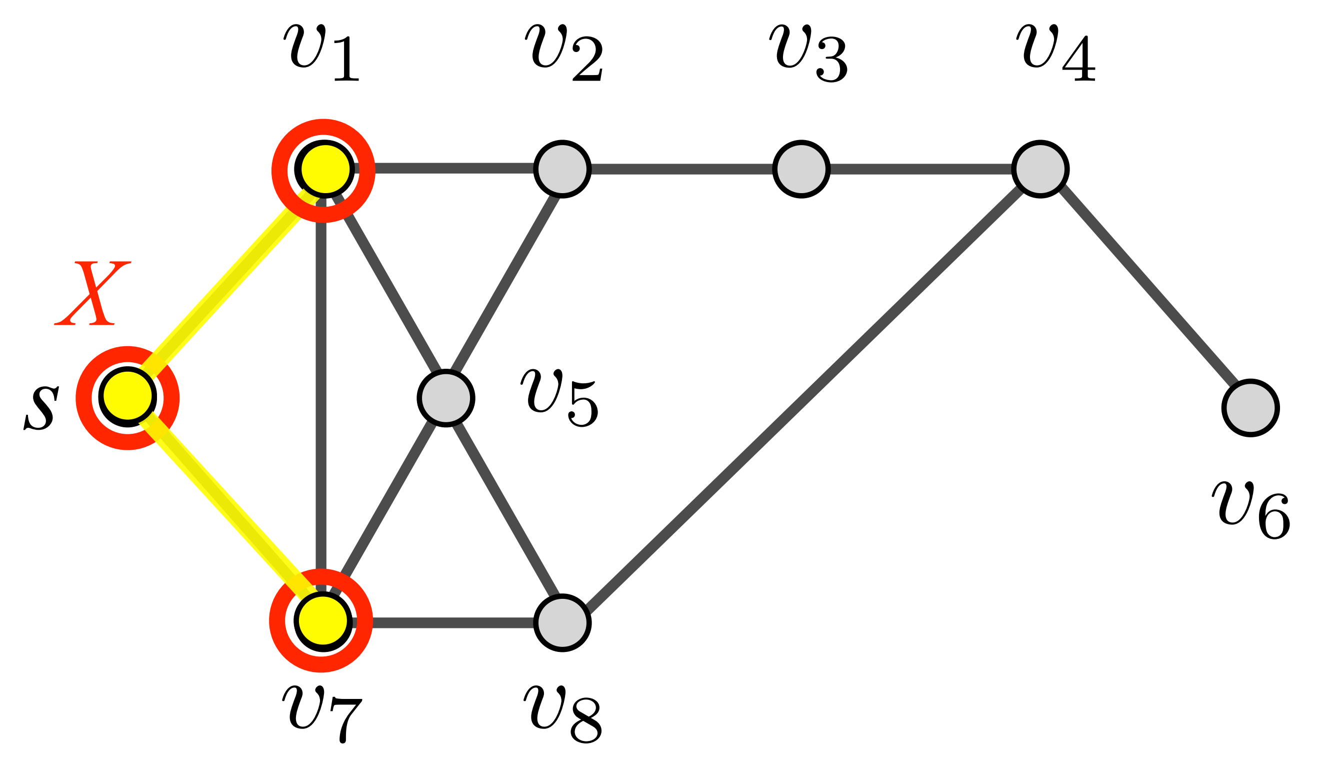
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v ∈ R
    - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w} ∈ E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

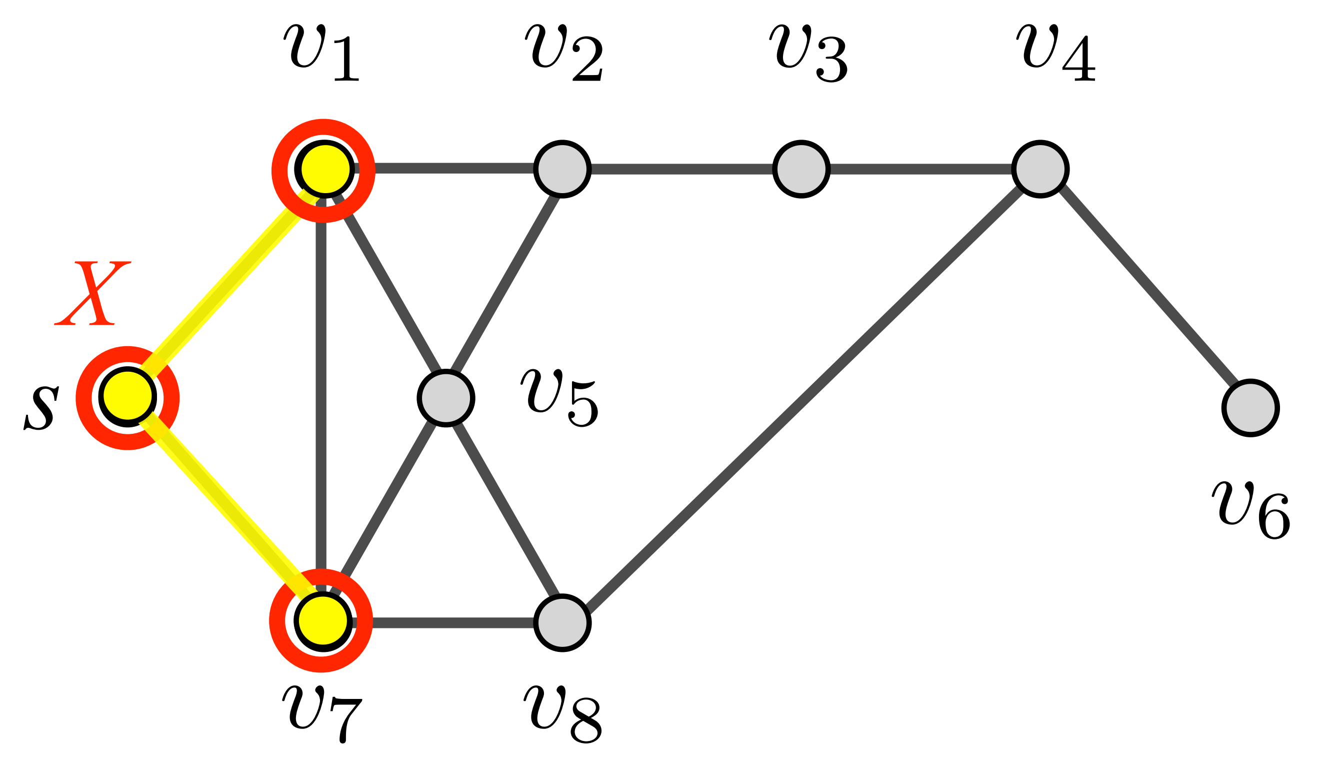
Algorithmus 3.7

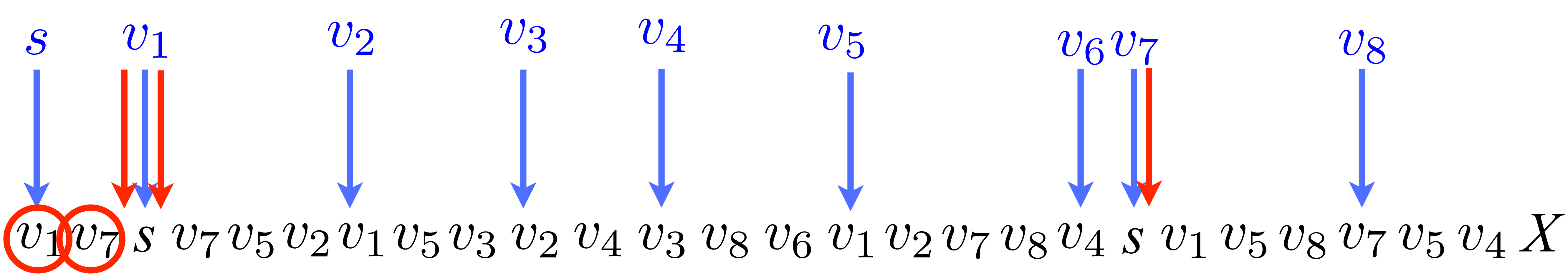
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP



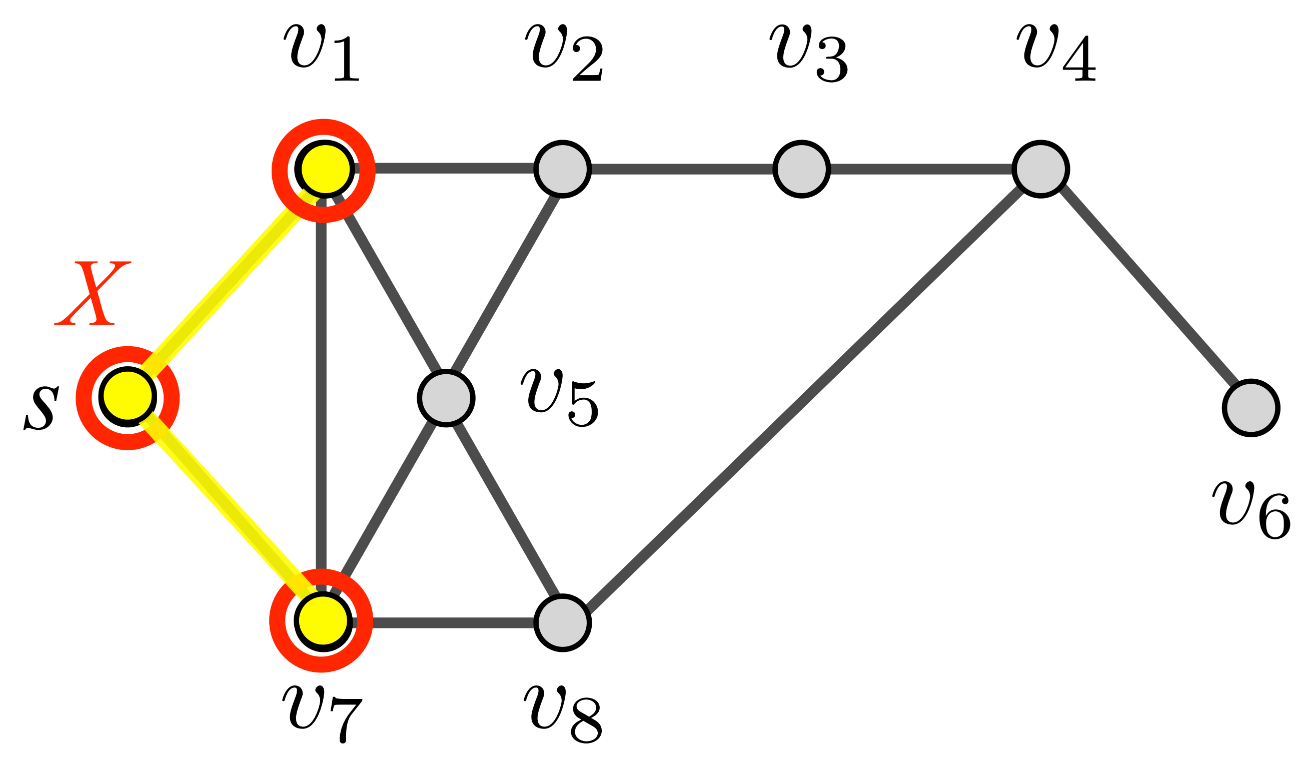


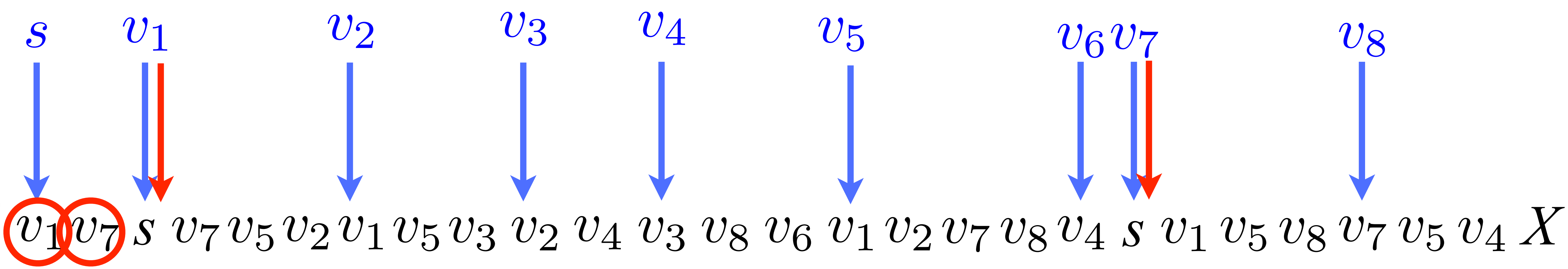
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP





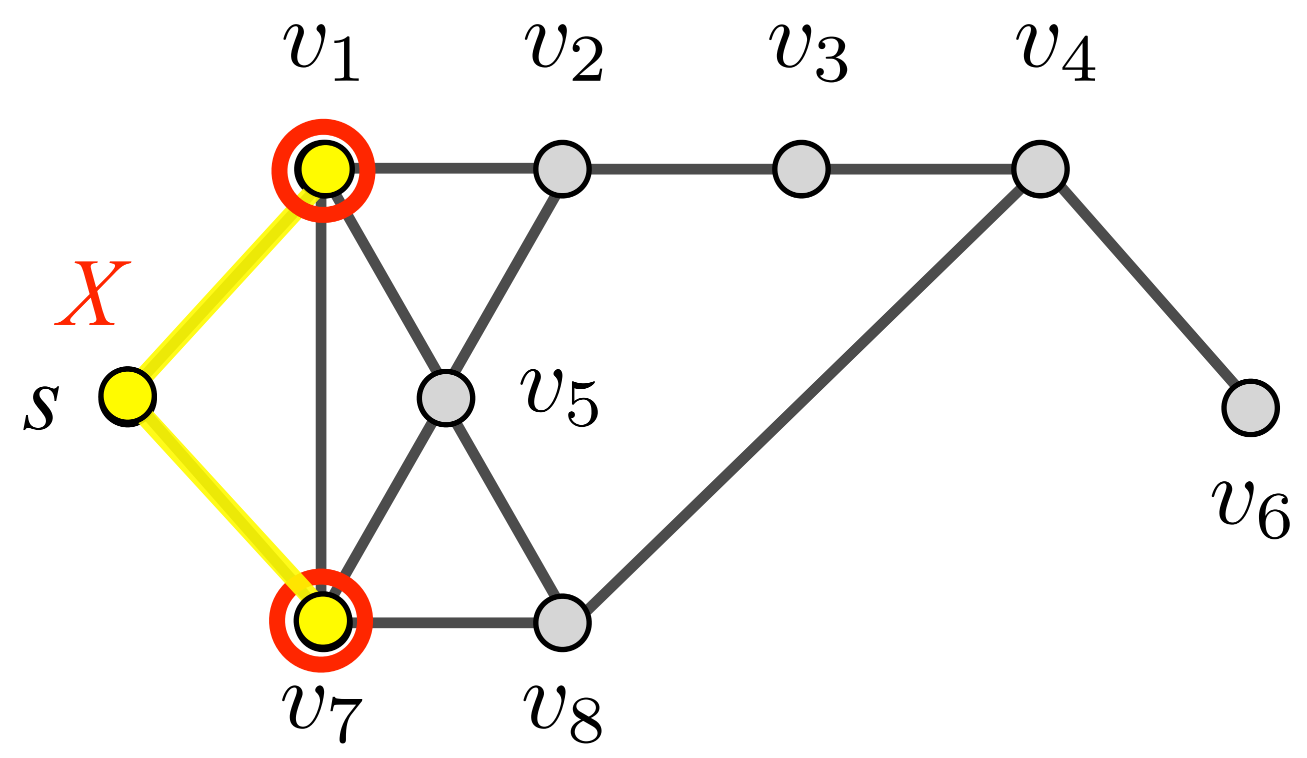
Algorithmus 3.7

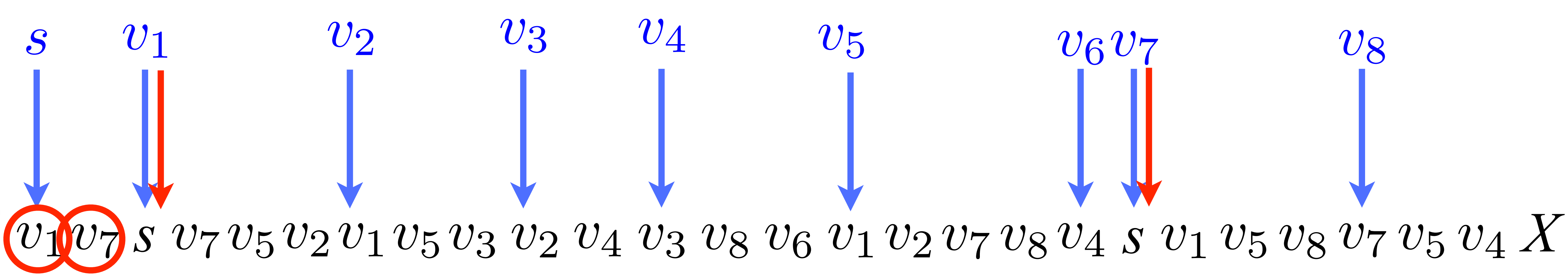
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

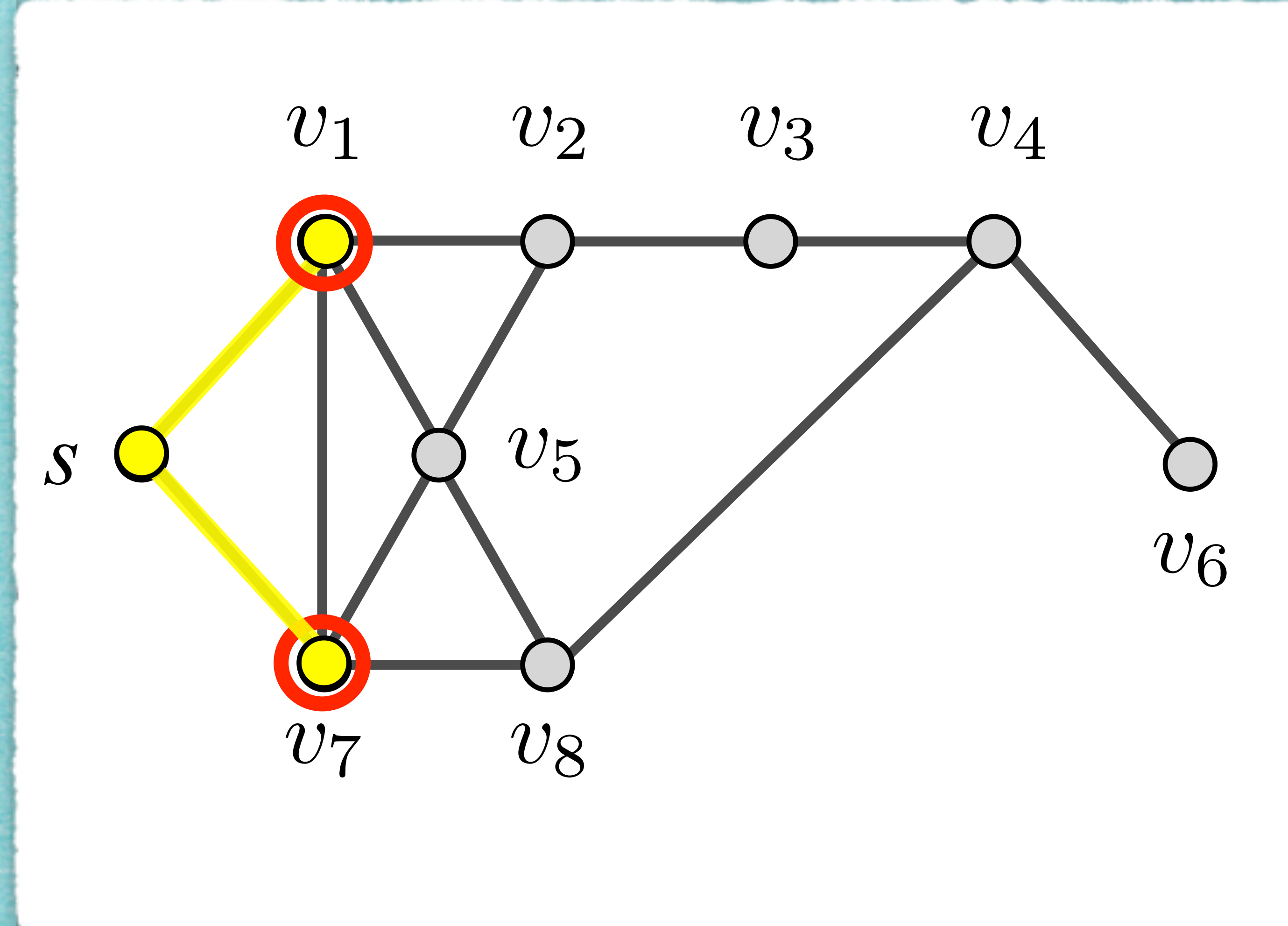
INPUT: Graph $G = (V,E)$, Knoten s

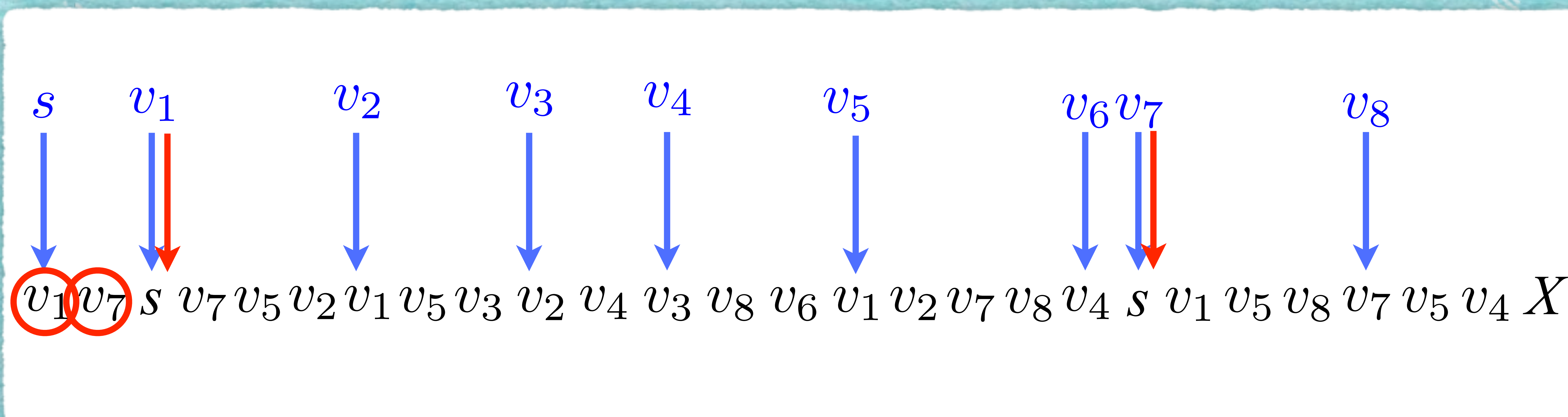
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }

   }
3. STOP





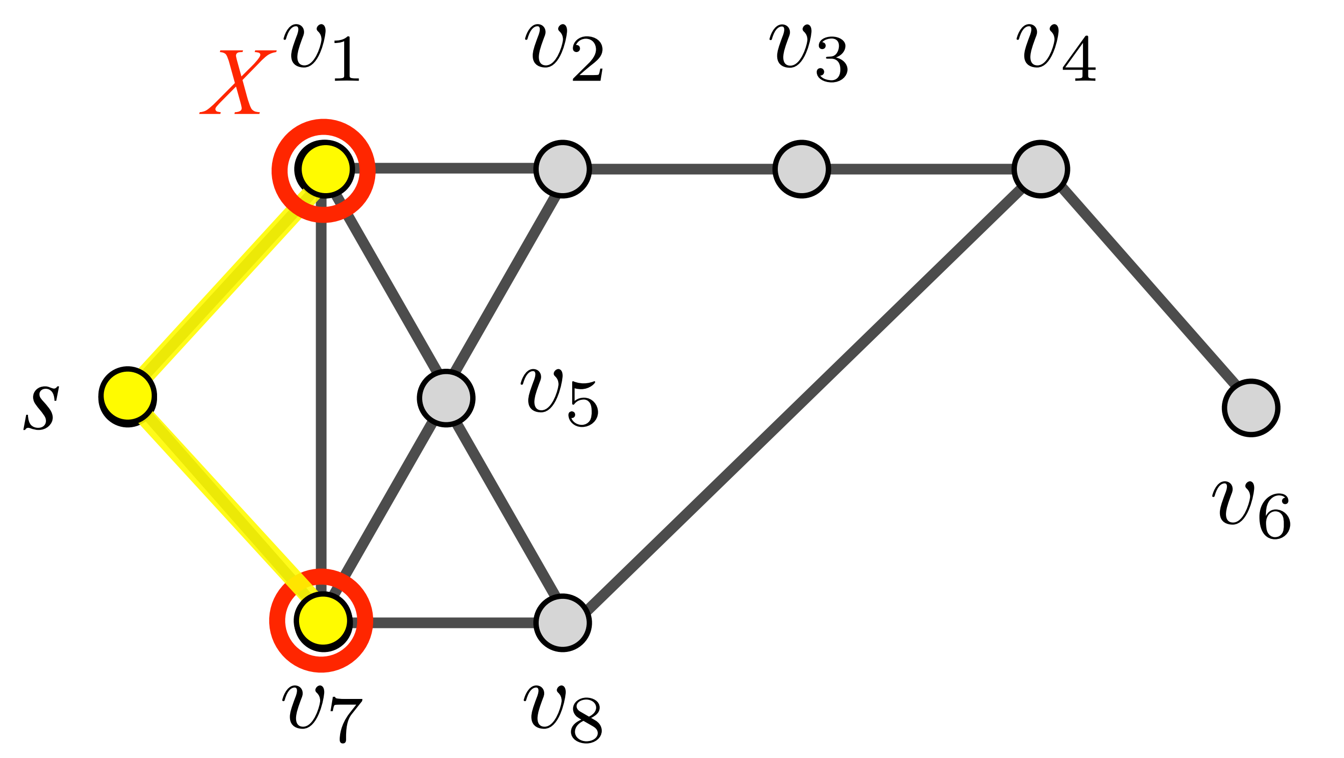
Algorithmus 3.7

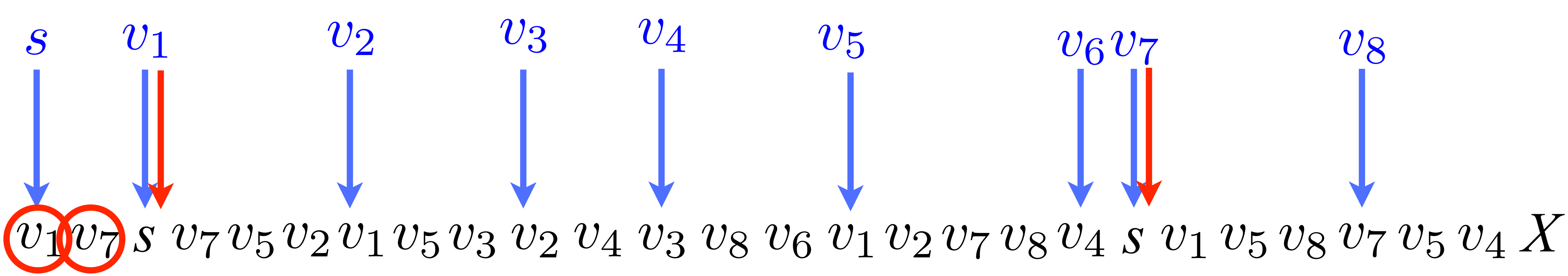
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





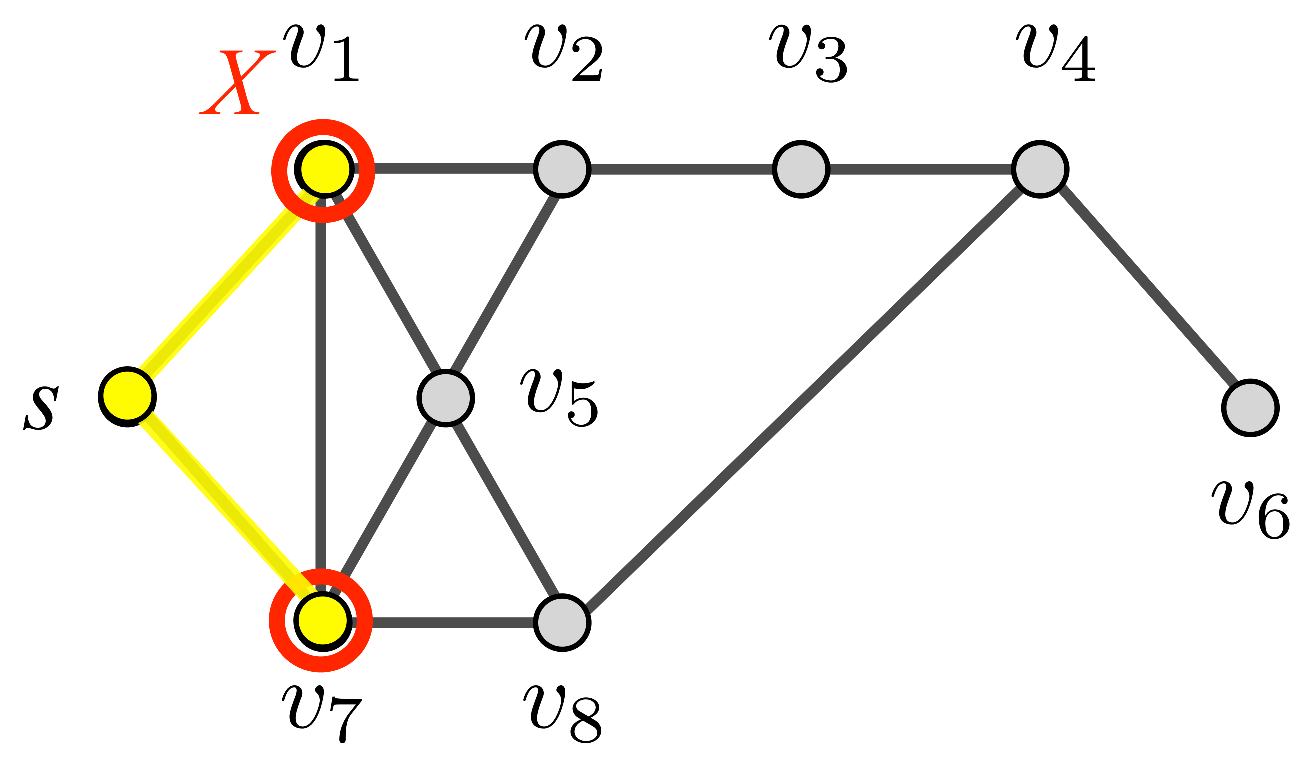
Algorithmus 3.7

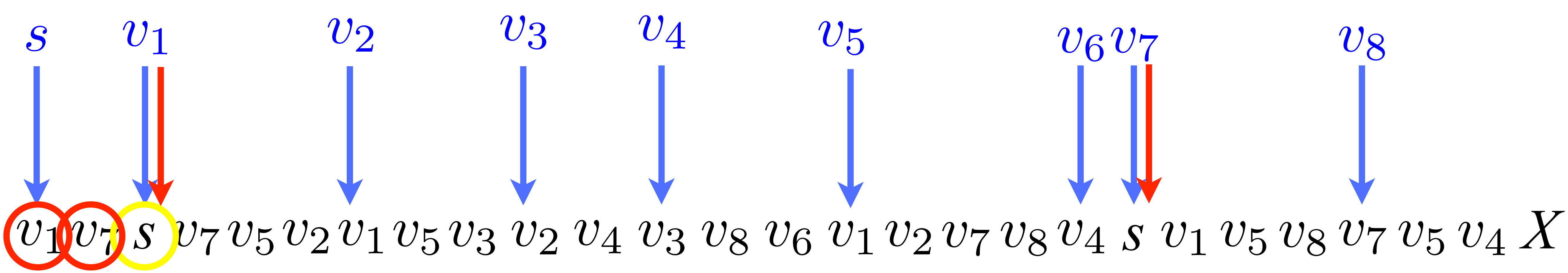
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

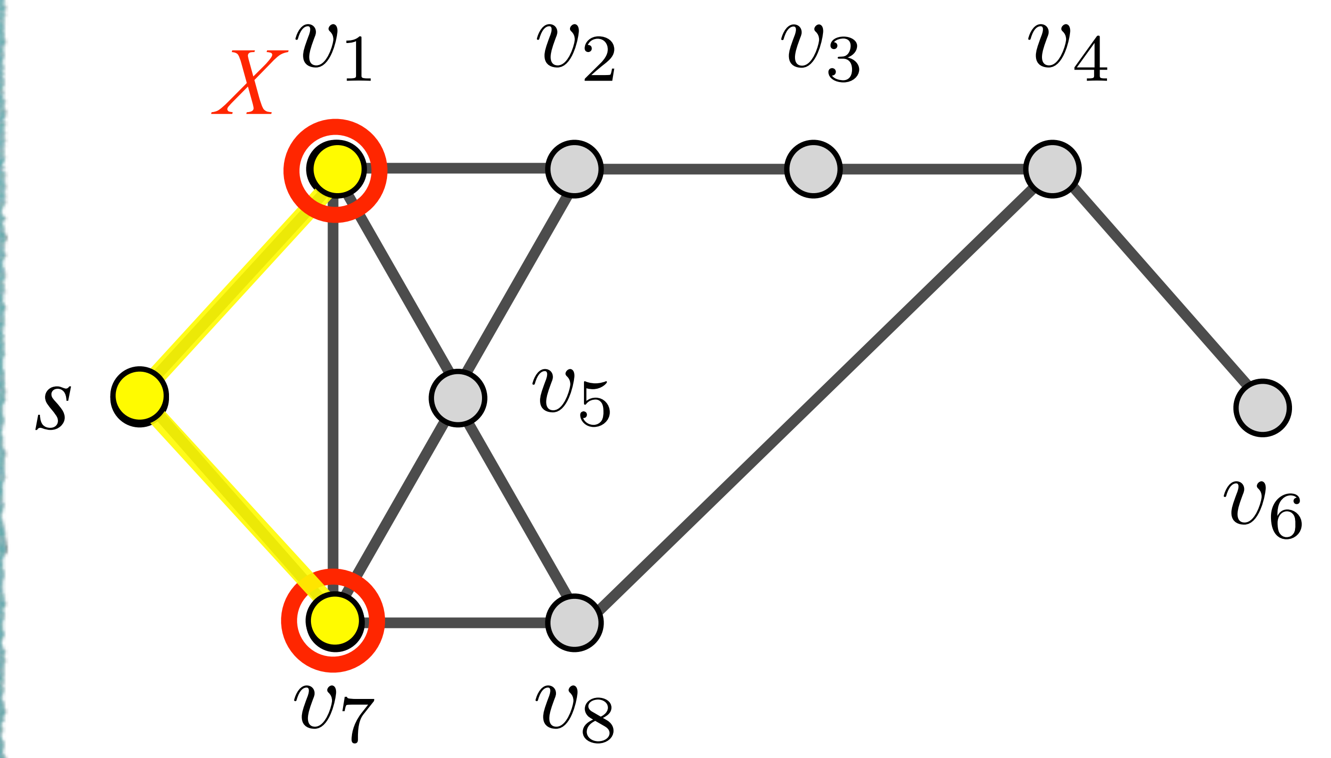
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

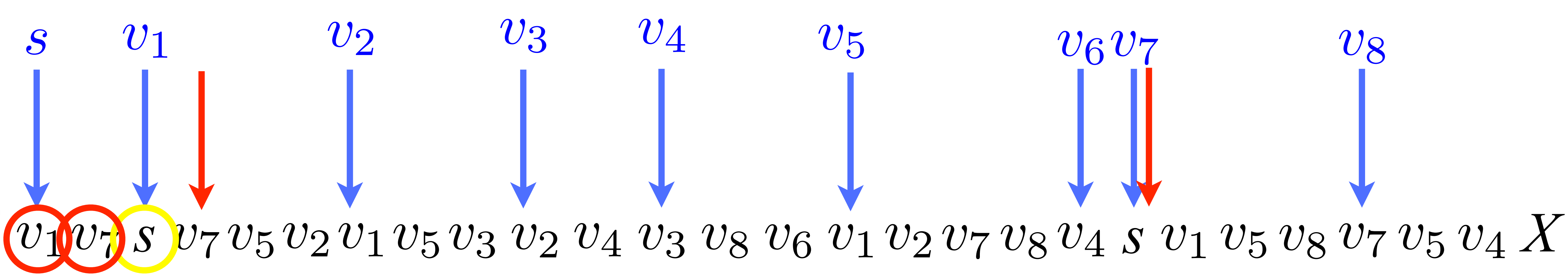
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP

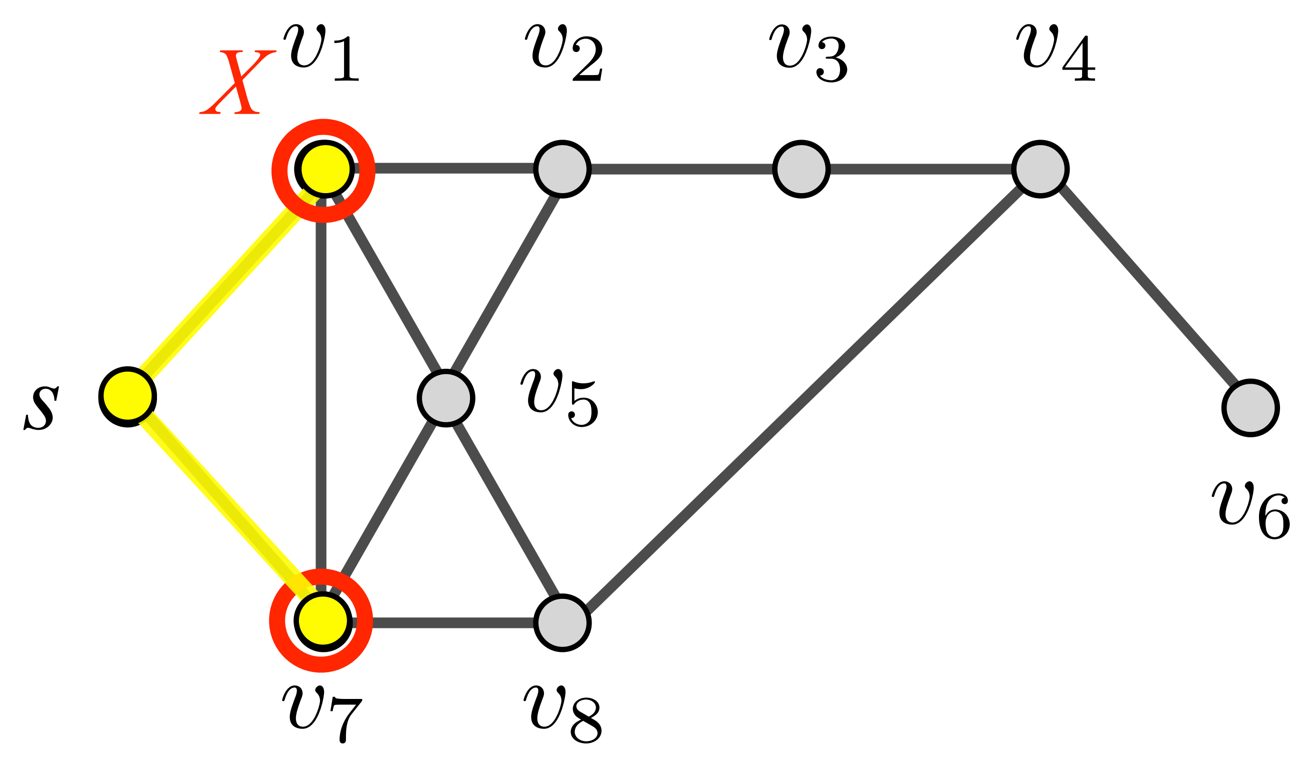


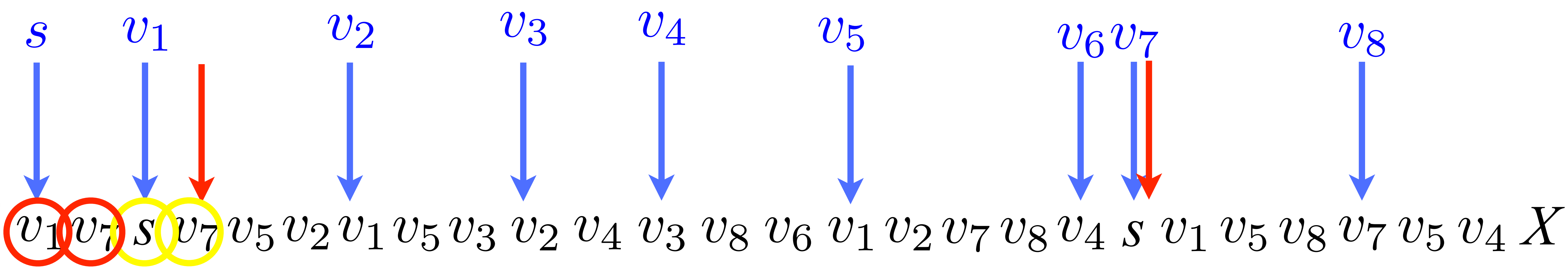


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈ R
   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
      2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
      }
   }
3. STOP





Algorithmus 3.7

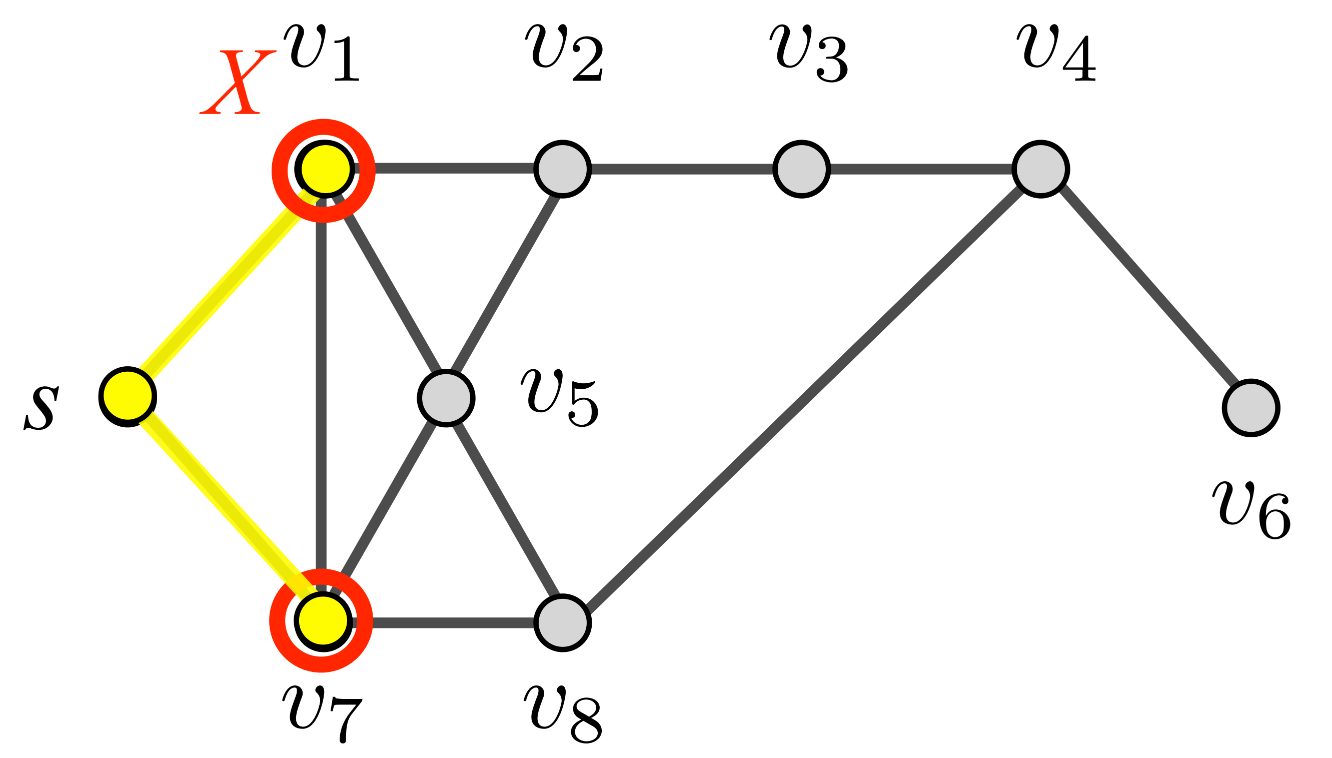
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

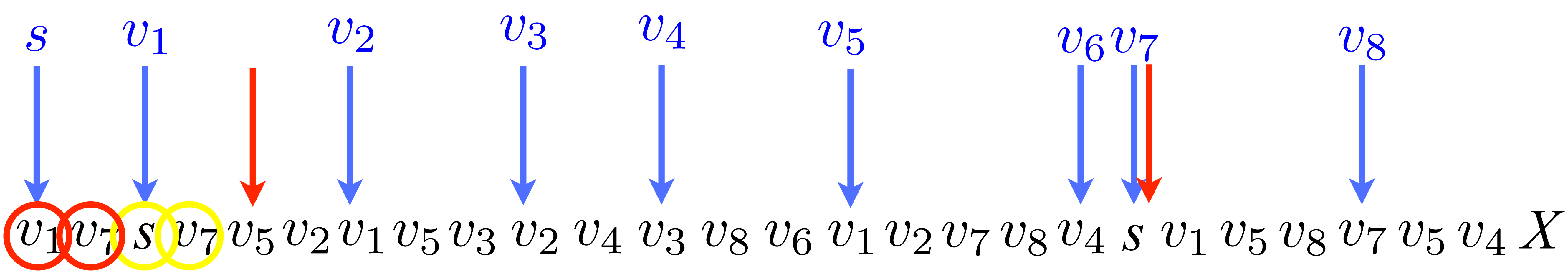
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP

2





Algorithmus 3.7

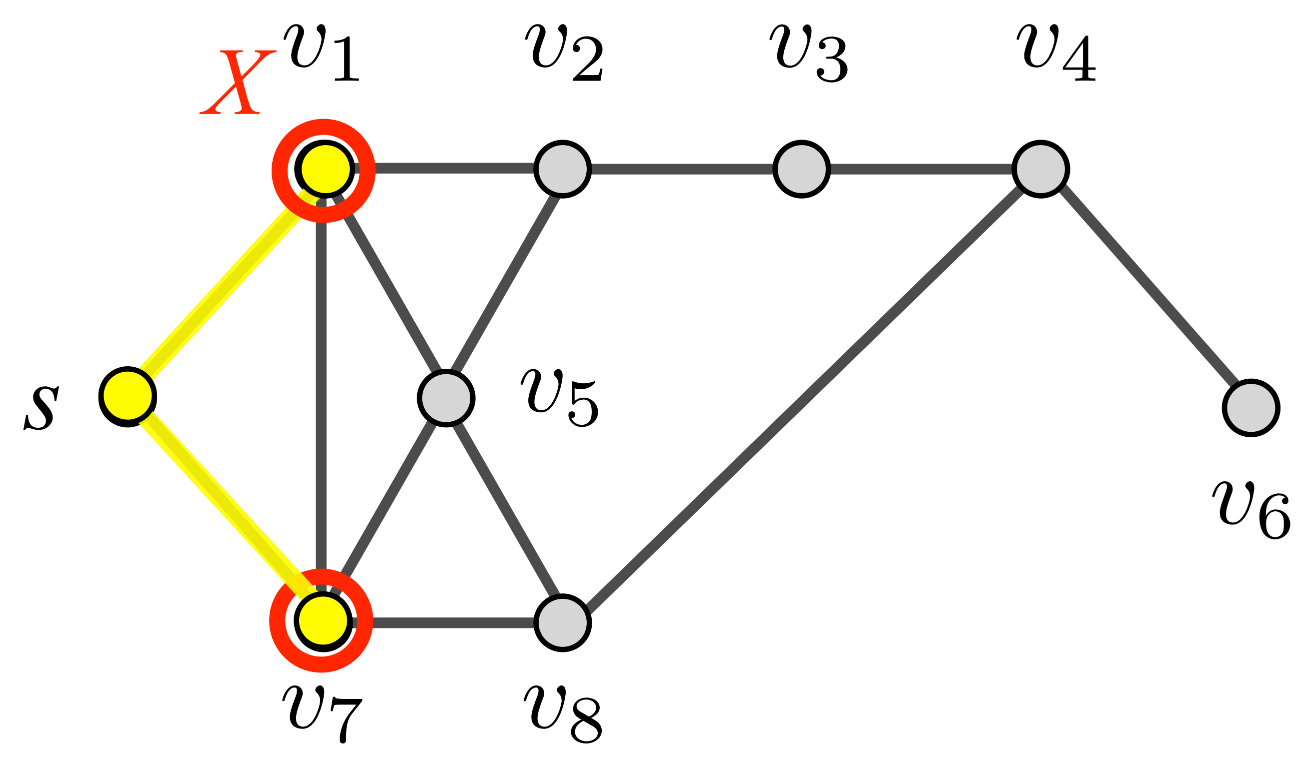
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

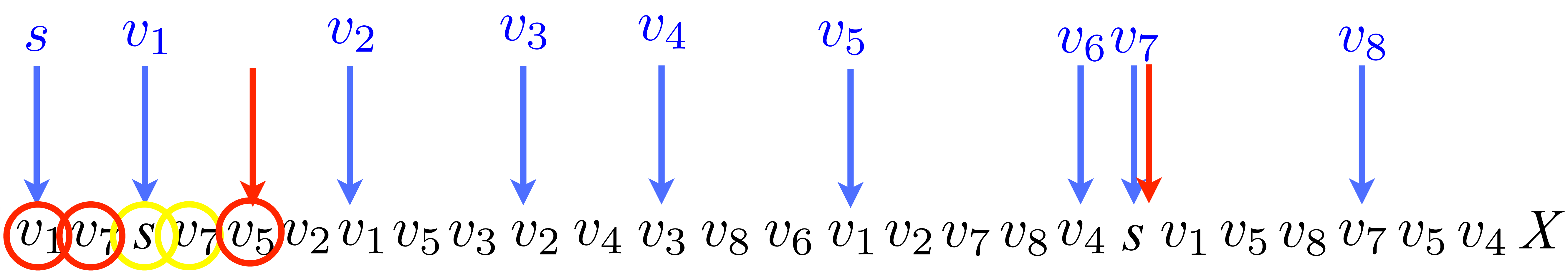
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP

2





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

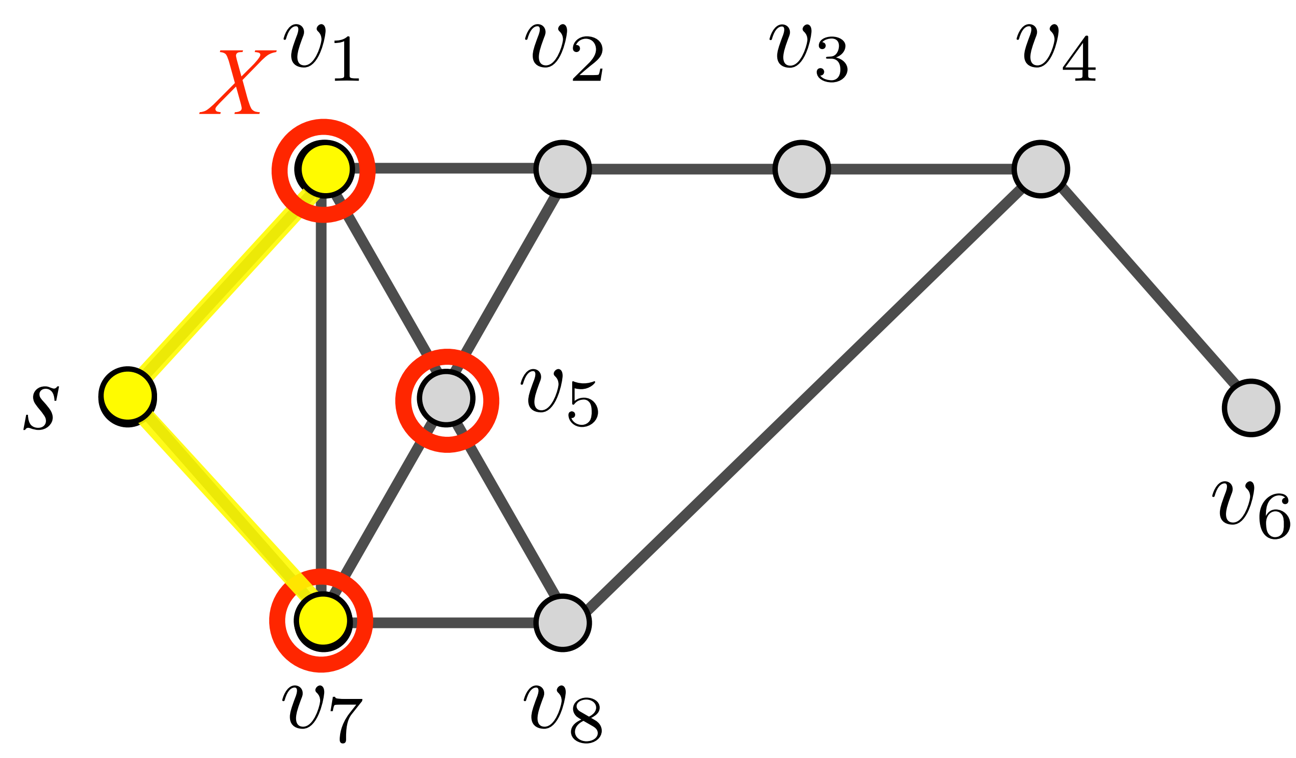
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

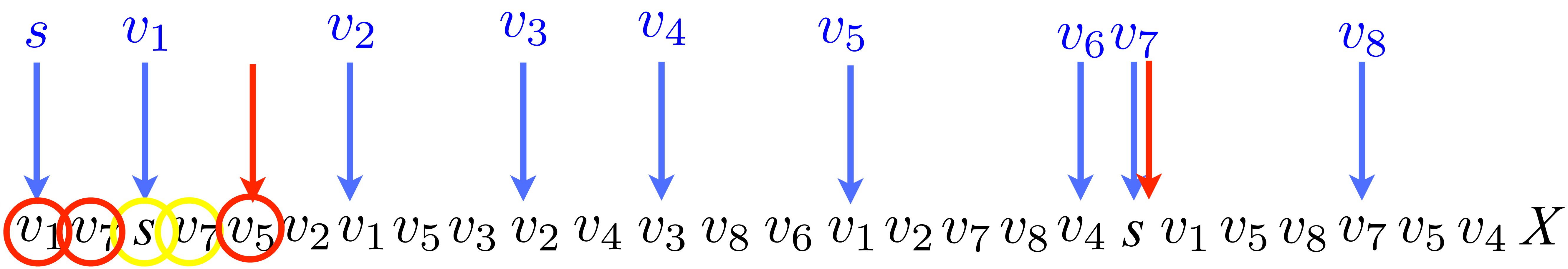
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

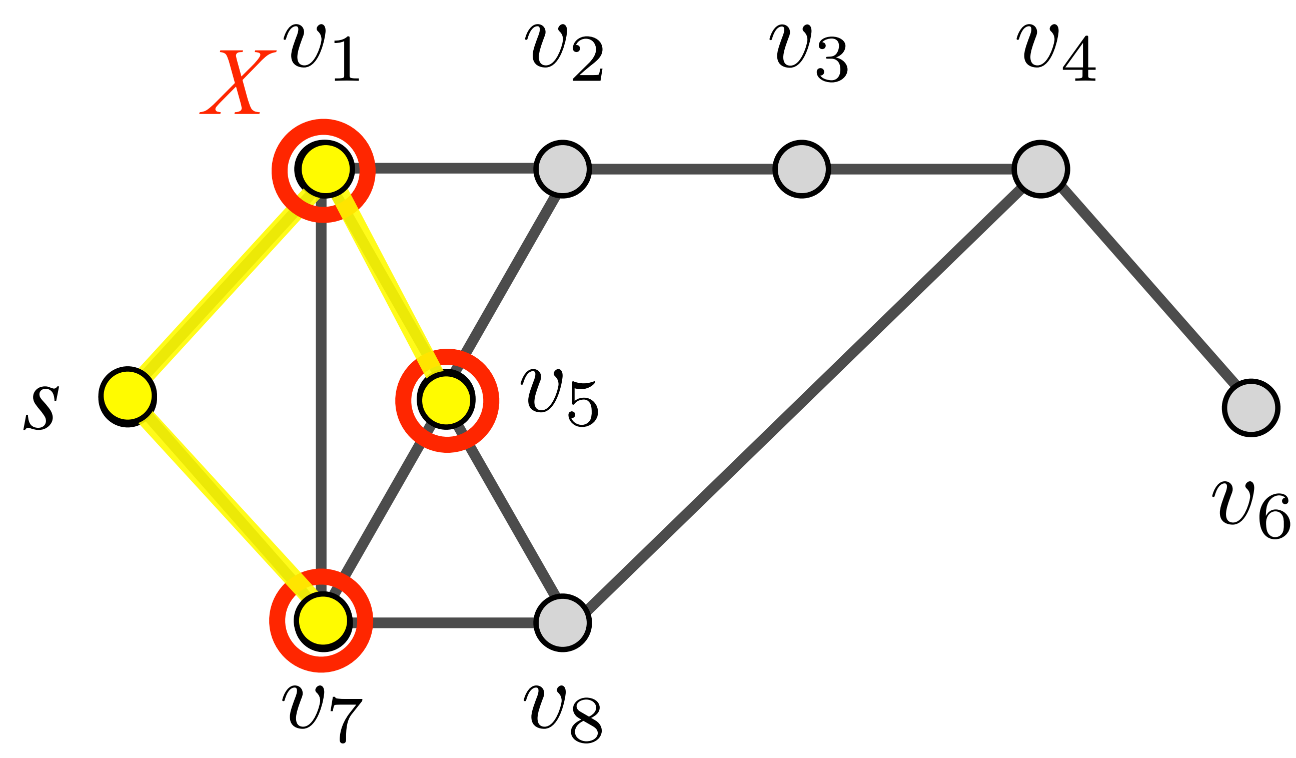
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

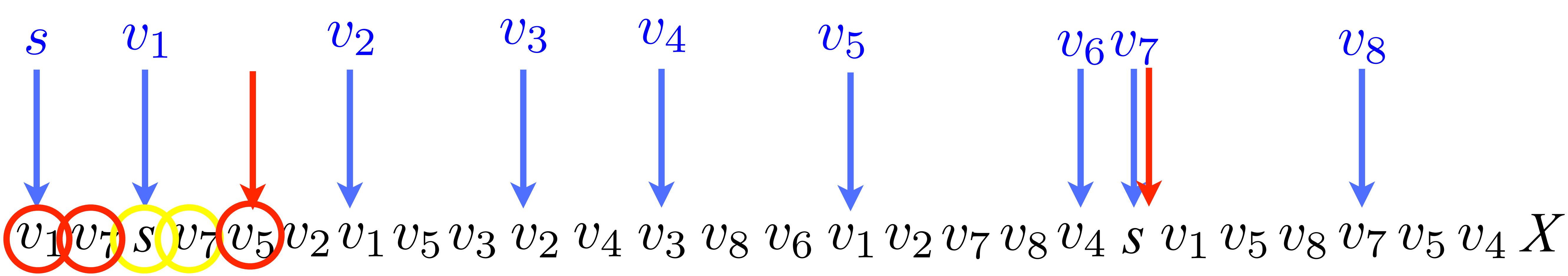
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

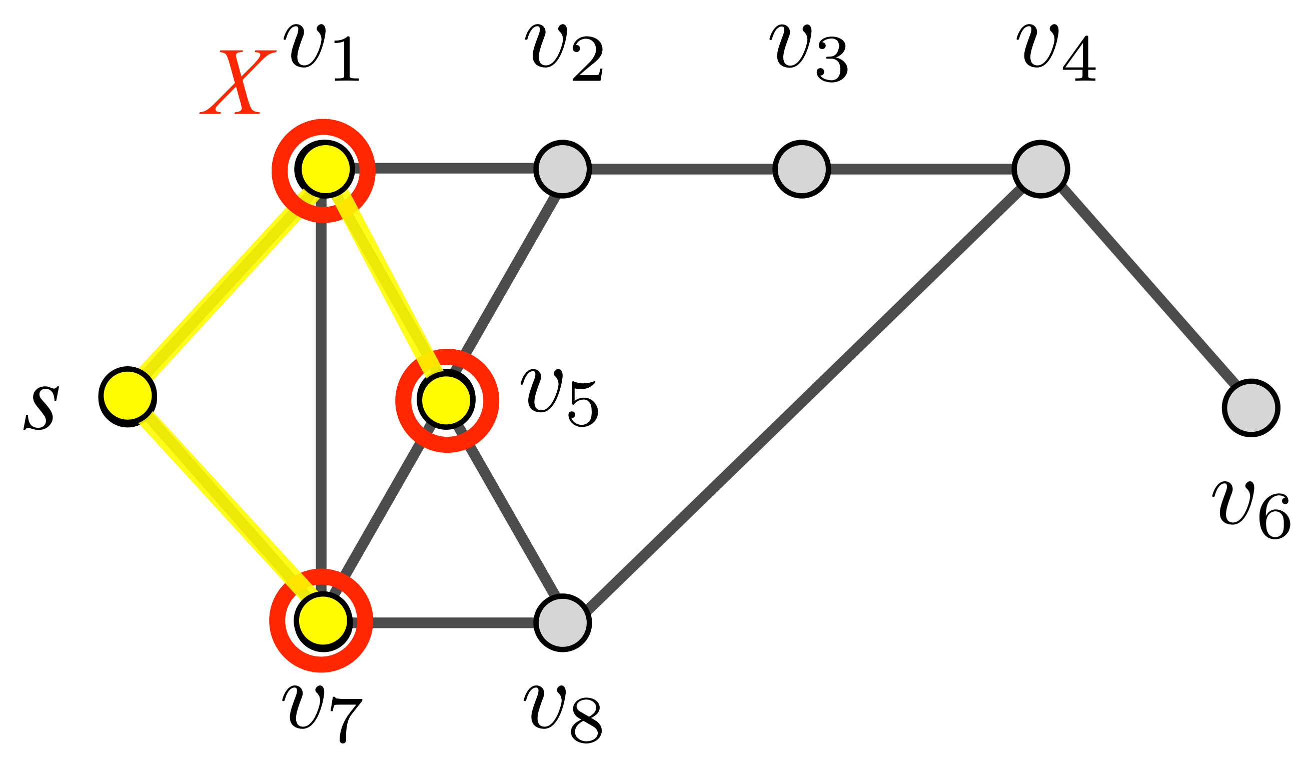
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

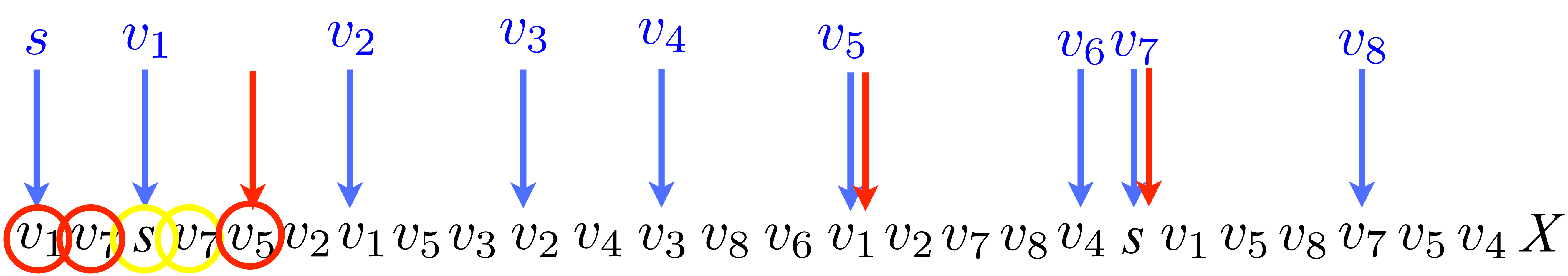
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

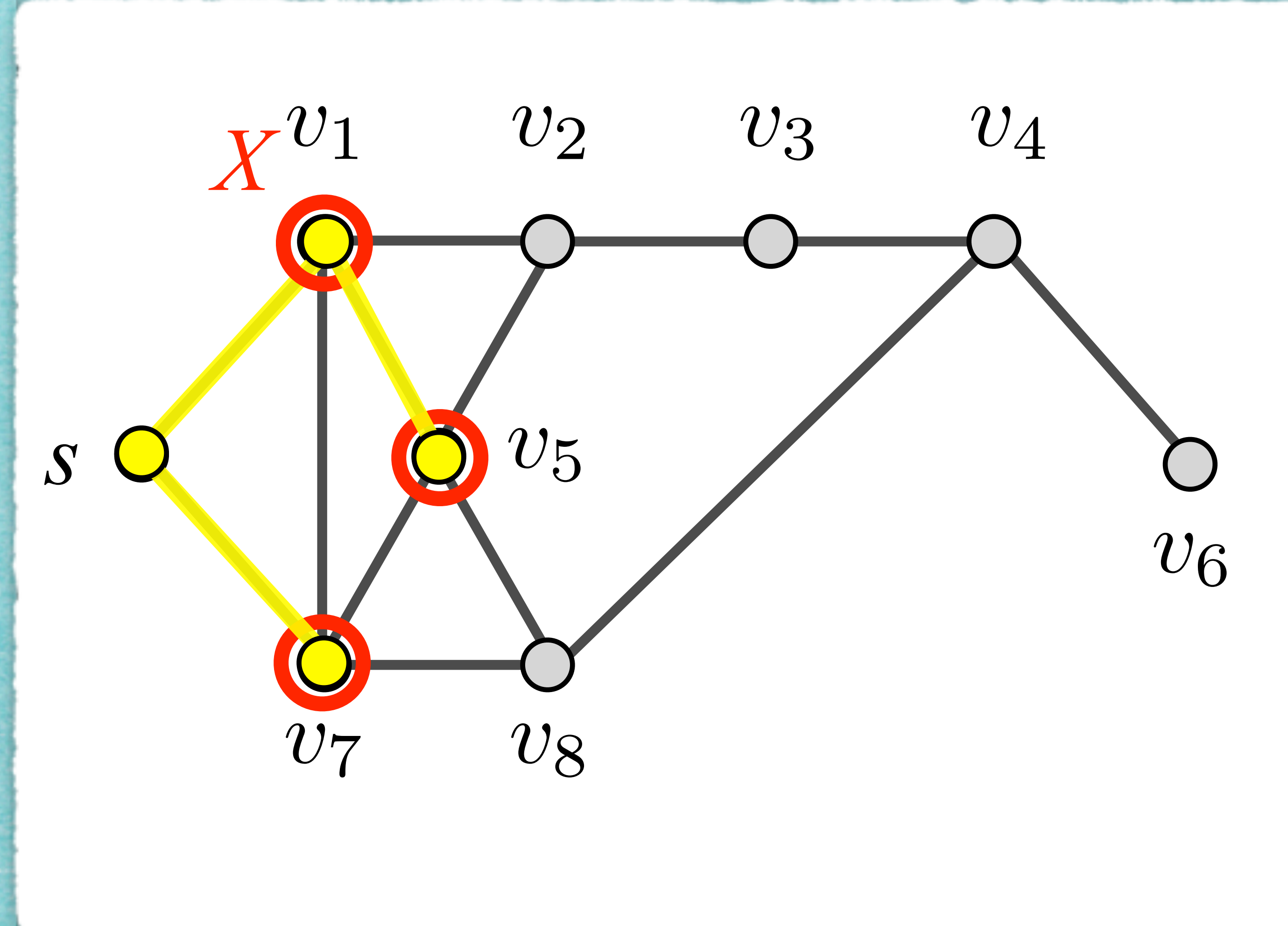
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

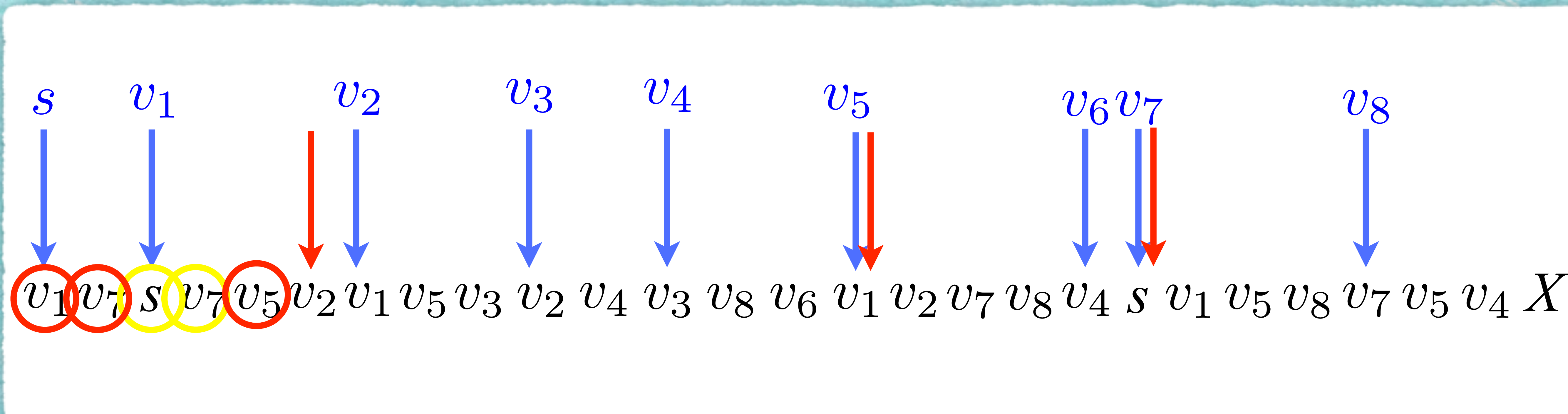
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

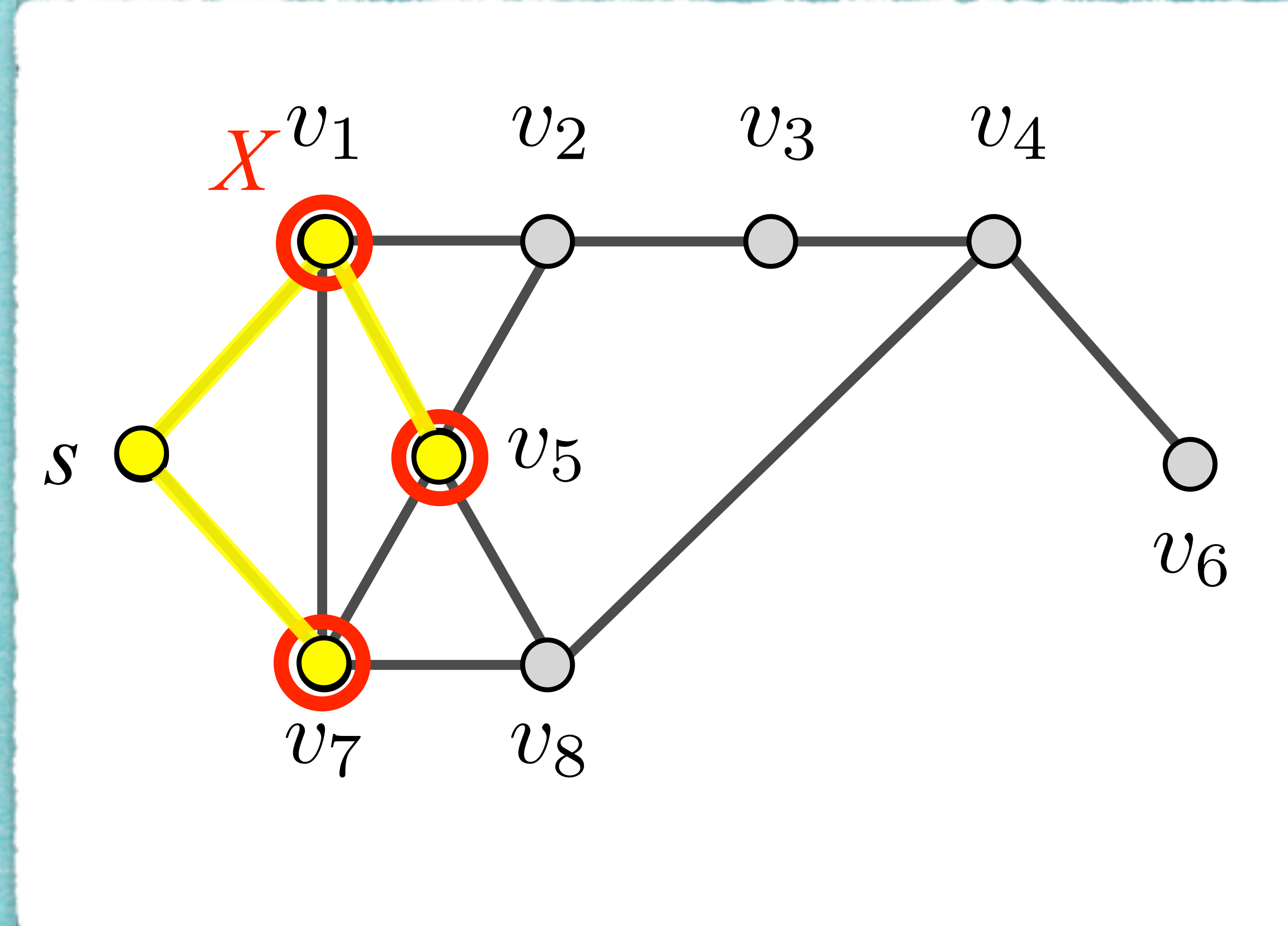
INPUT: Graph G = (V,E), Knoten s

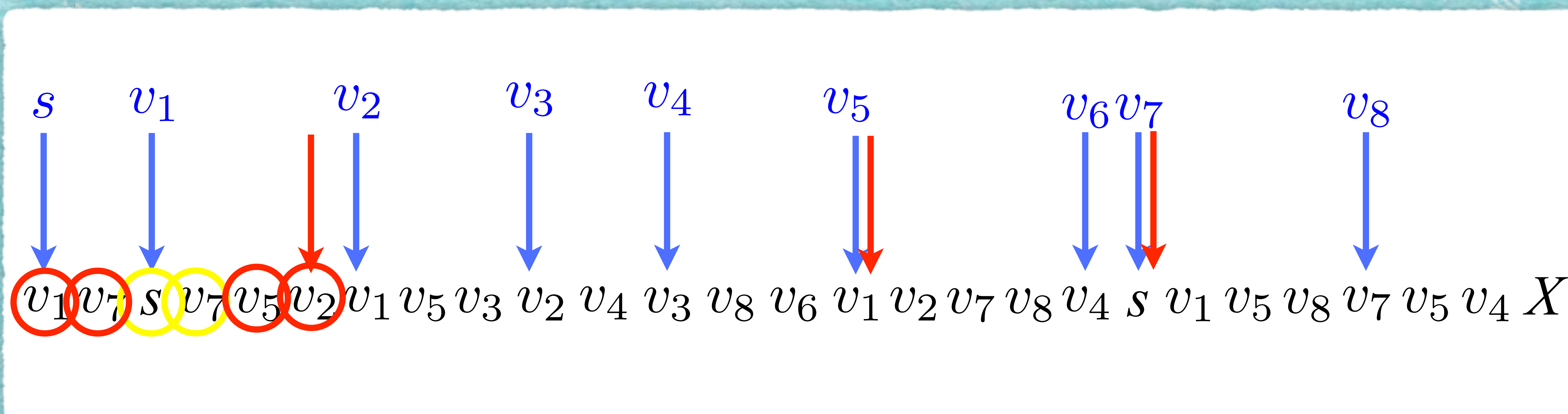
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





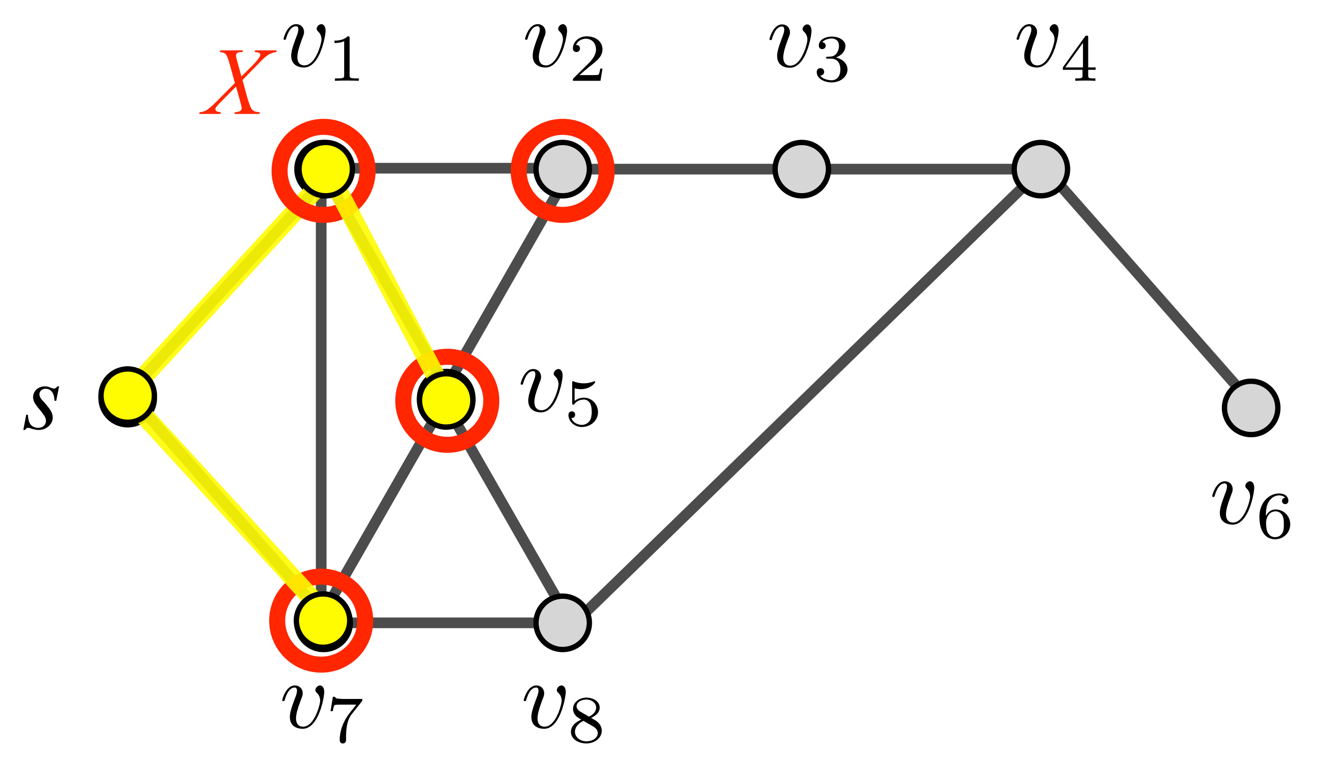
Algorithmus 3.7

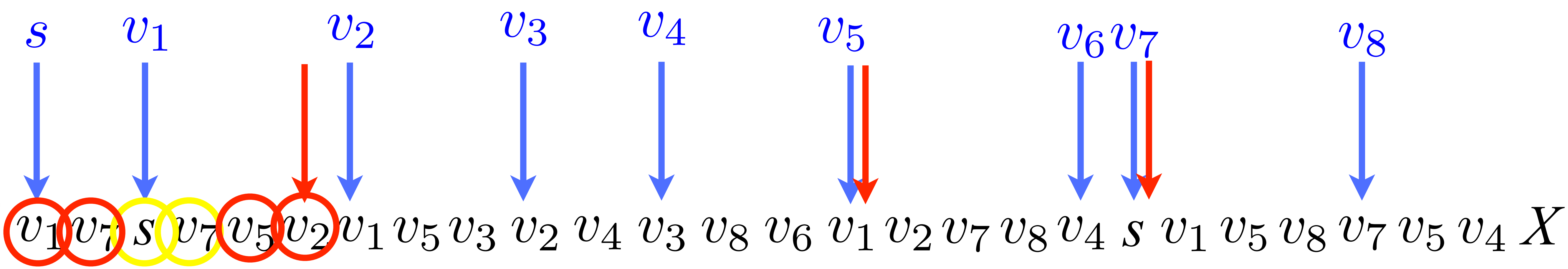
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
    }
}
3. STOP





Algorithmus 3.7

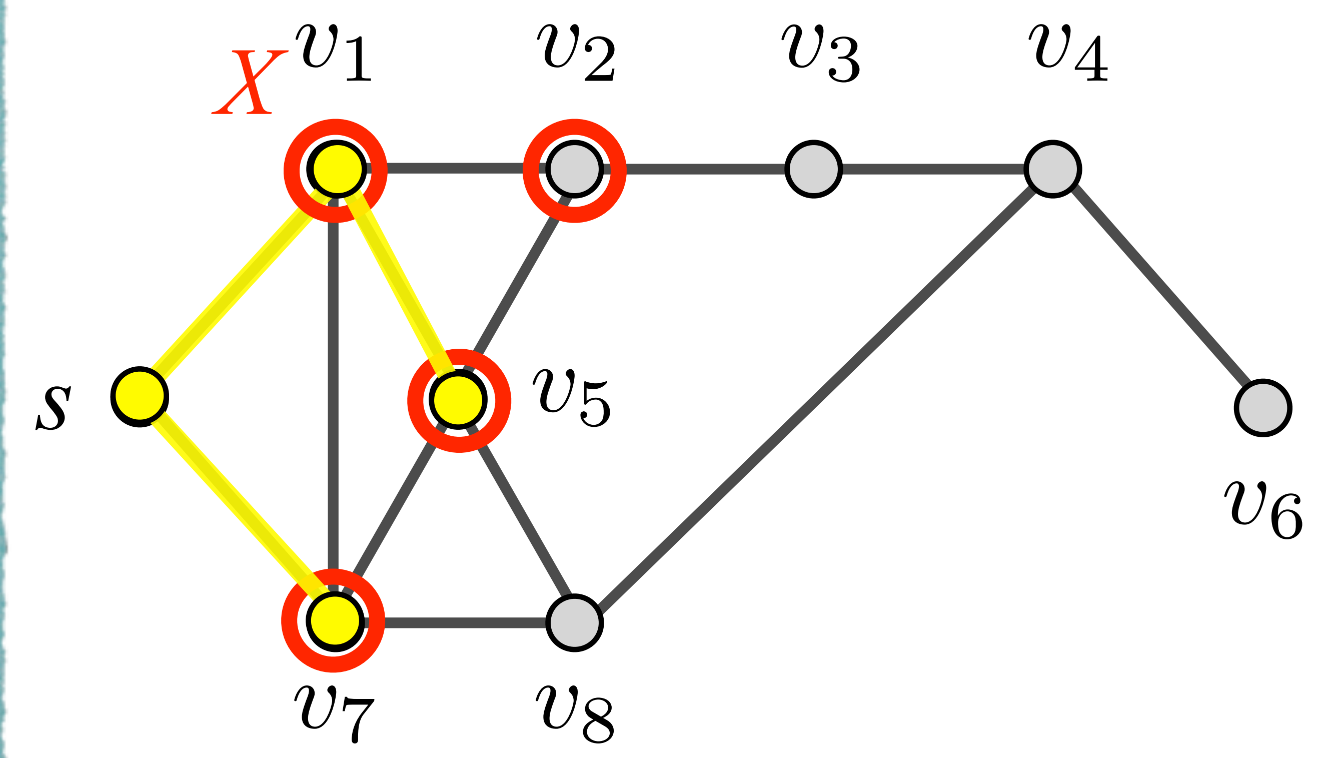
INPUT: Graph G = (V,E), Knoten s

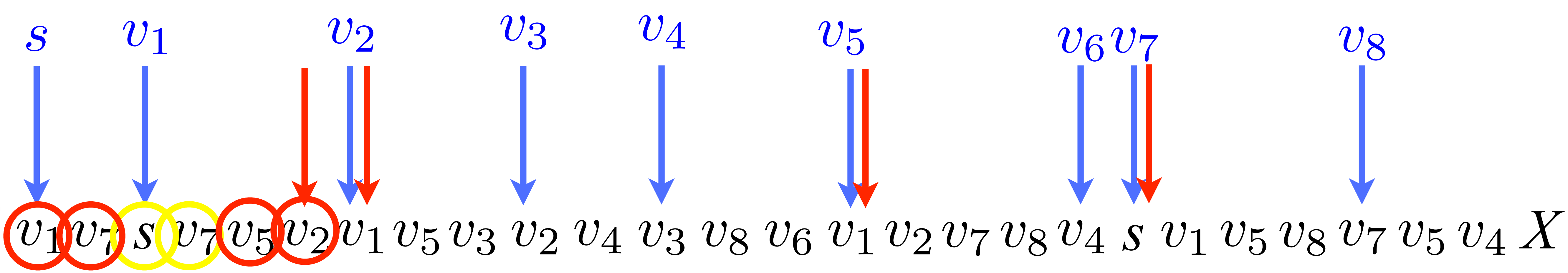
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
    }
}
3. STOP

2





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

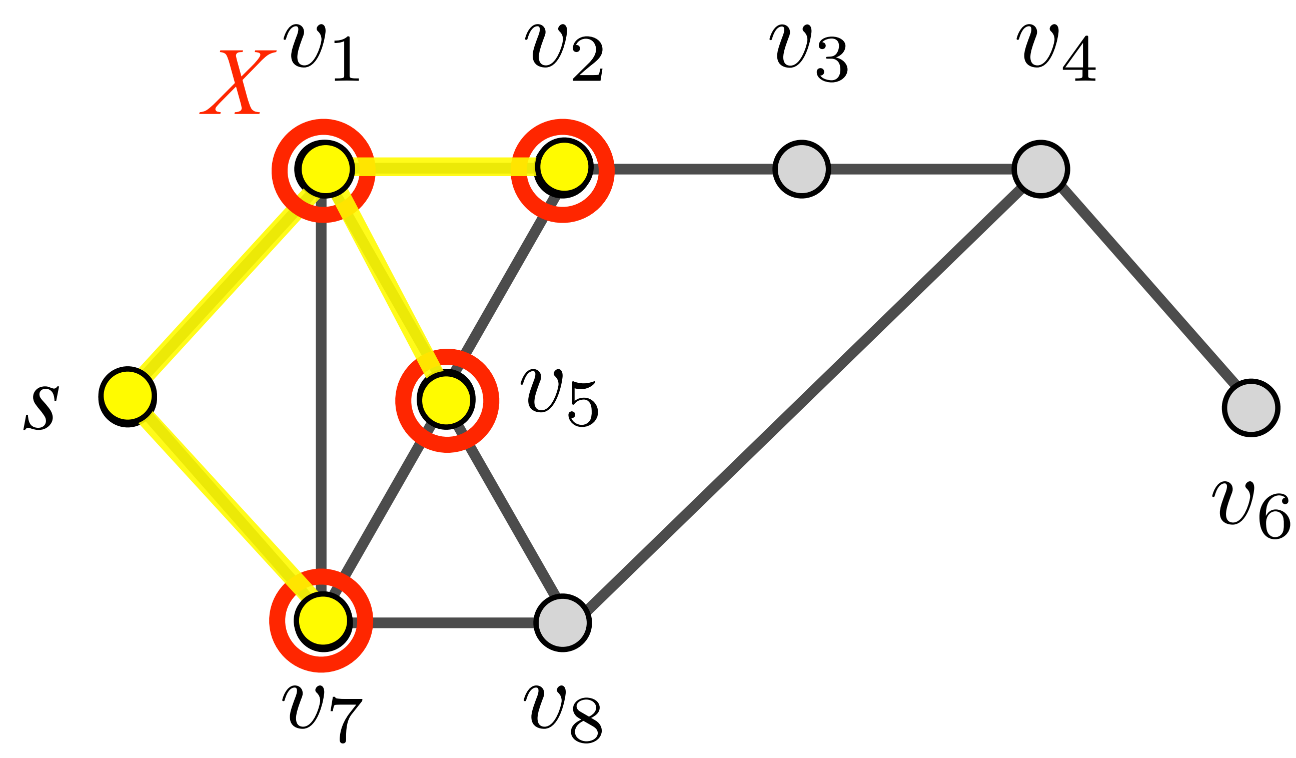
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

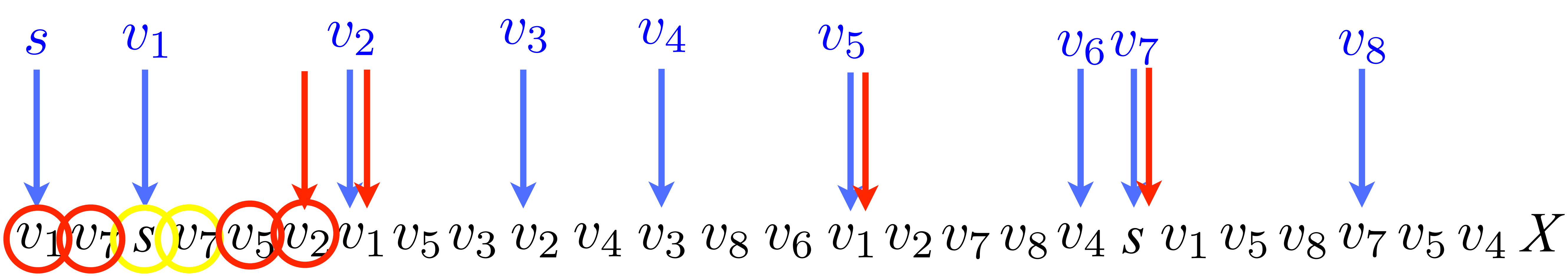
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

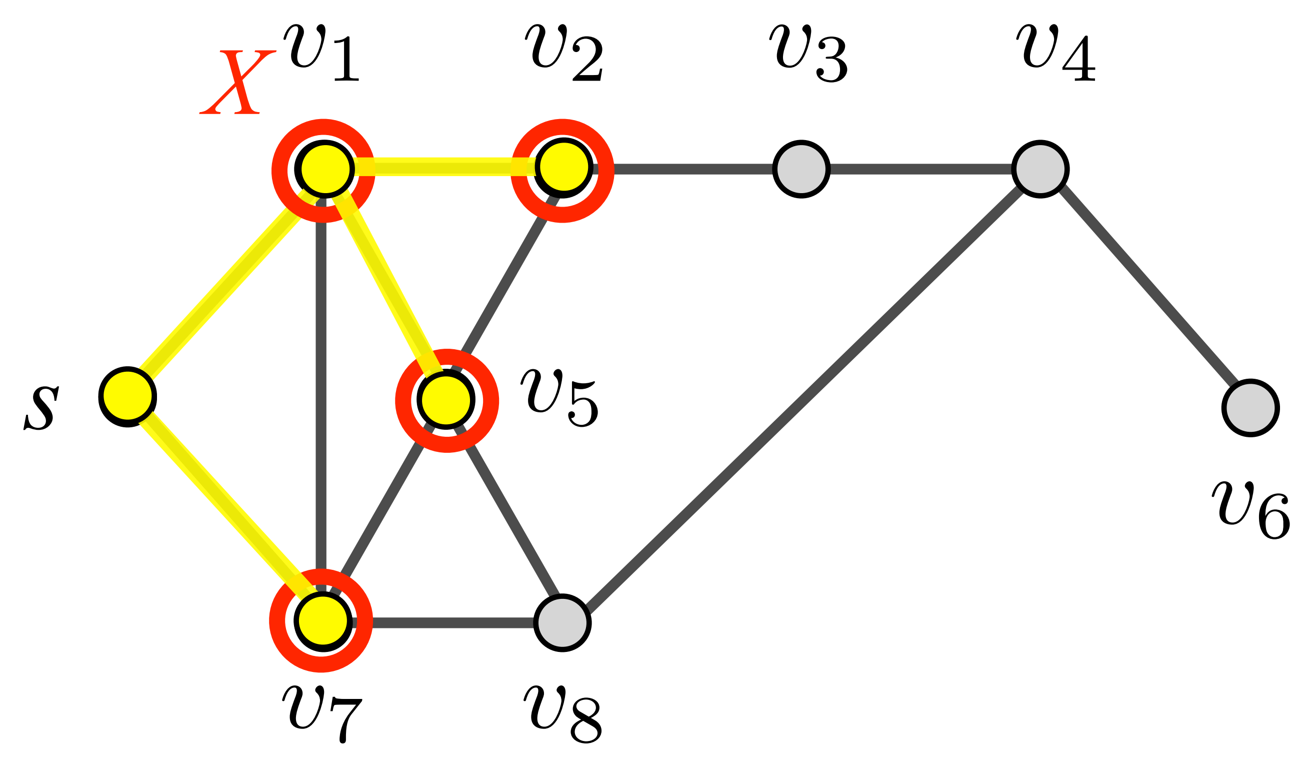
INPUT: Graph G = (V,E), Knoten s

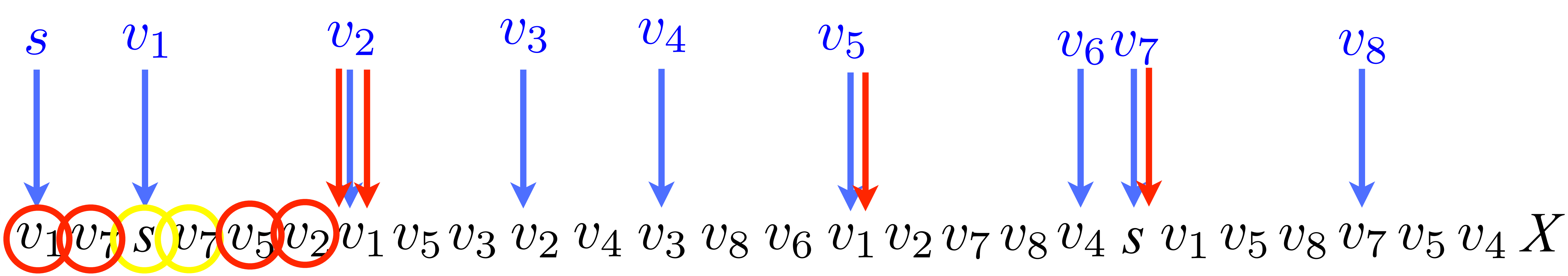
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





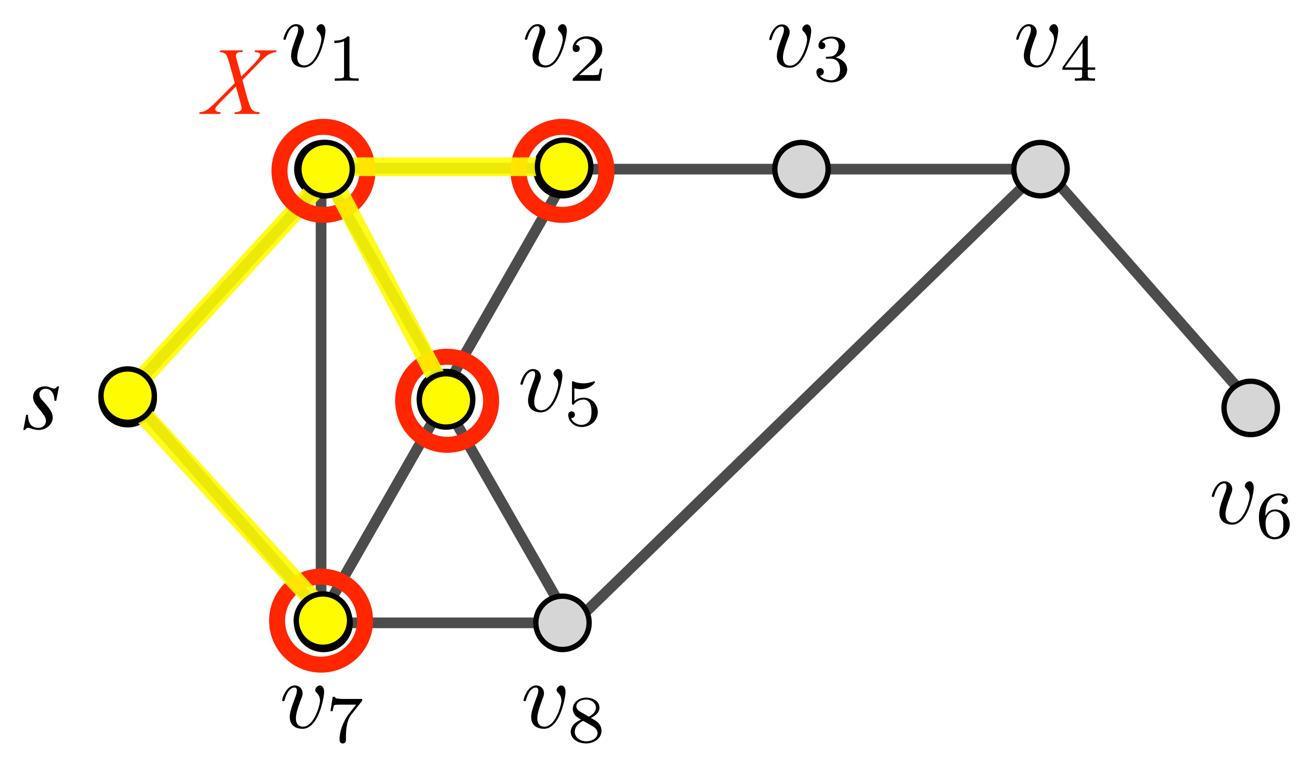
Algorithmus 3.7

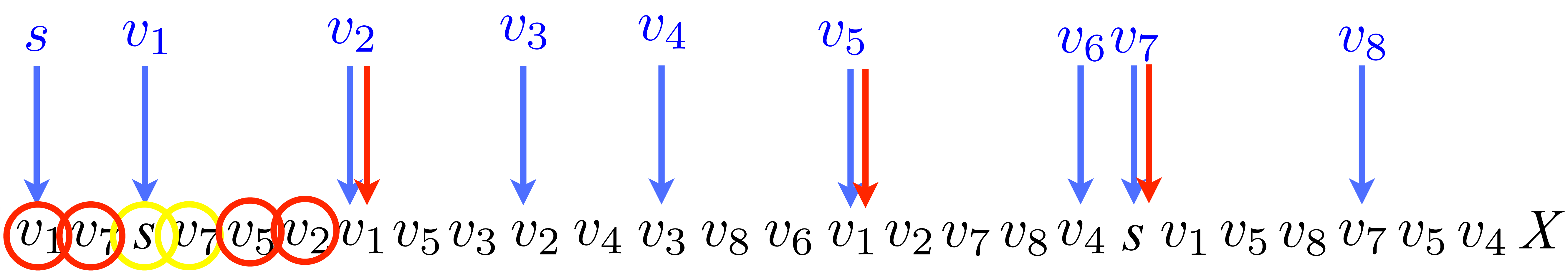
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

   }
3. STOP





Algorithmus 3.7

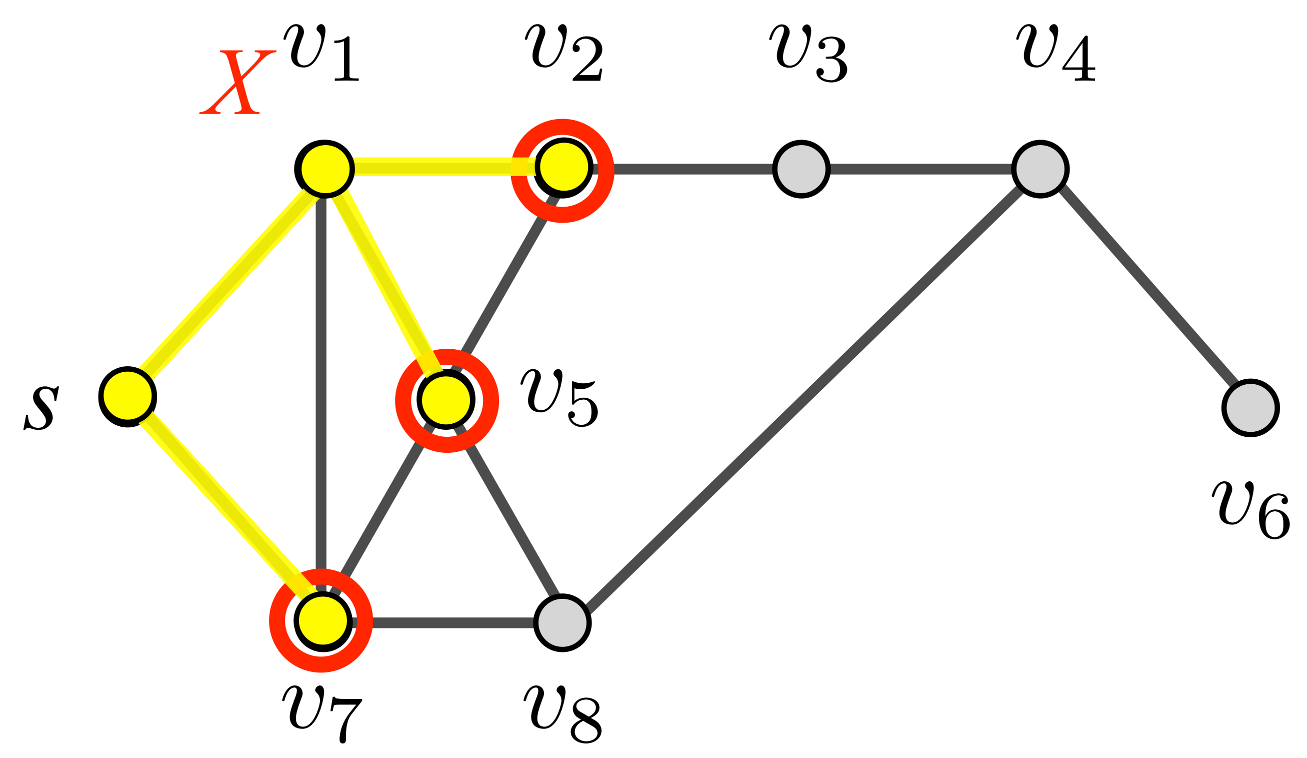
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

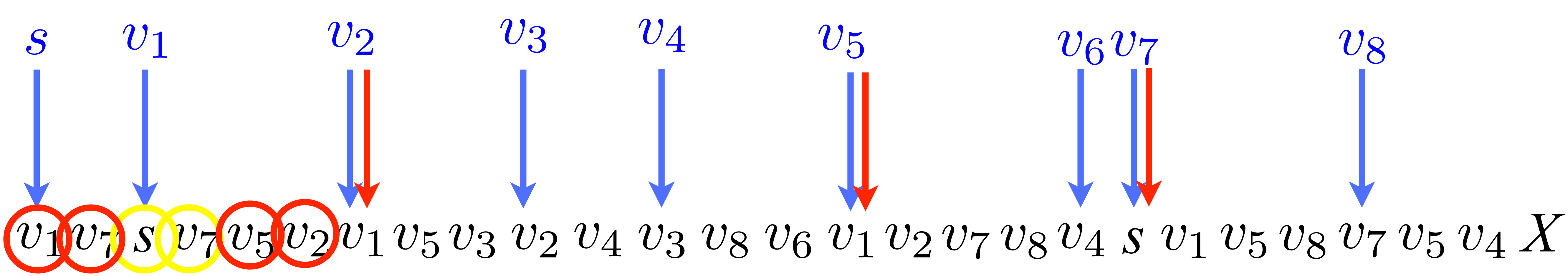
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2





Algorithmus 3.7

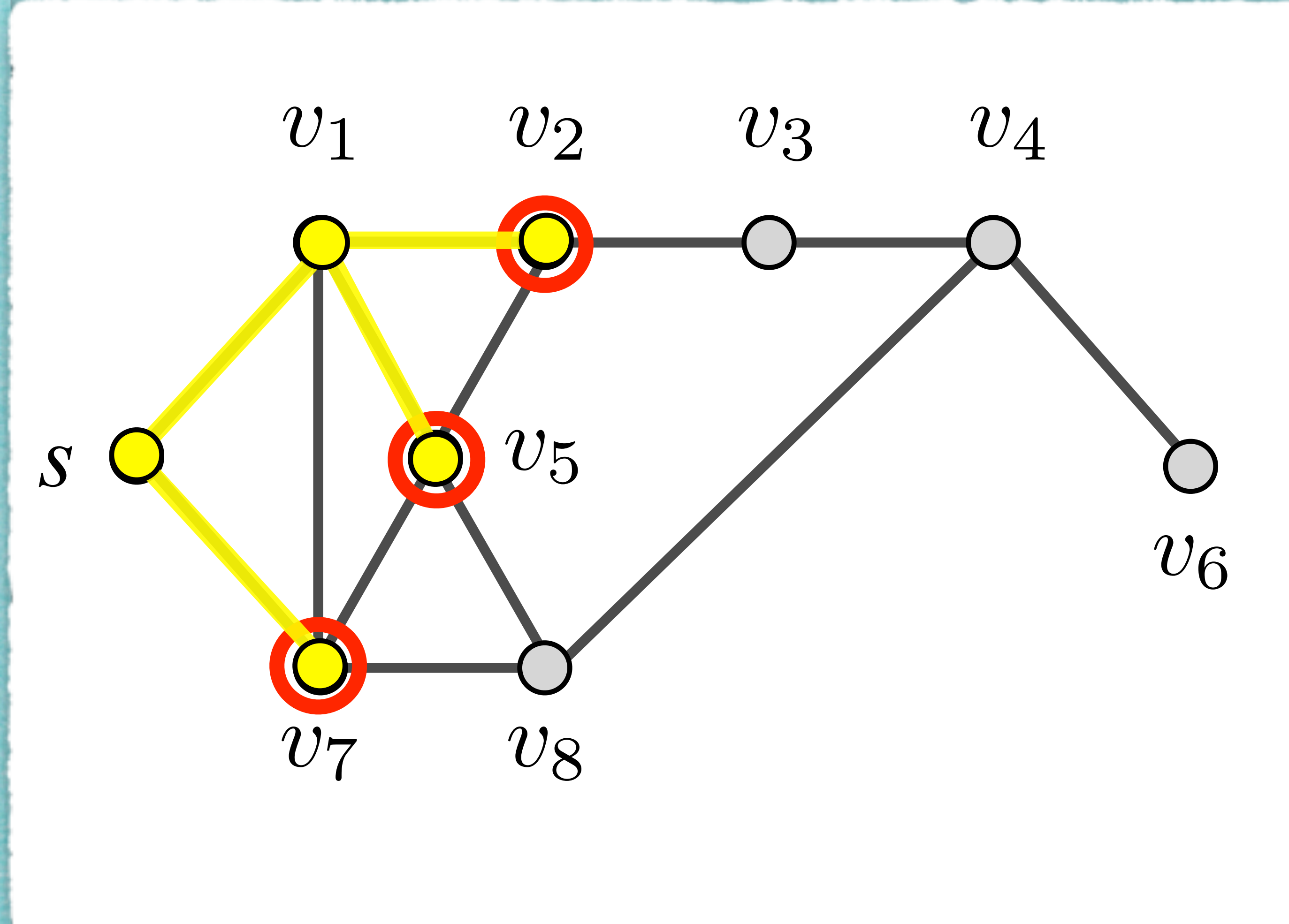
INPUT: Graph $G = (V,E)$, Knoten s

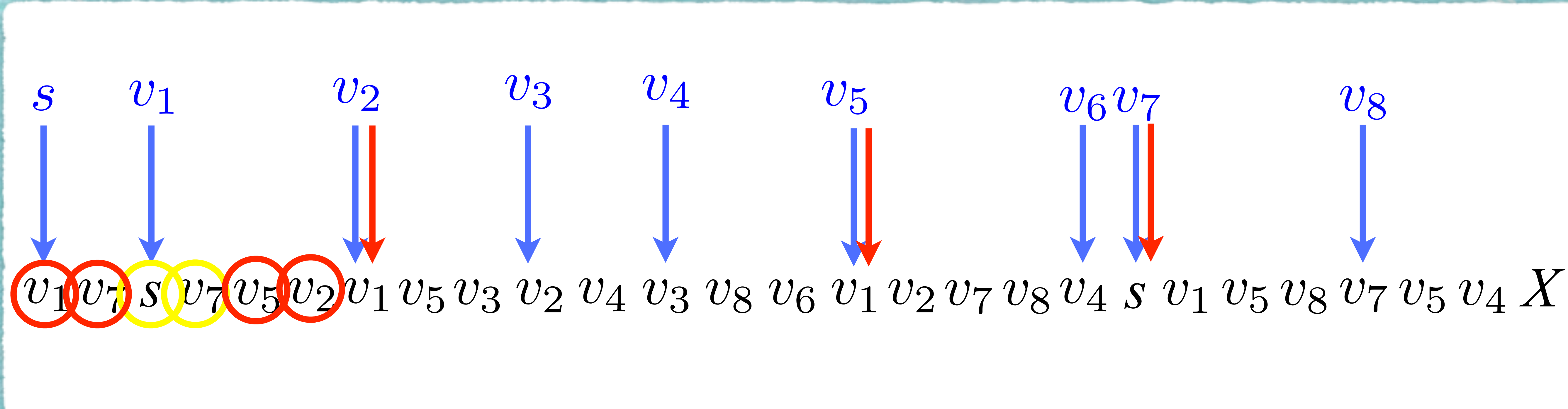
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



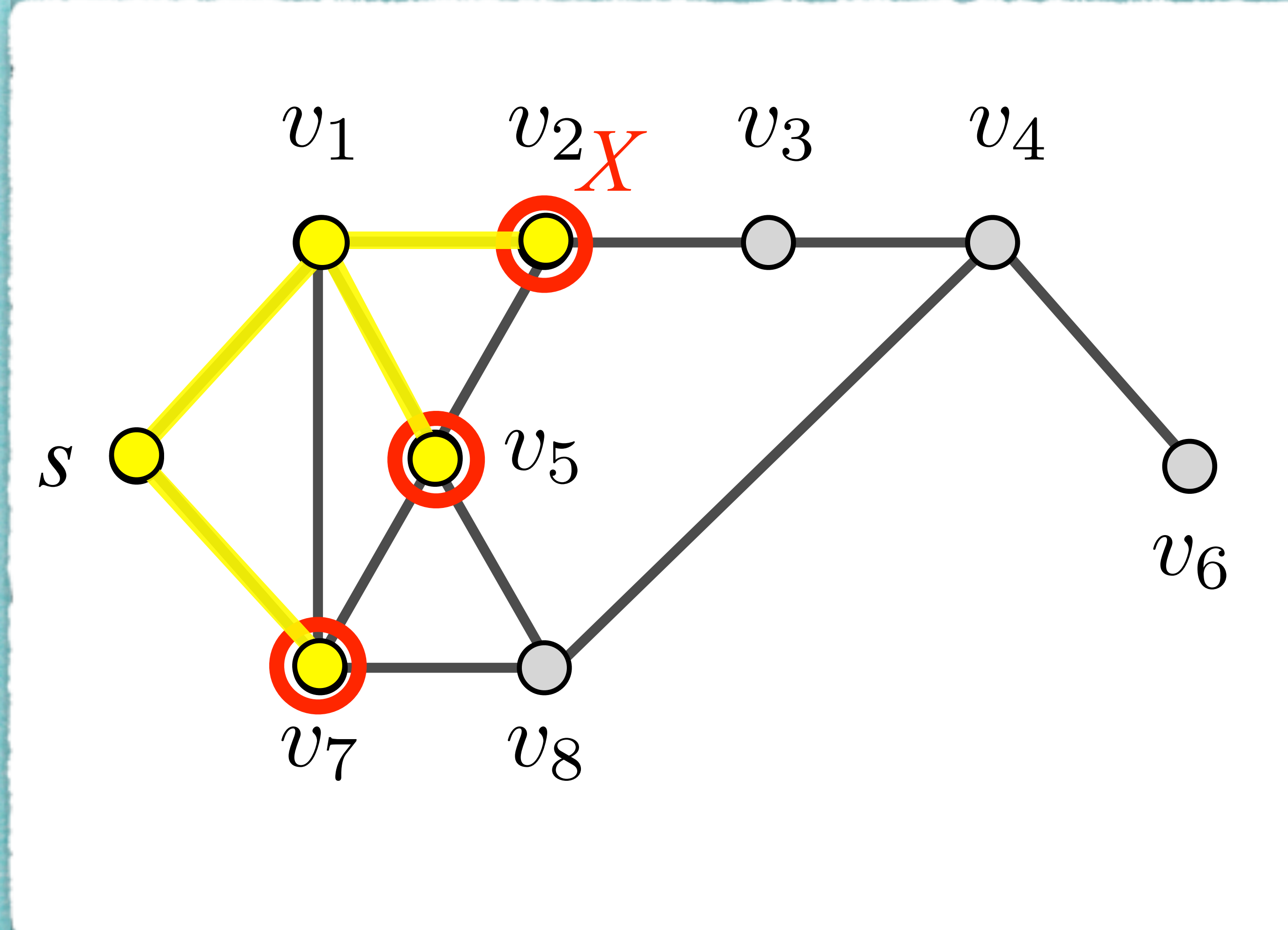


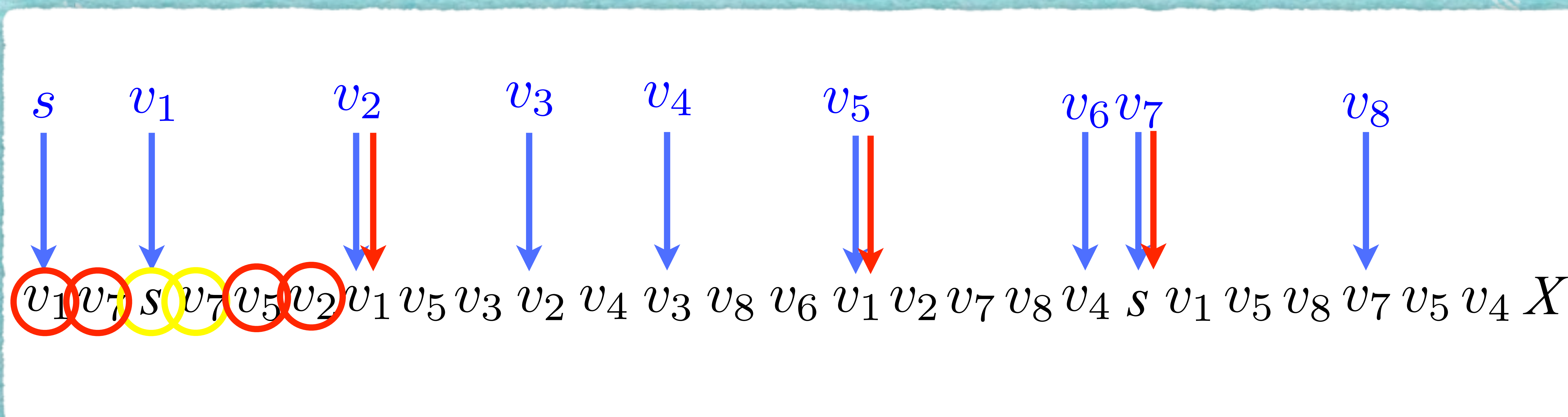
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        - }

    }
3. STOP



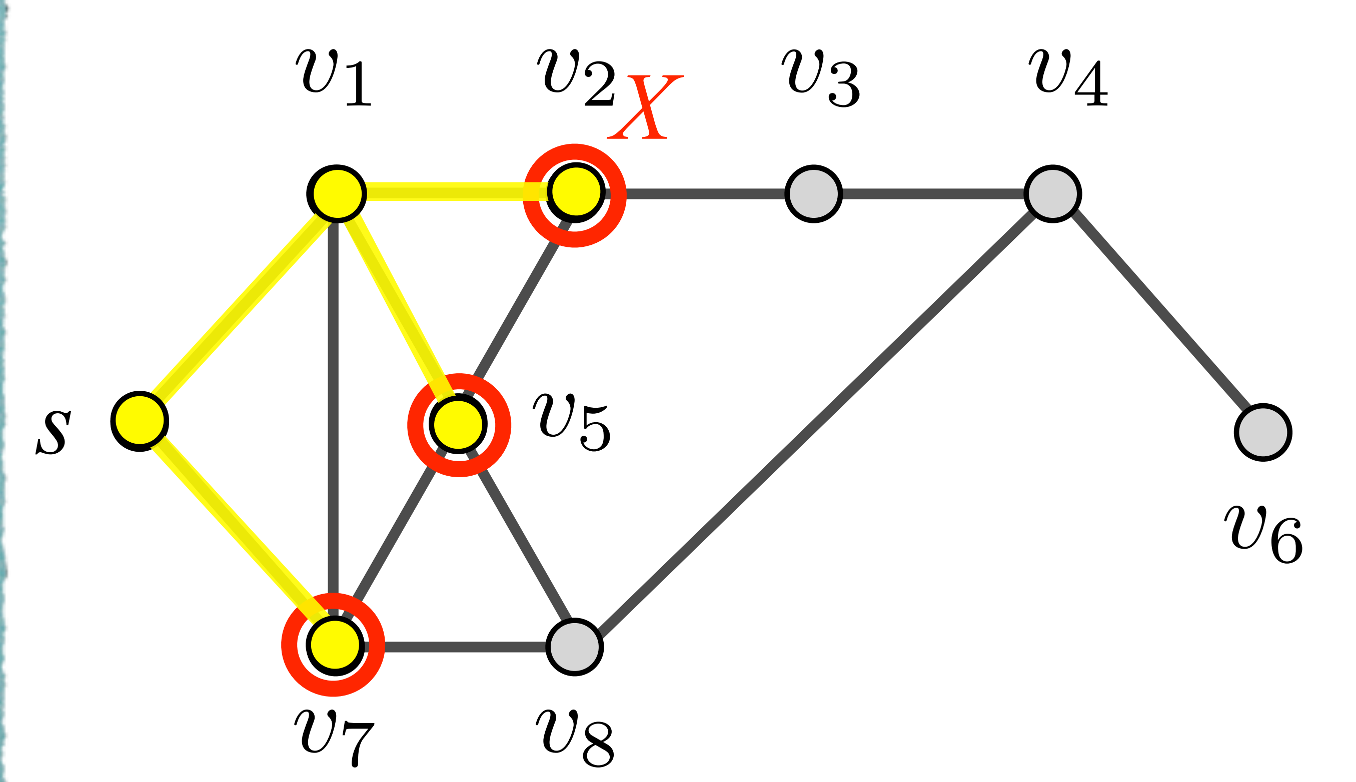


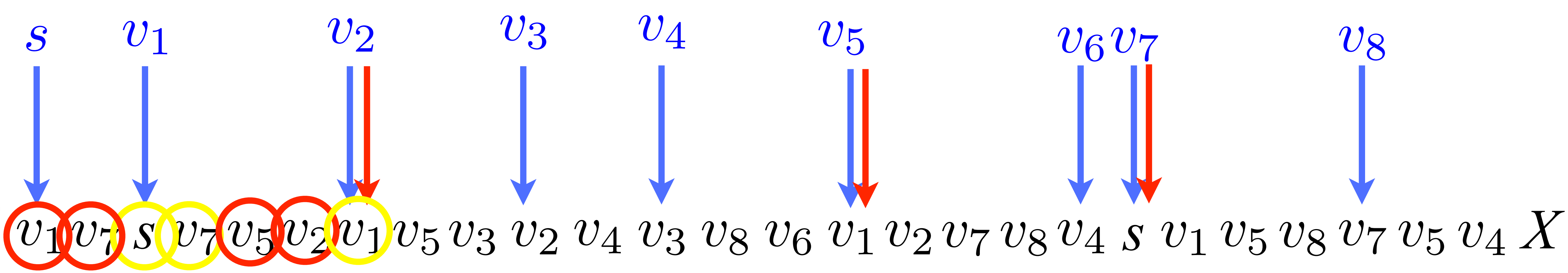
Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
     2.1. Wähle v∈R
     2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
          2.2.1. R:=R\{v}
     2.3. ELSE {
          2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
          2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
          }
   }
3. STOP
```



Algorithmus 3.7

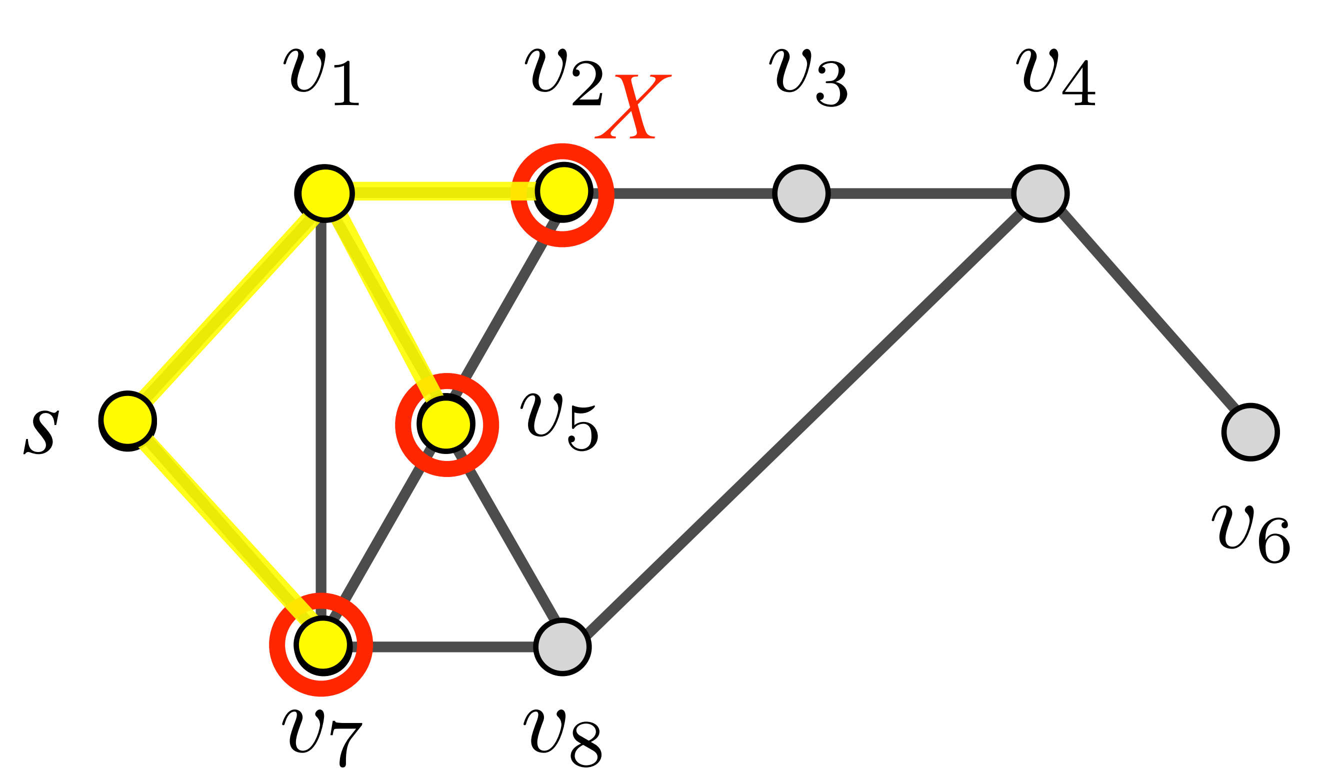
INPUT: Graph $G = (V,E)$, Knoten s

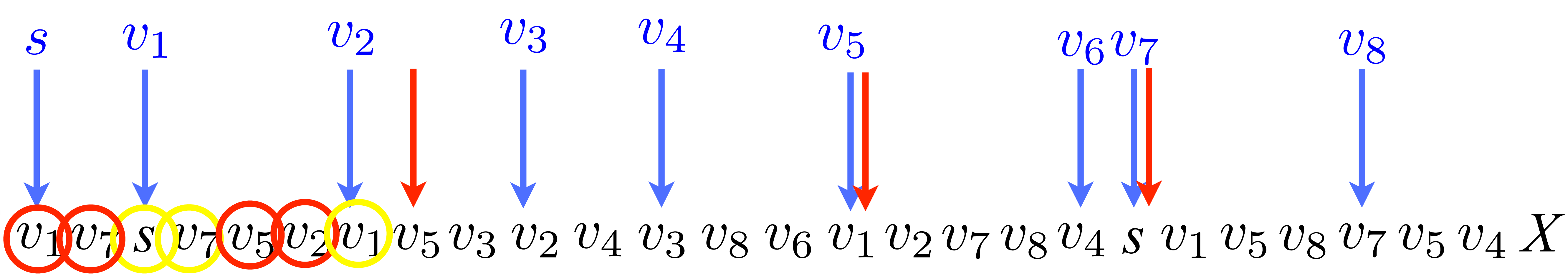
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
   }
3. STOP
```

2





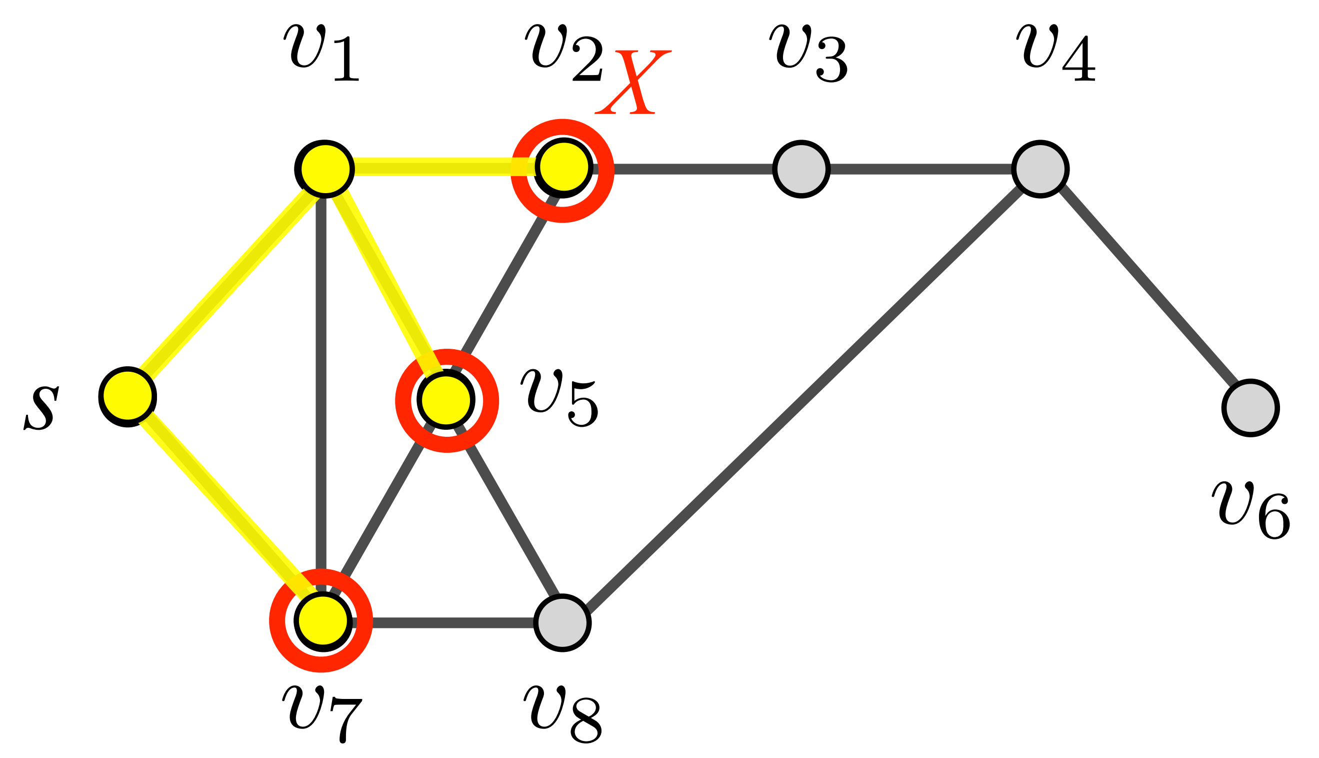
Algorithmus 3.7

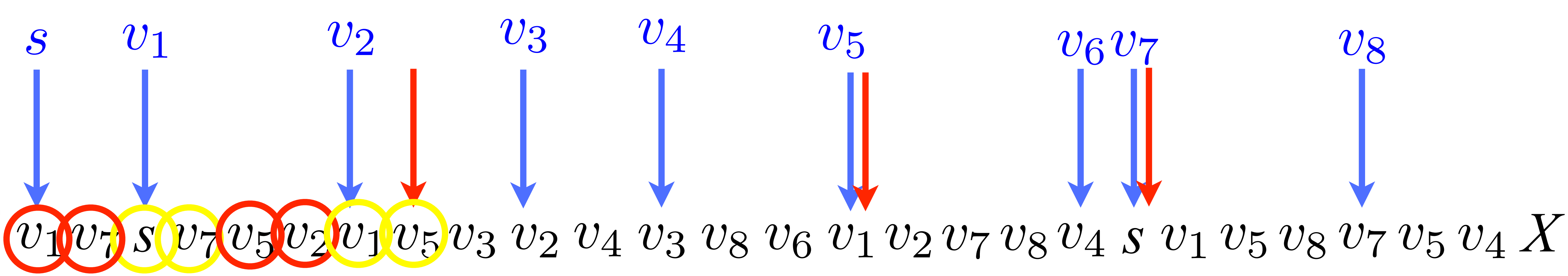
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v ∈ R
   - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w} ∈ E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

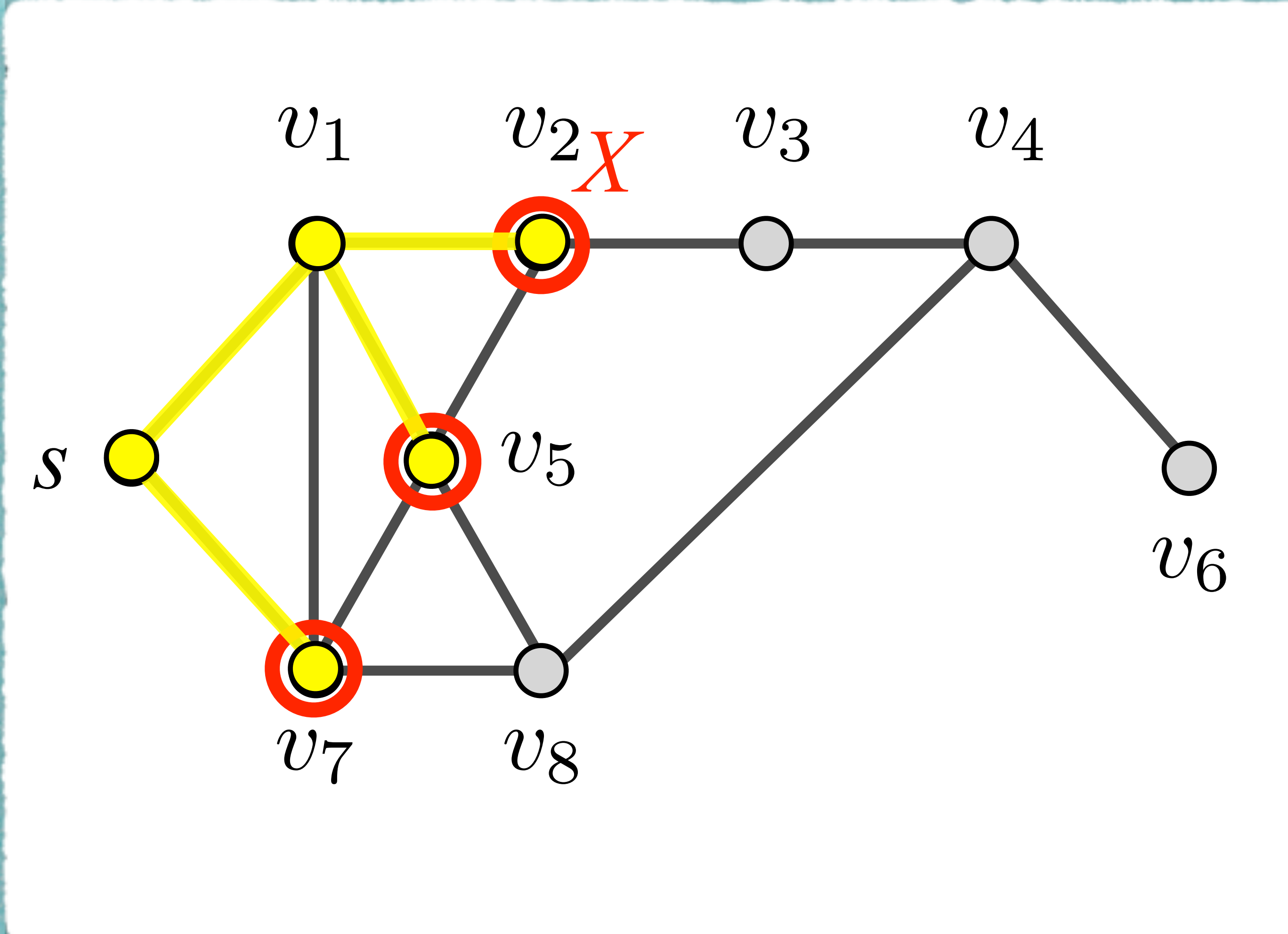
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist, Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

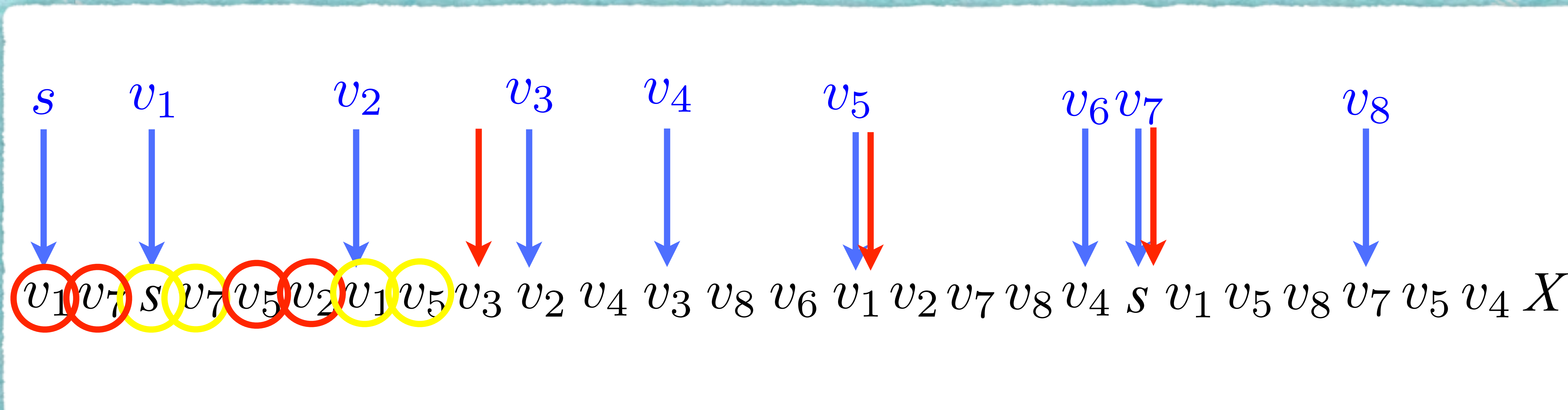
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2





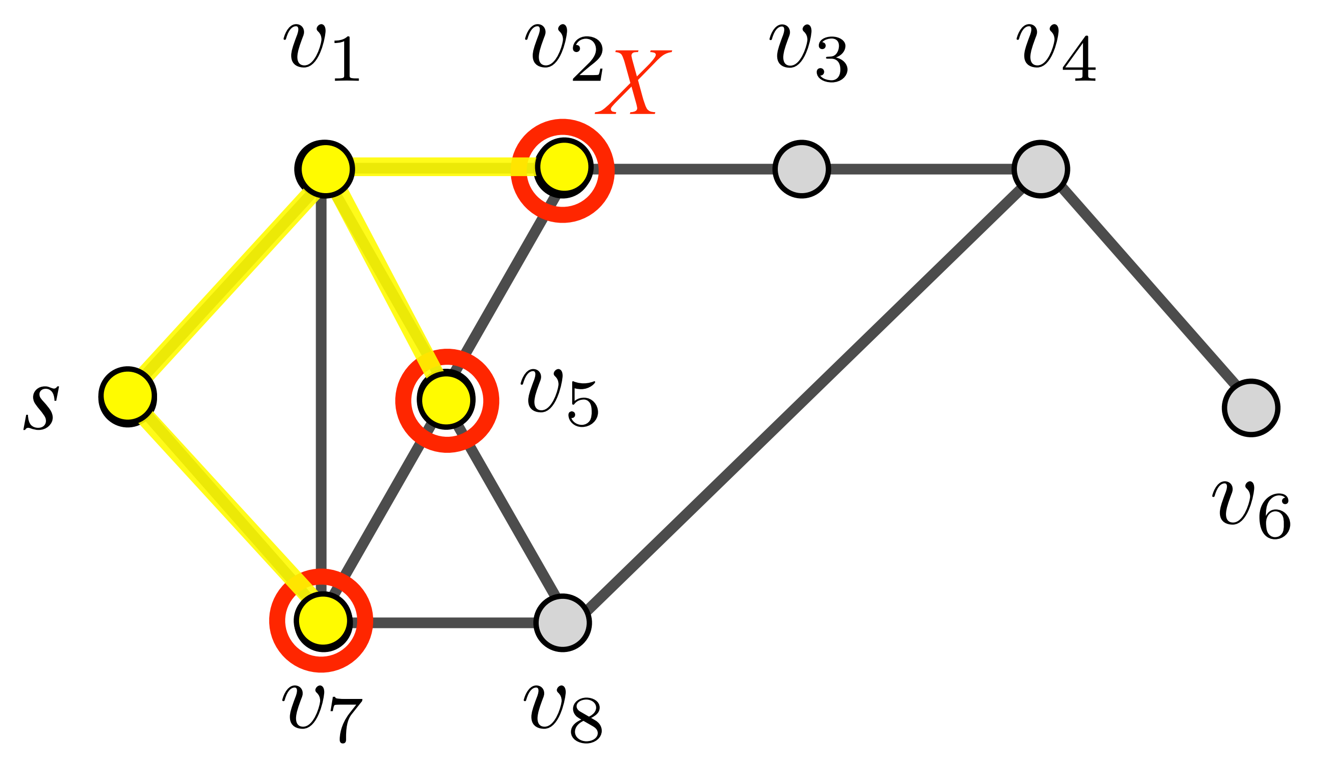
Algorithmus 3.7

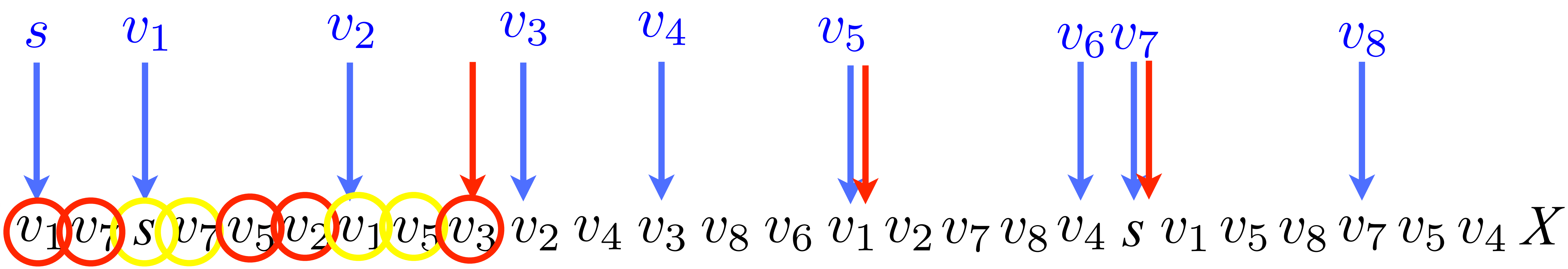
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





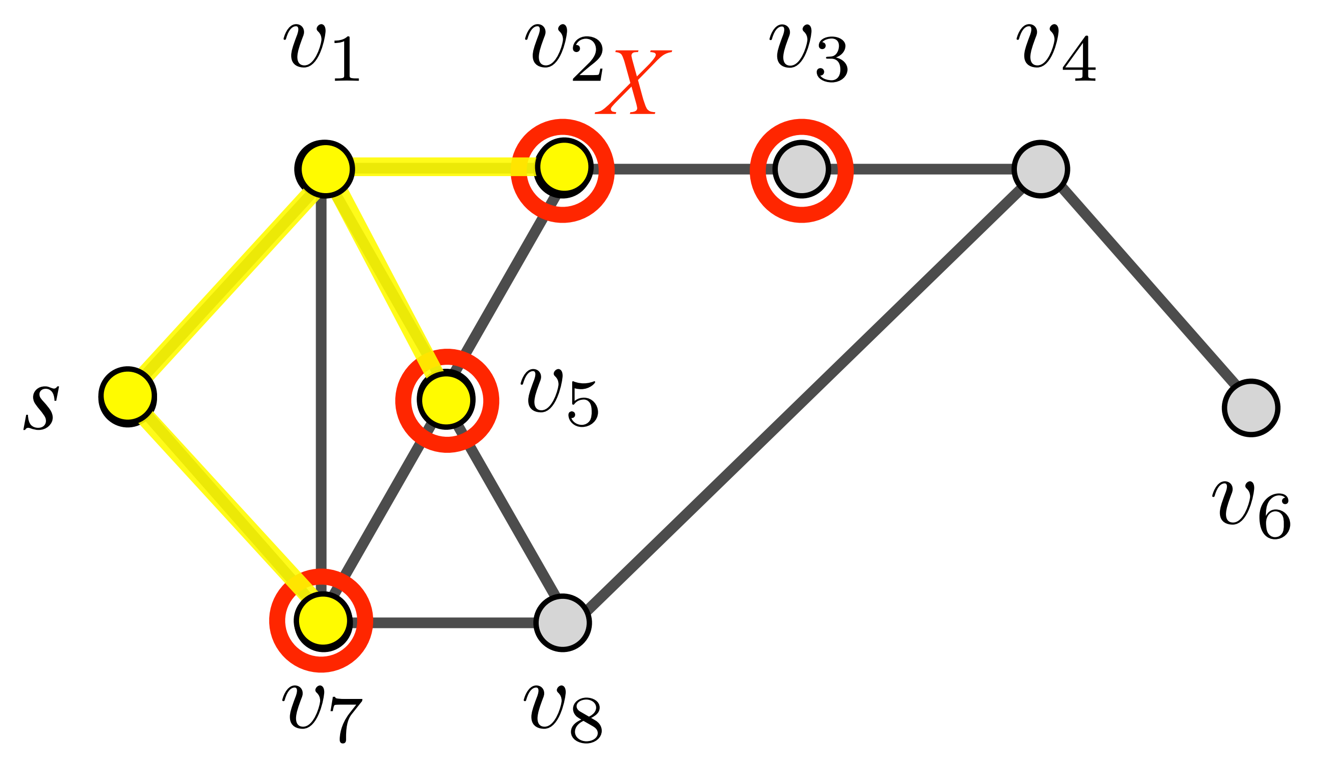
Algorithmus 3.7

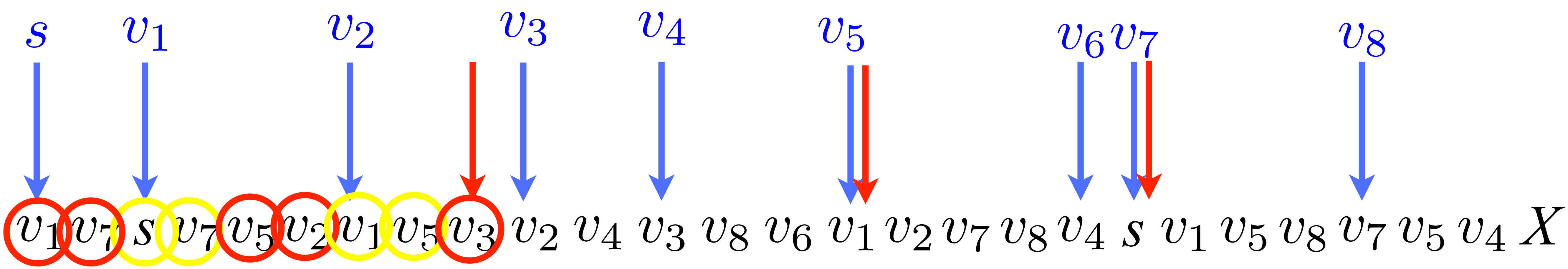
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
           2.2.1. R:=R\{v}
      2.3. ELSE {
           2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
           2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
           }
    }
3.  STOP
```



Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

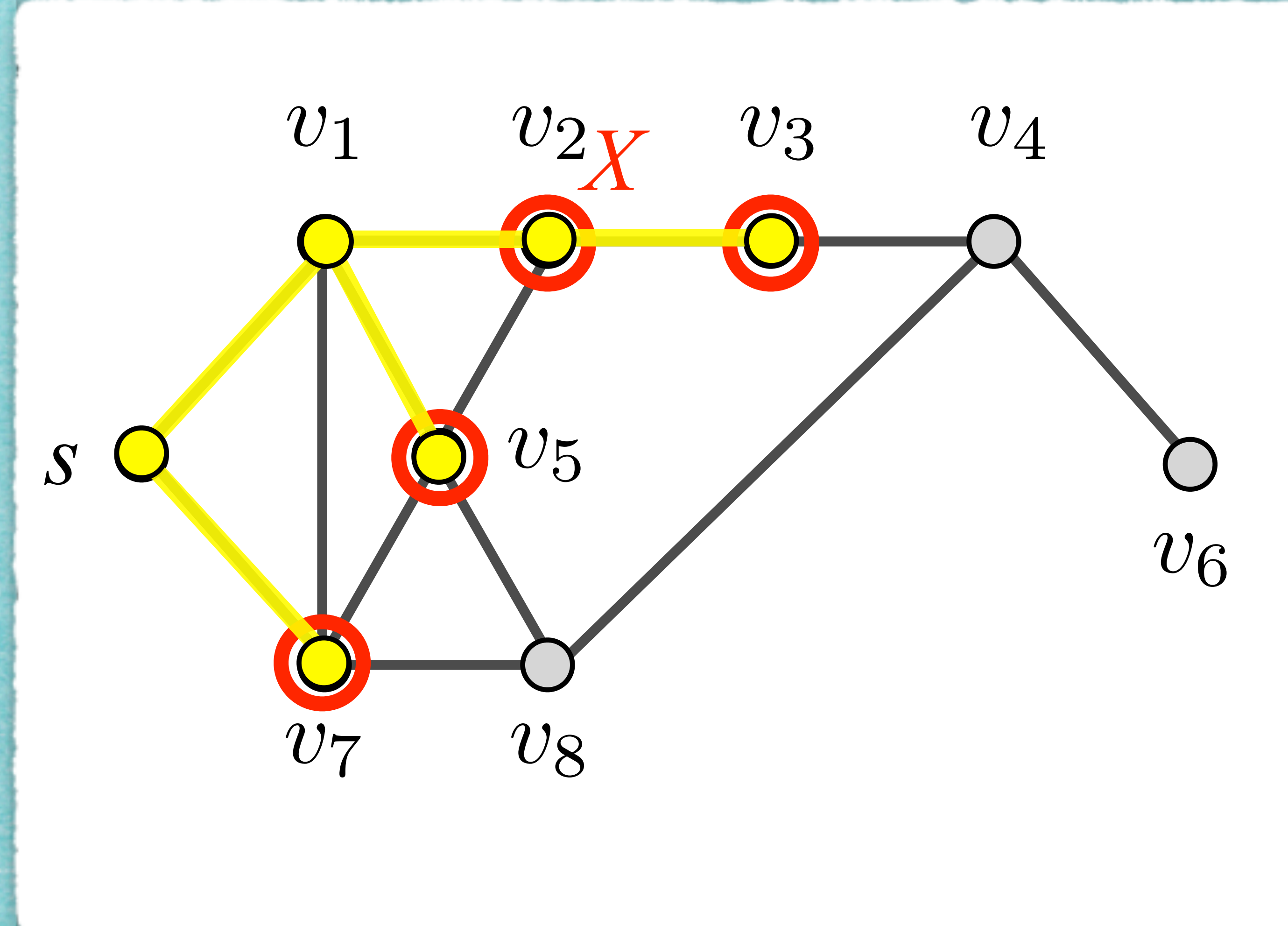
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

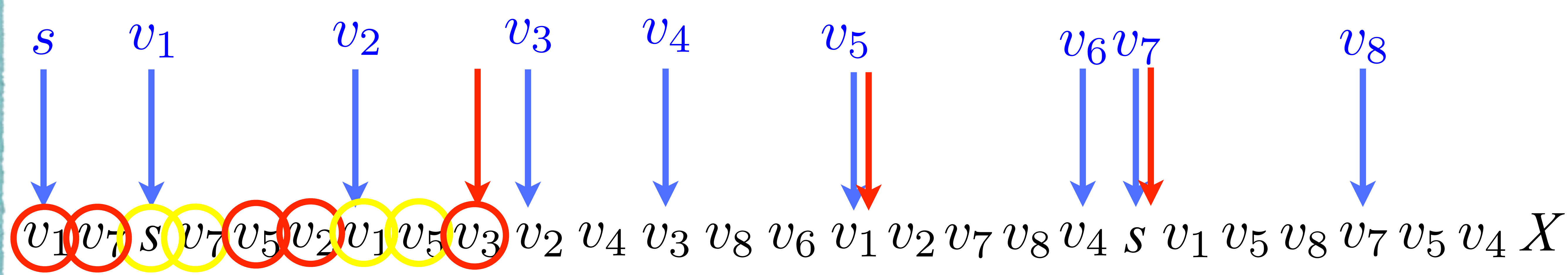
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP

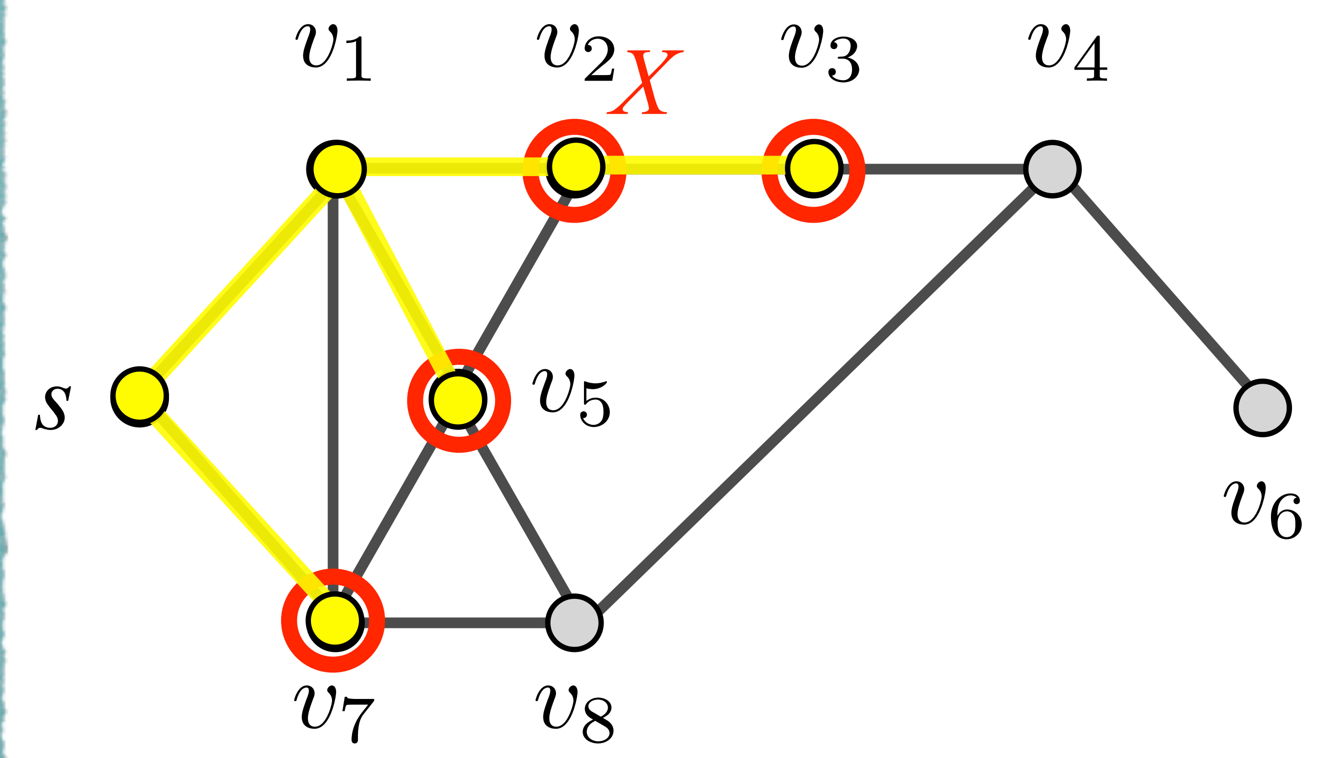


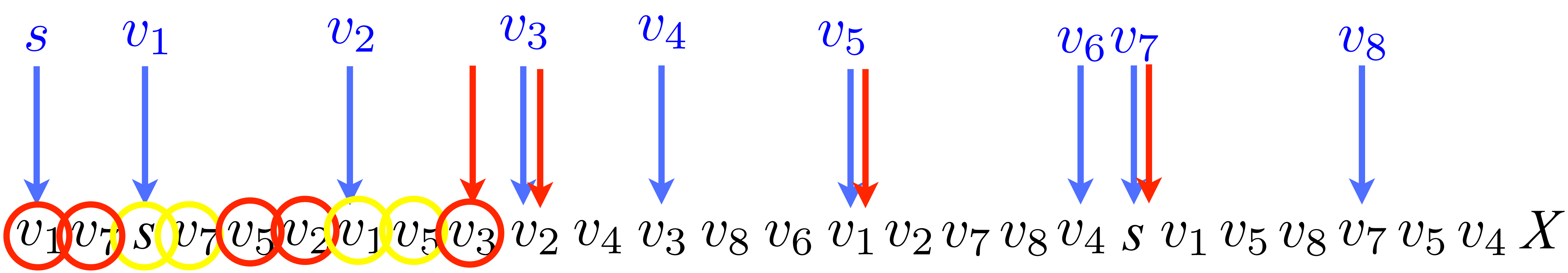


Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }
   - }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

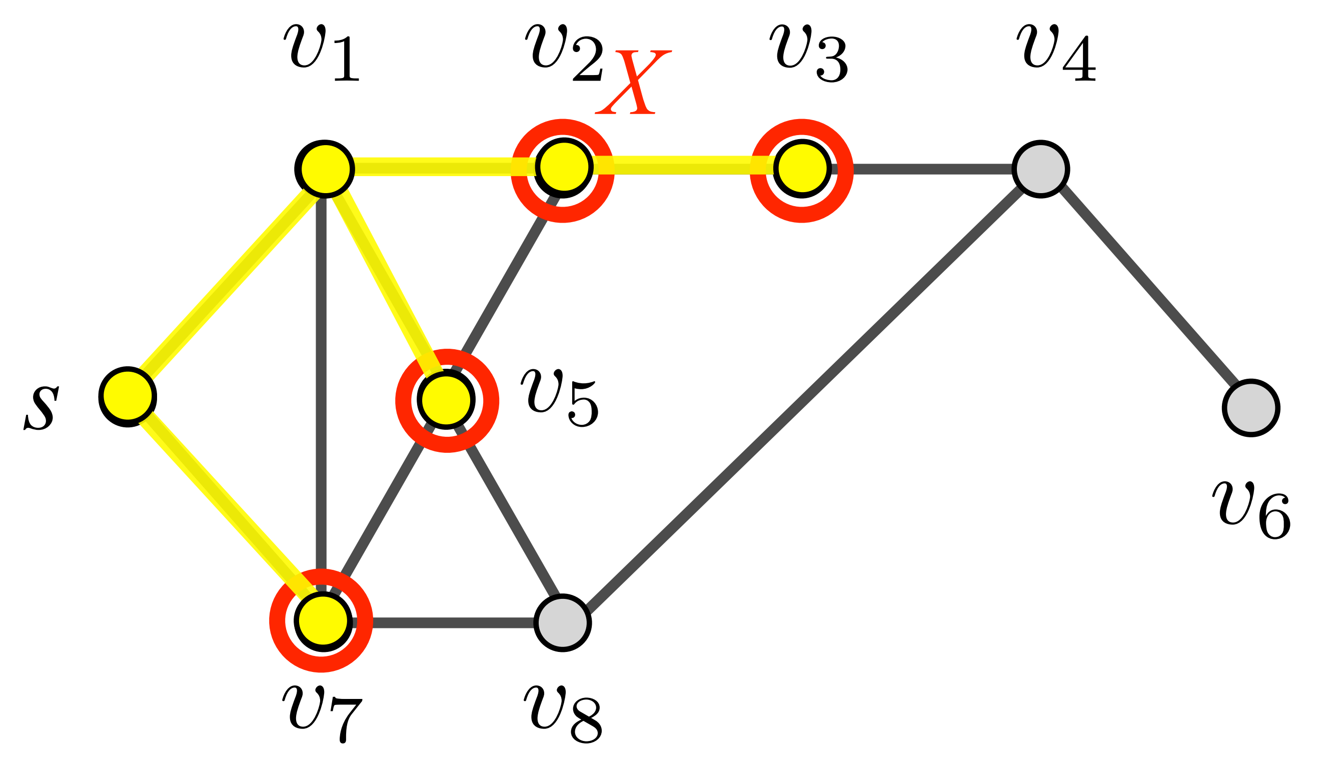
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

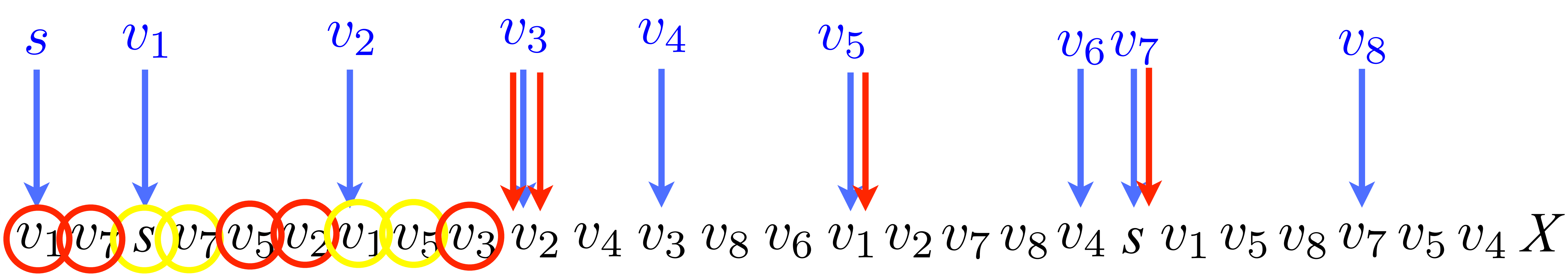
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

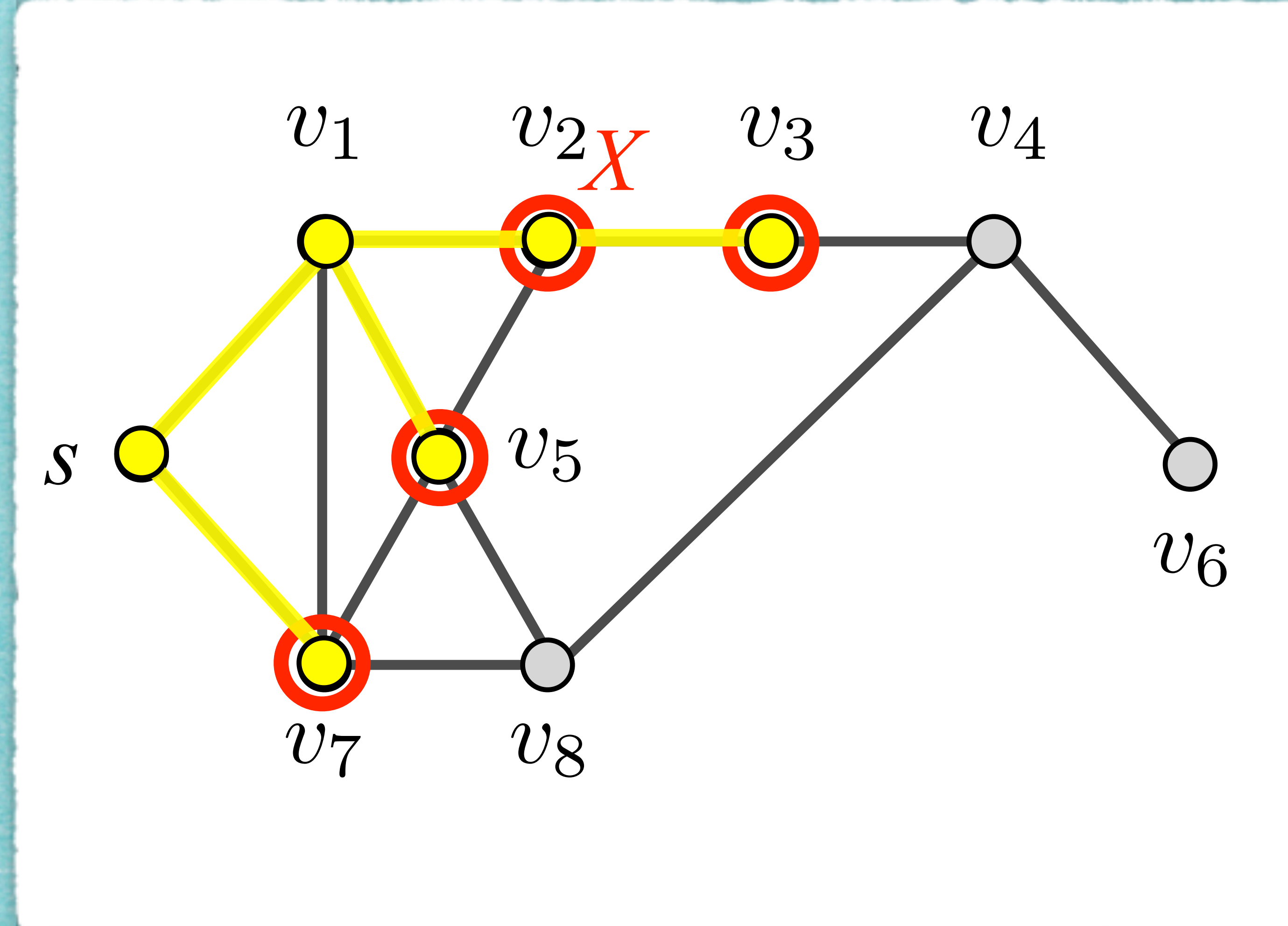
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

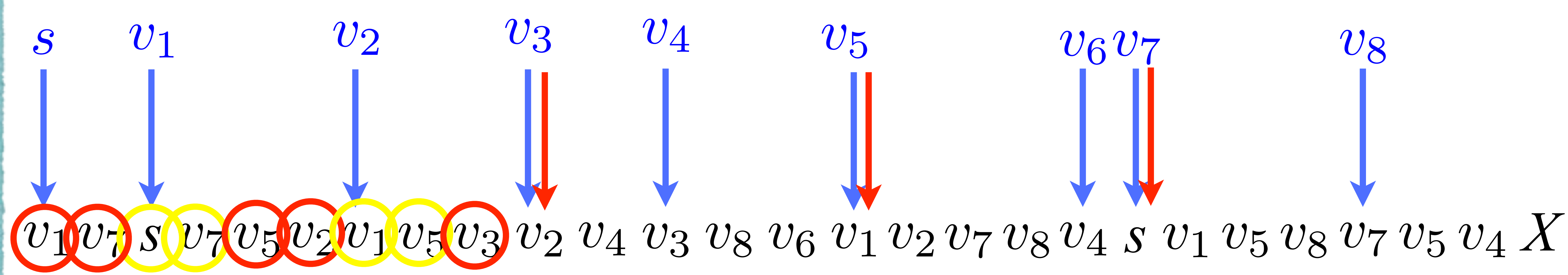
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

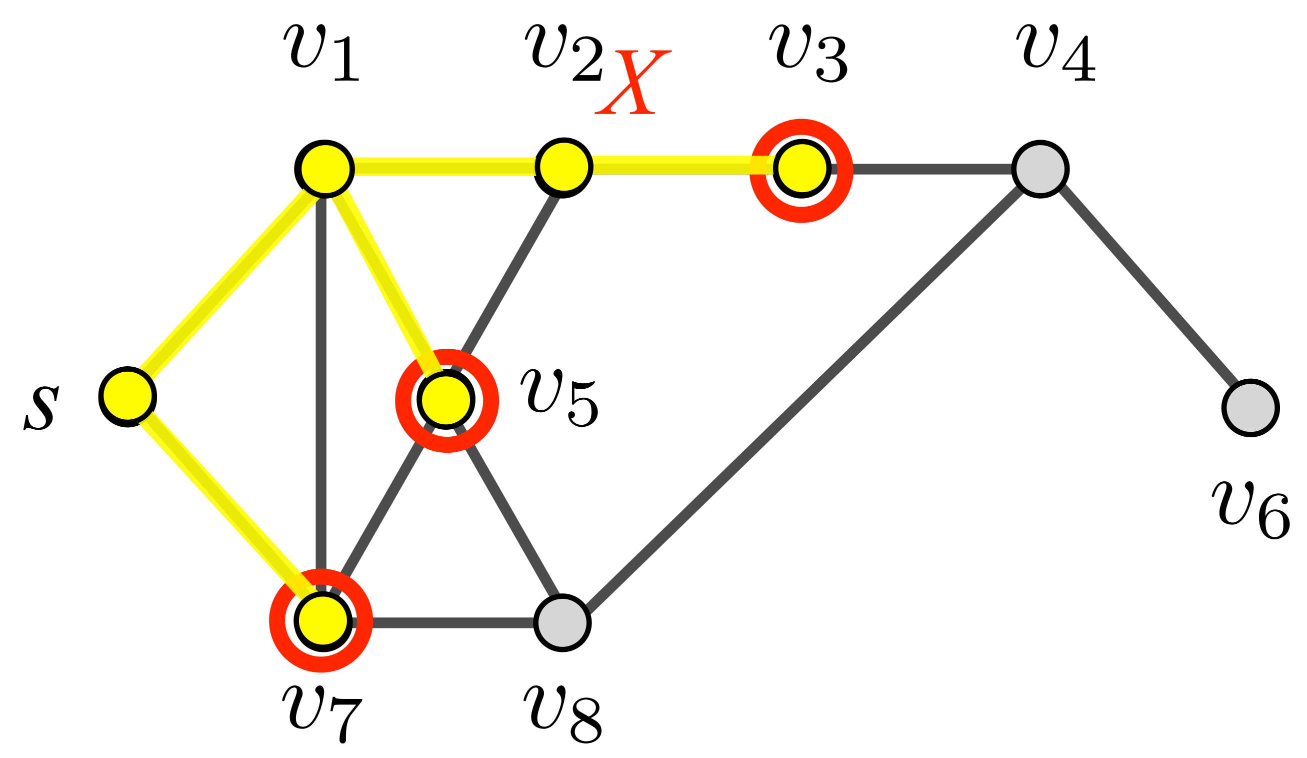
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

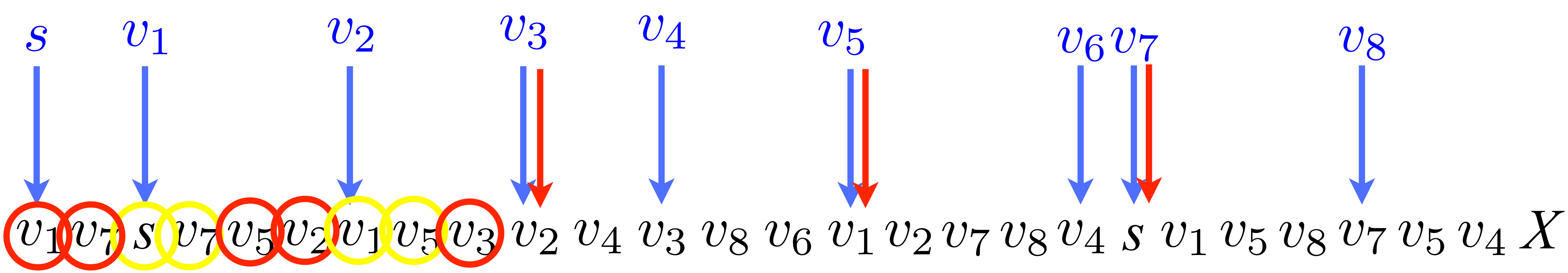
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

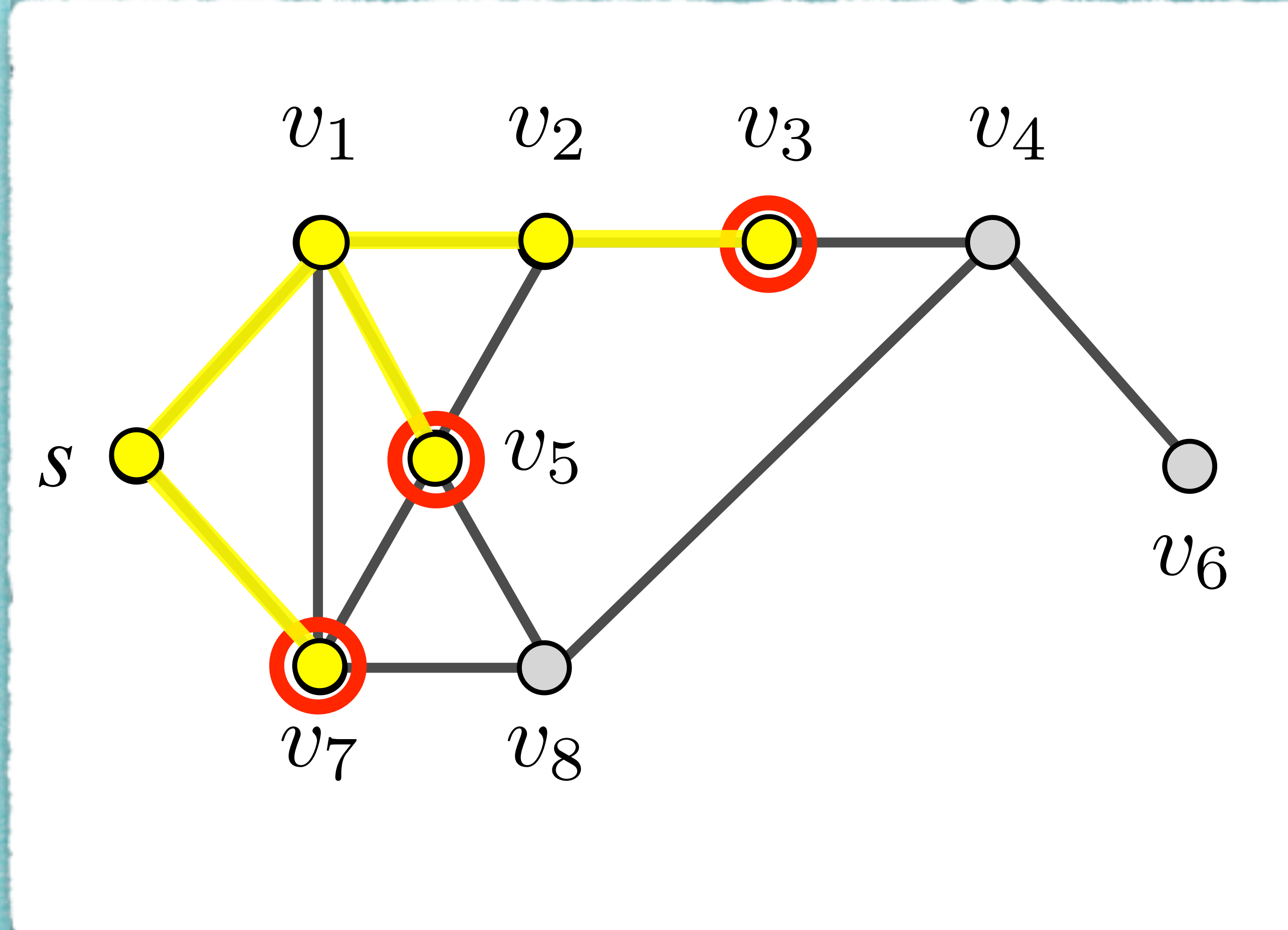
INPUT: Graph $G = (V,E)$, Knoten s

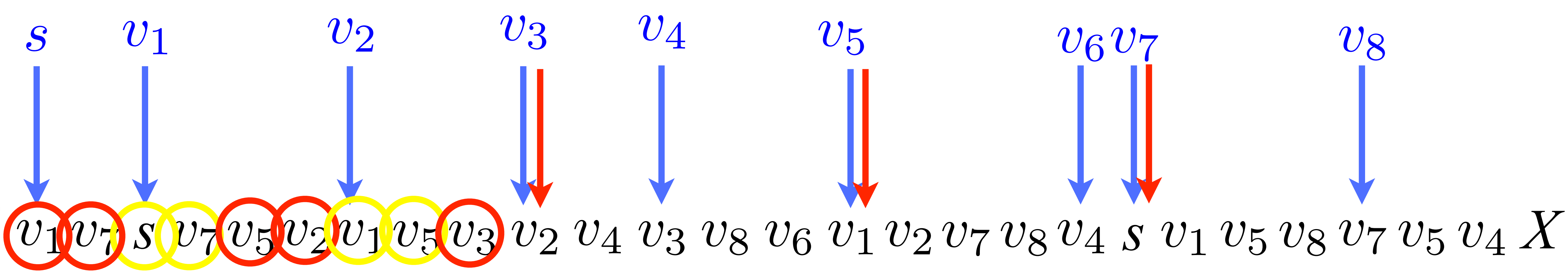
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   2.1. Wähle $v\in R$
   2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
      2.2.1. $R:=R\setminus\{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP

2





Algorithmus 3.7

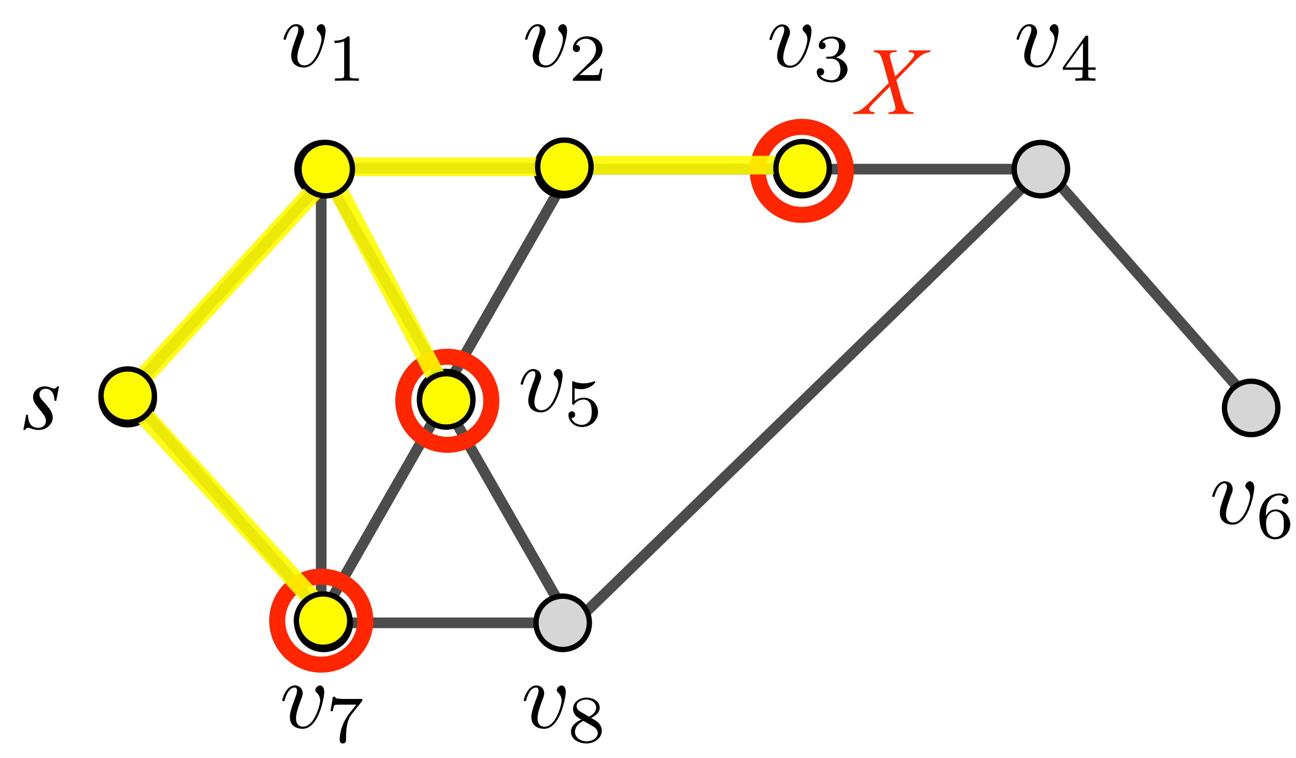
INPUT: Graph $G = (V,E)$, Knoten s

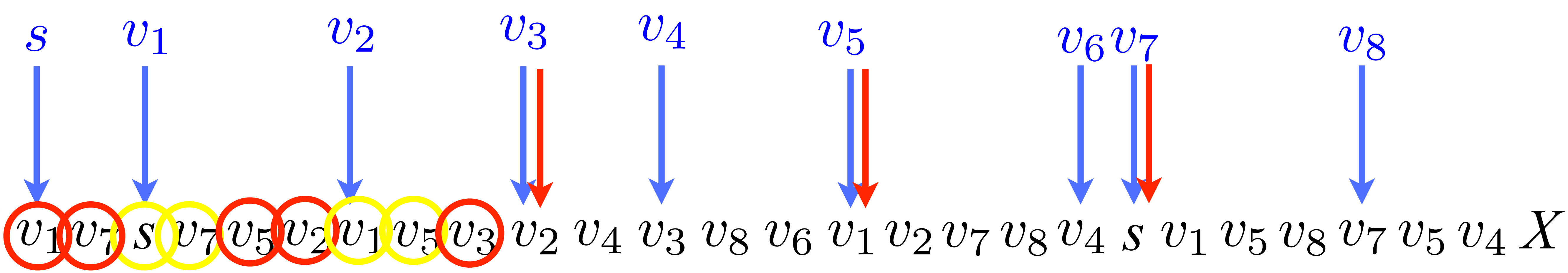
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

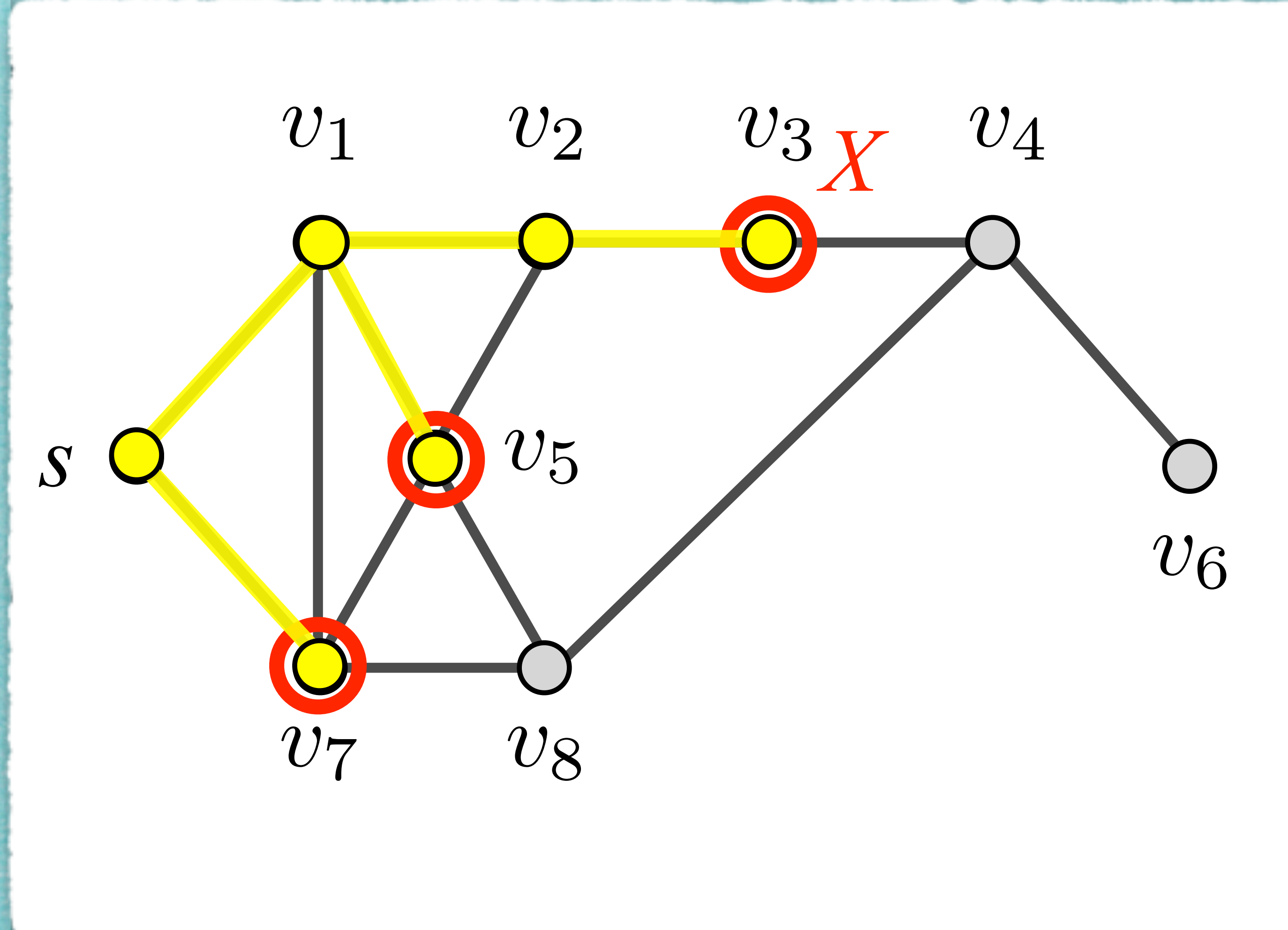
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

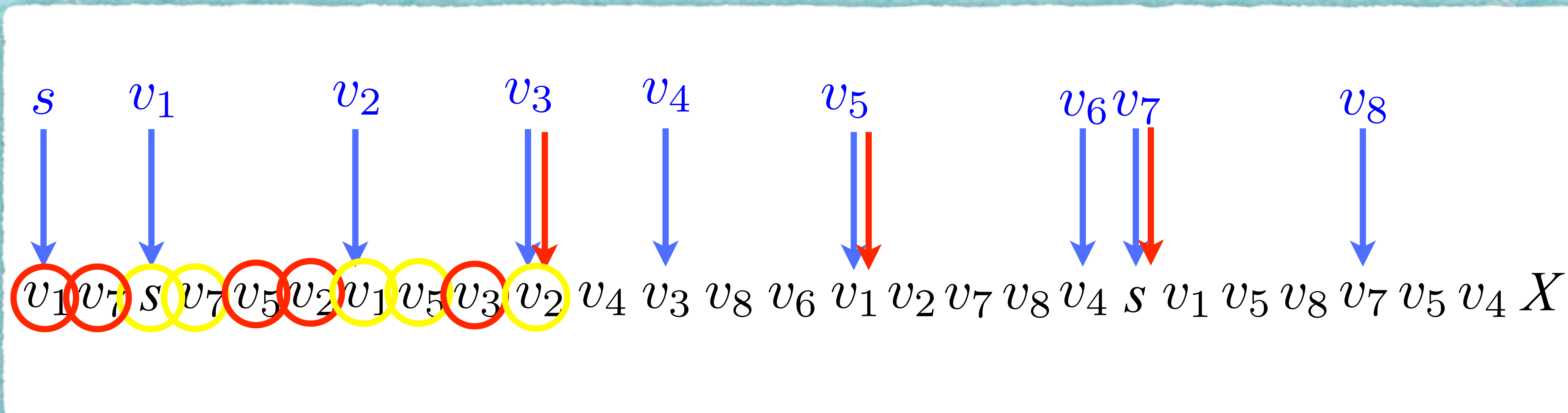
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



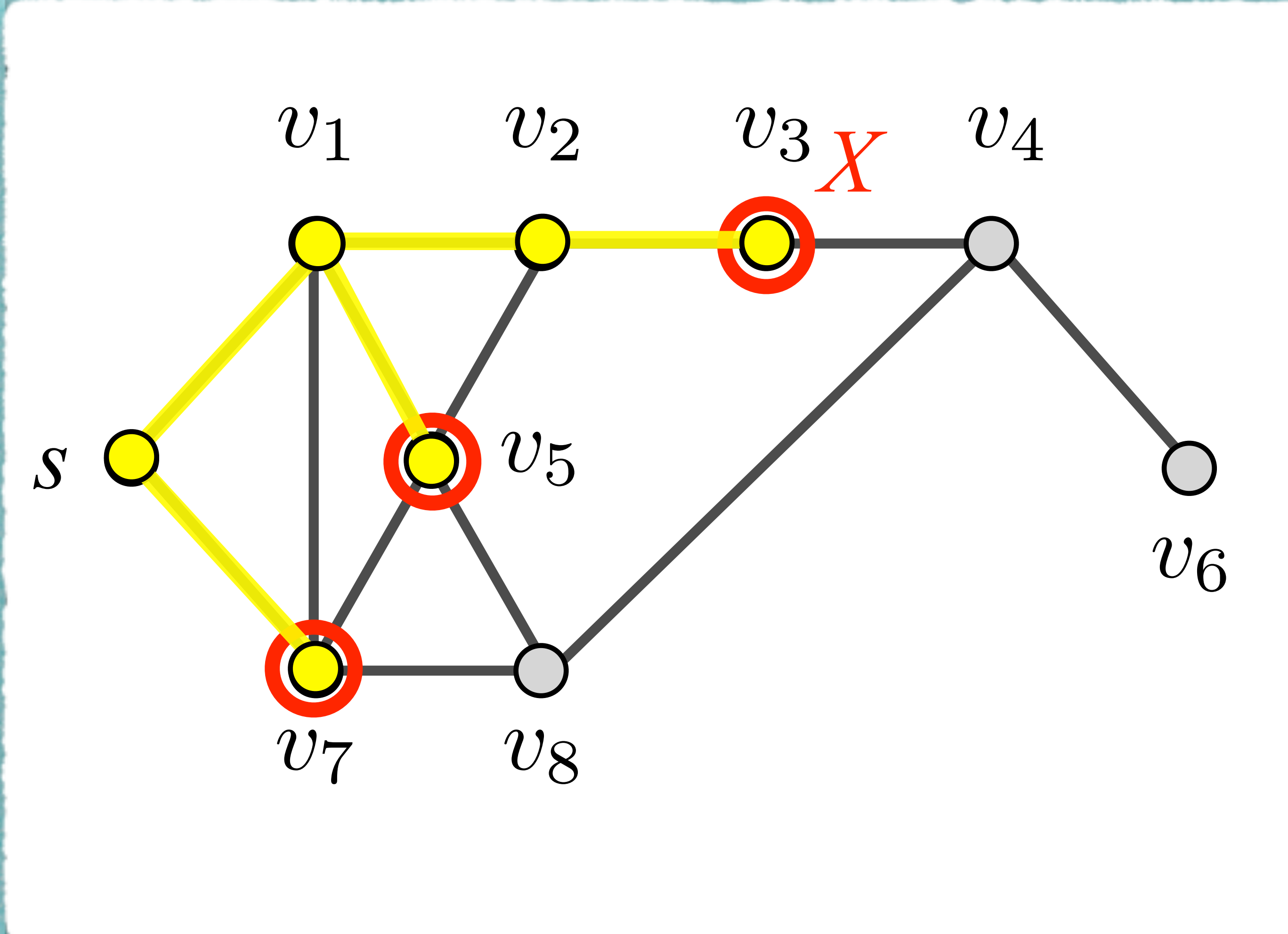


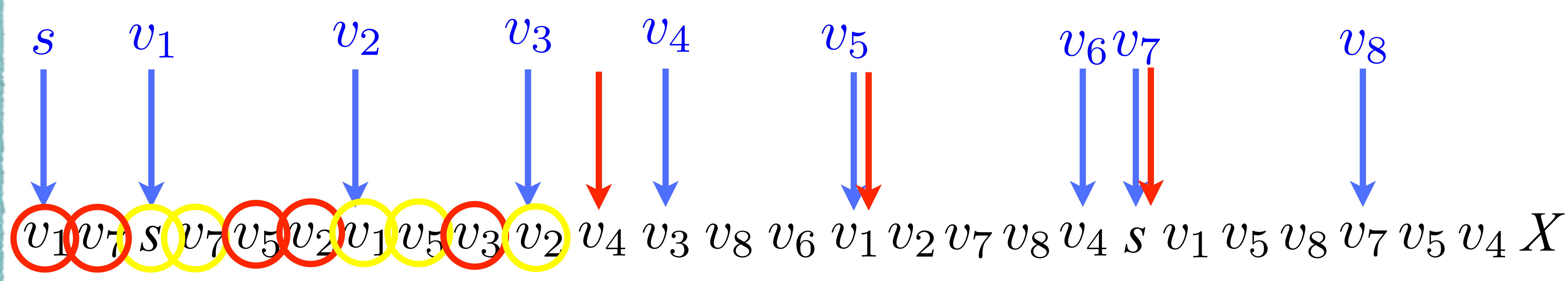
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    2.1. Wähle $v\in R$
    2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP

2





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

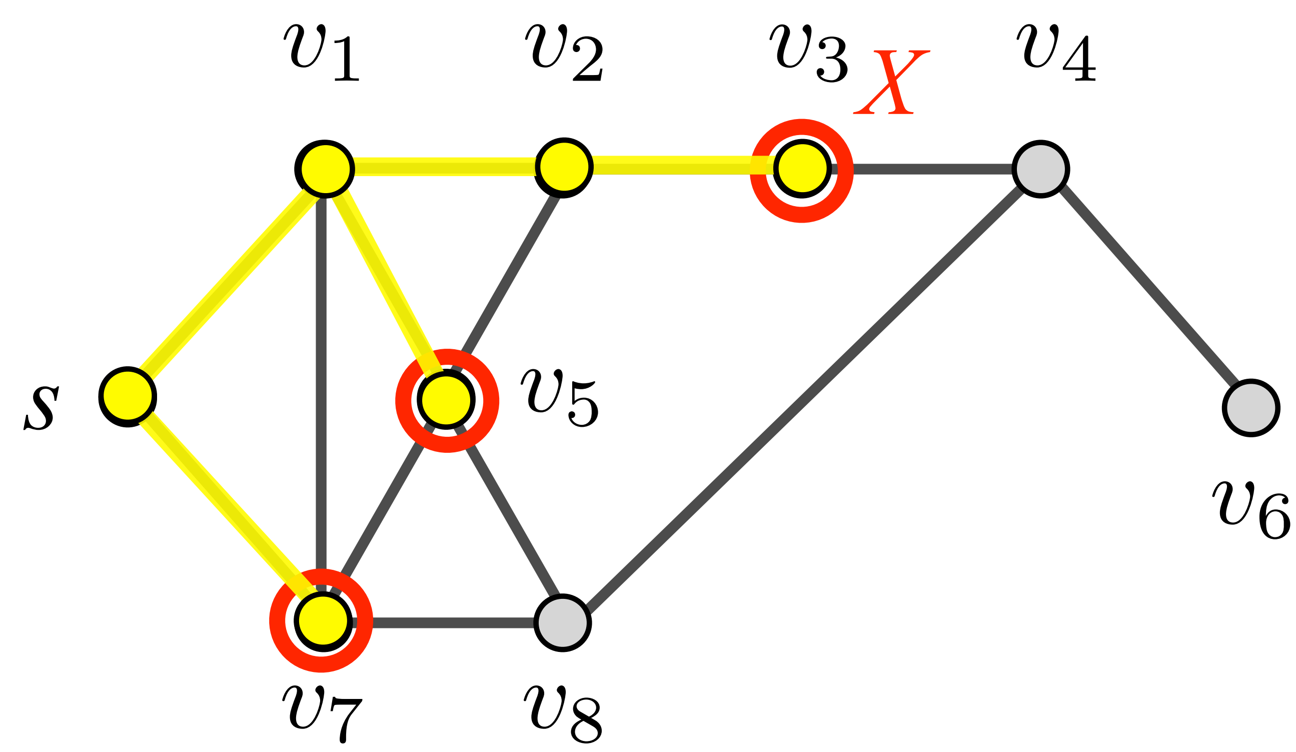
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

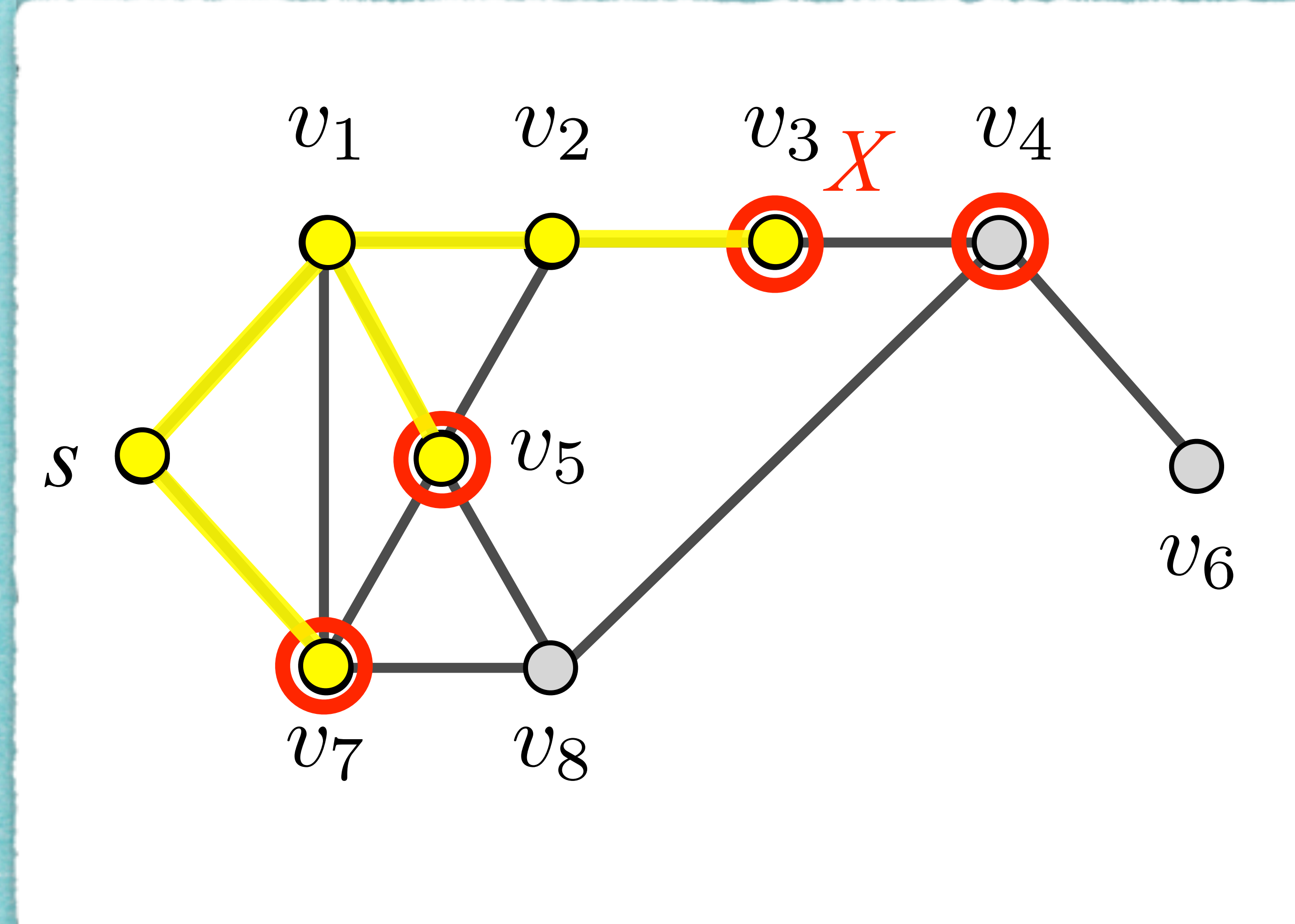
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

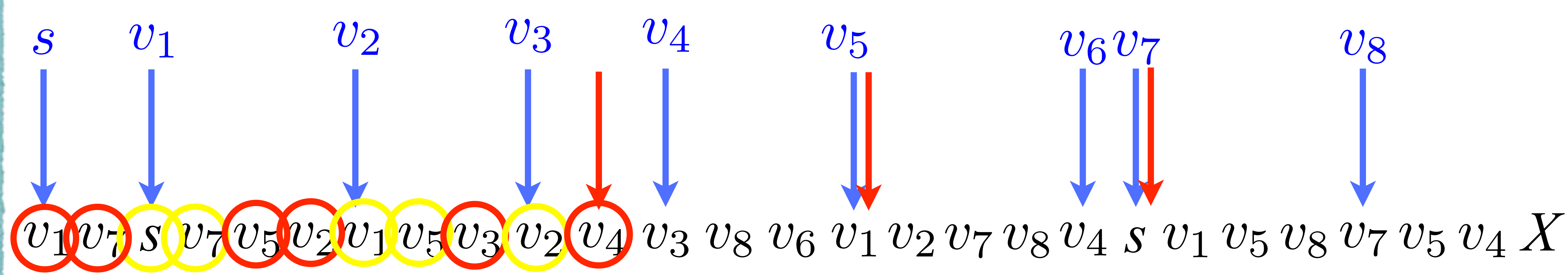
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

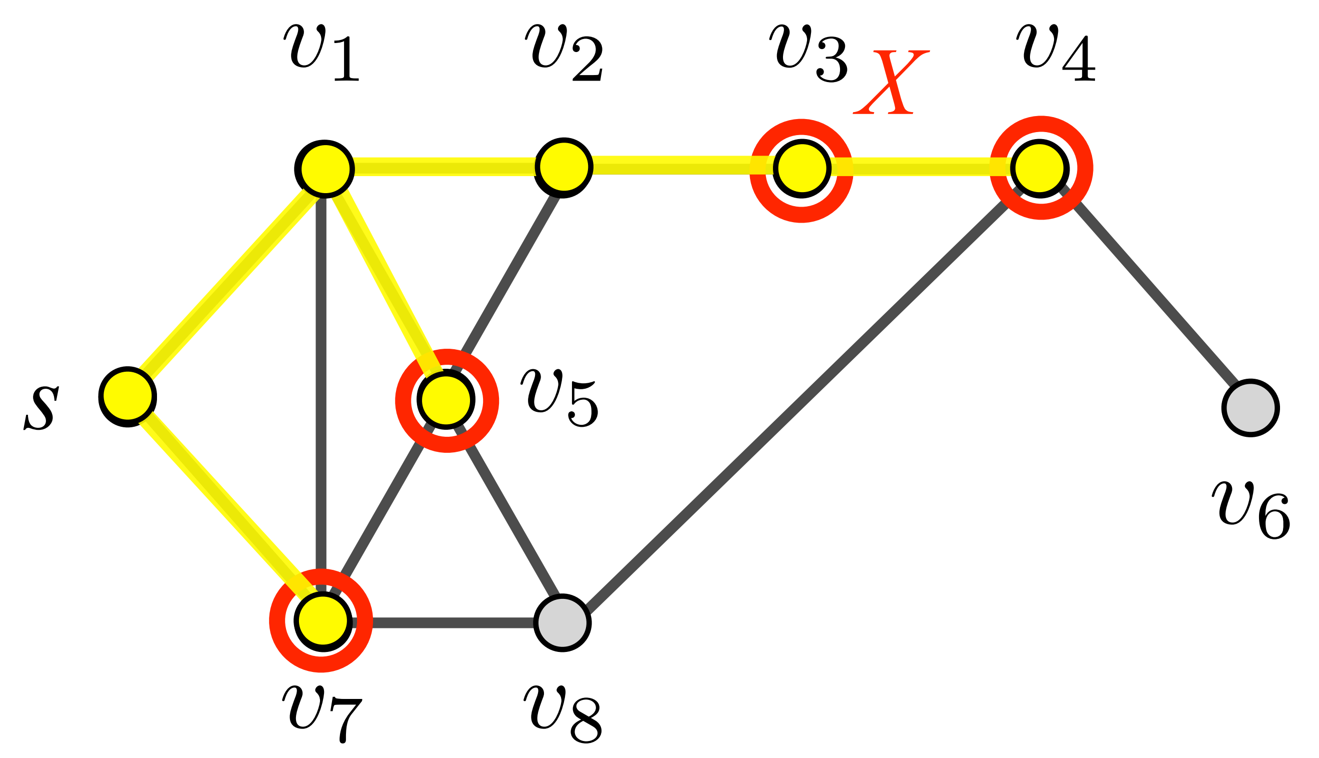
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

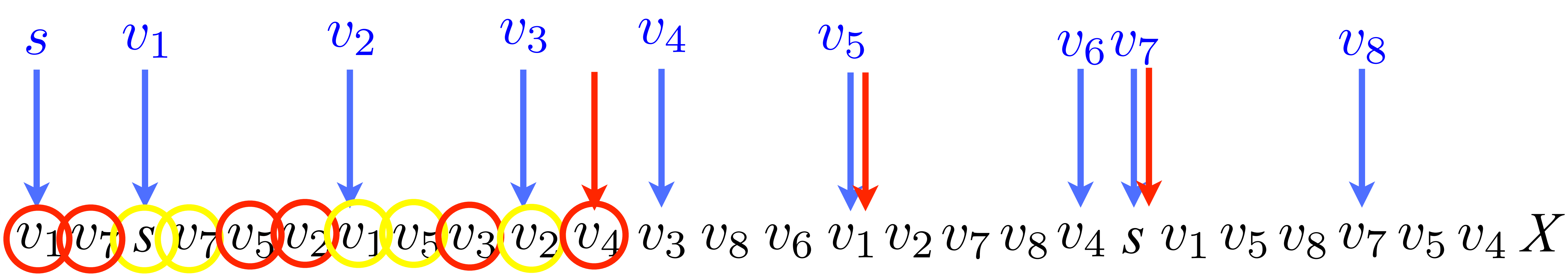
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

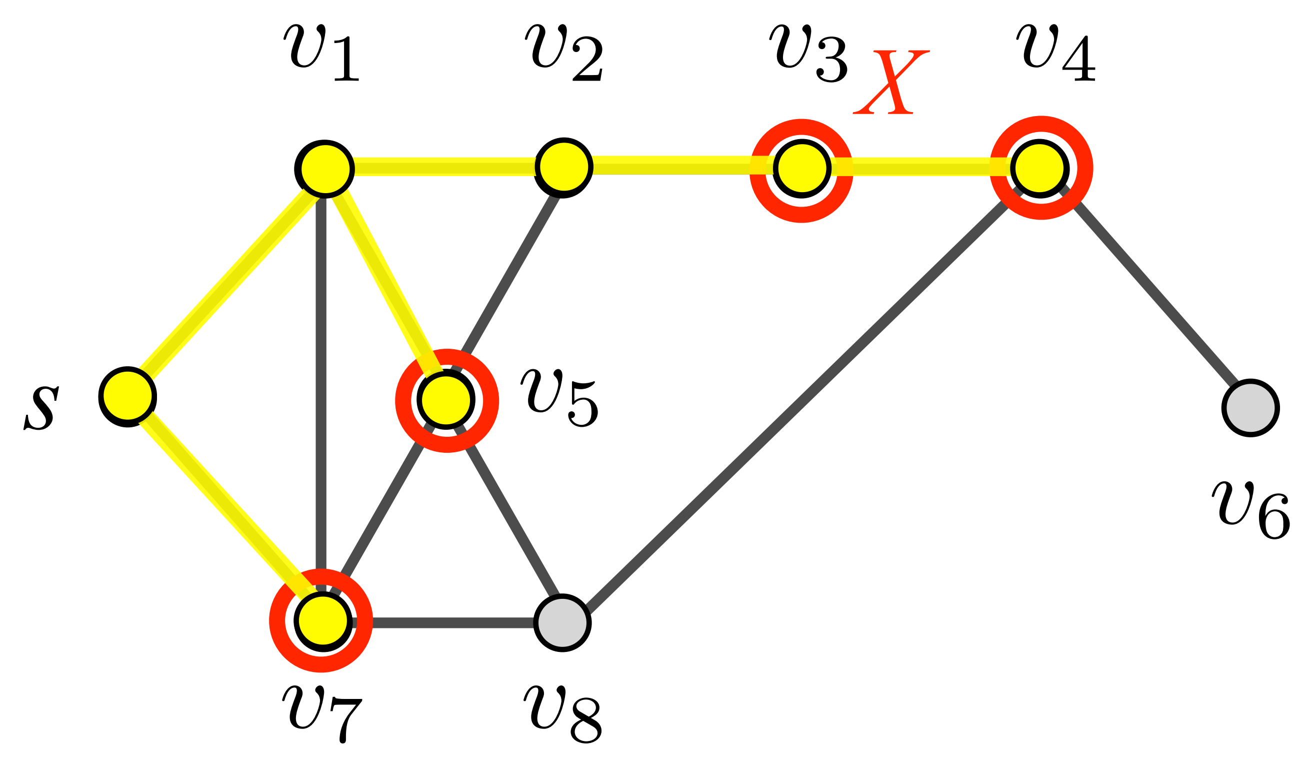
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

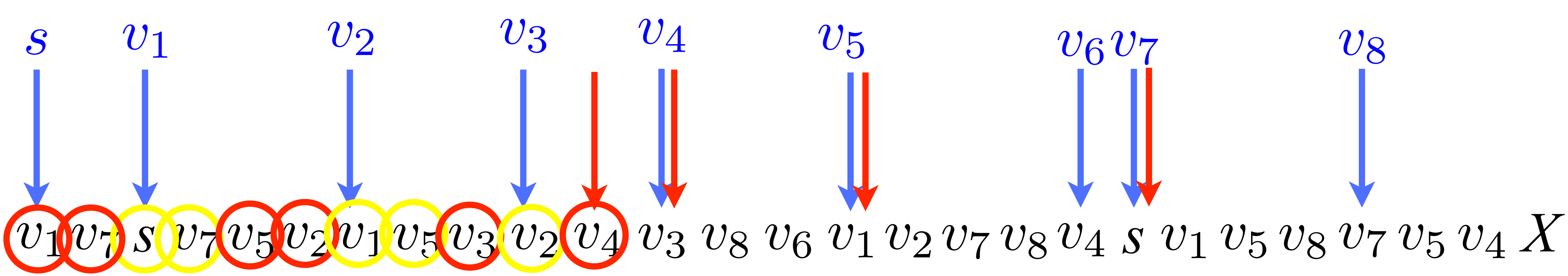
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





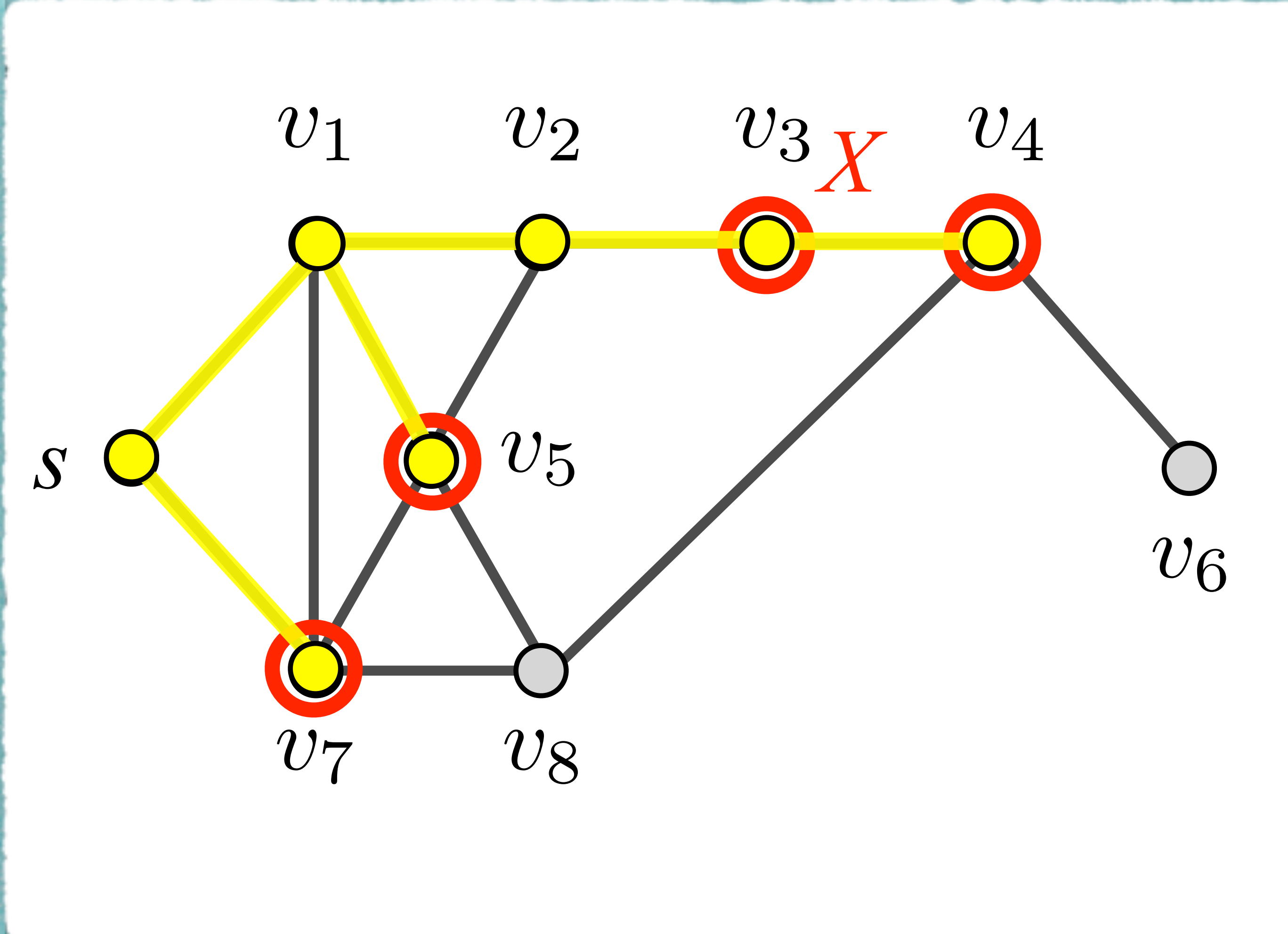
Algorithmus 3.7

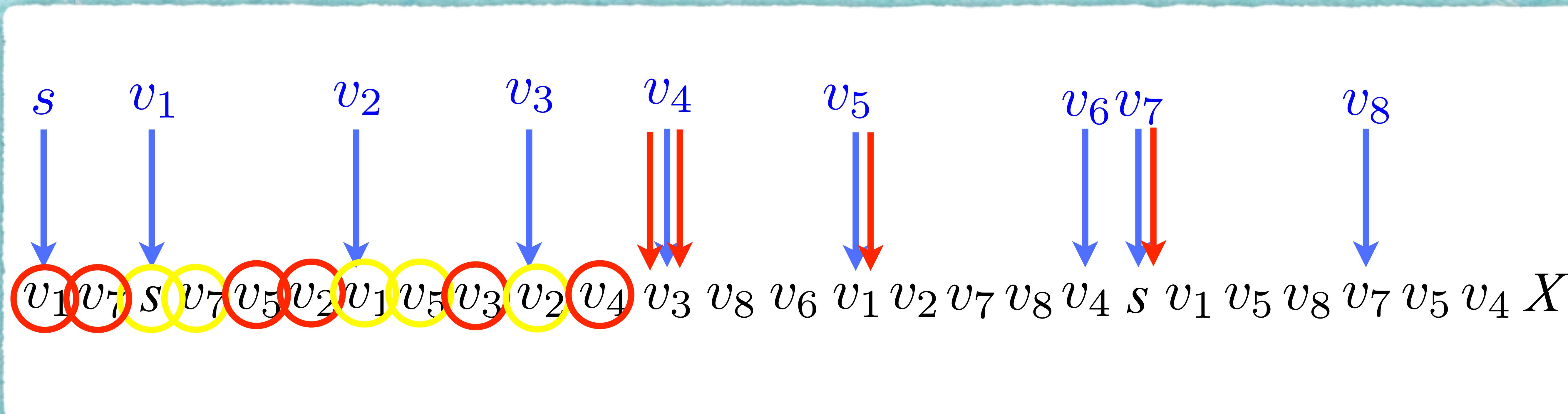
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

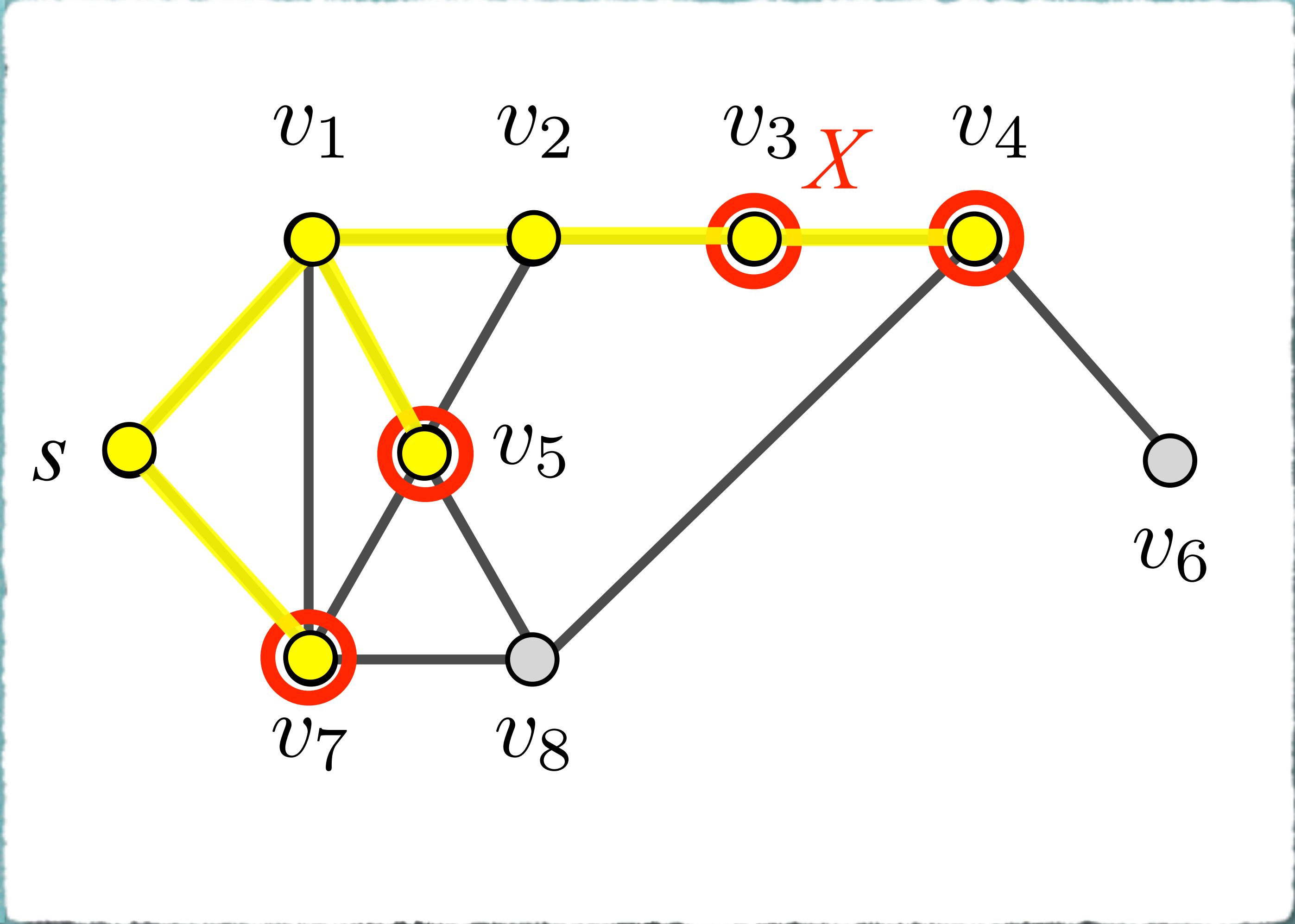
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

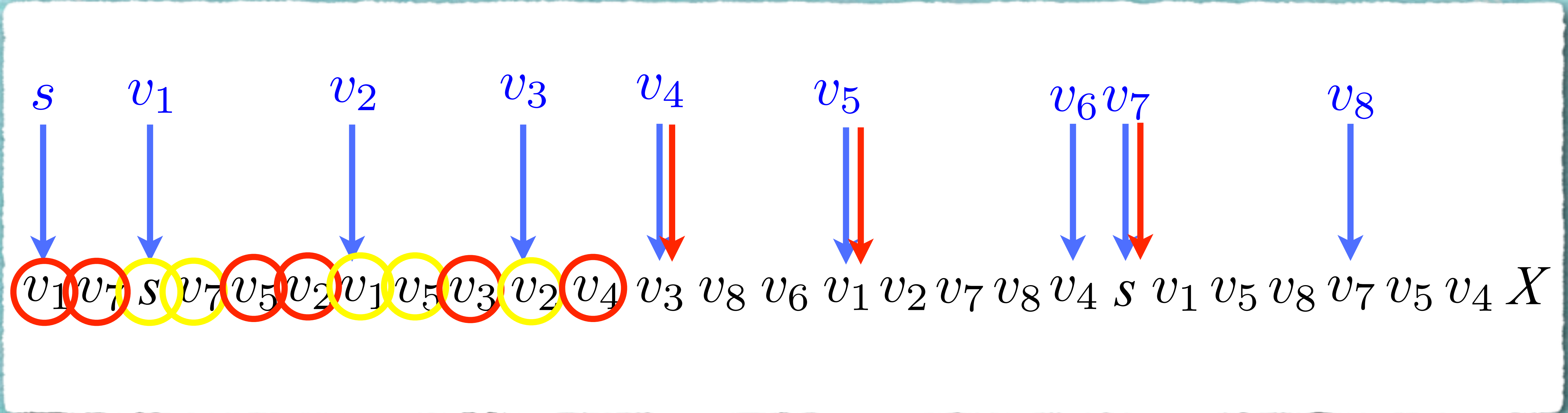
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

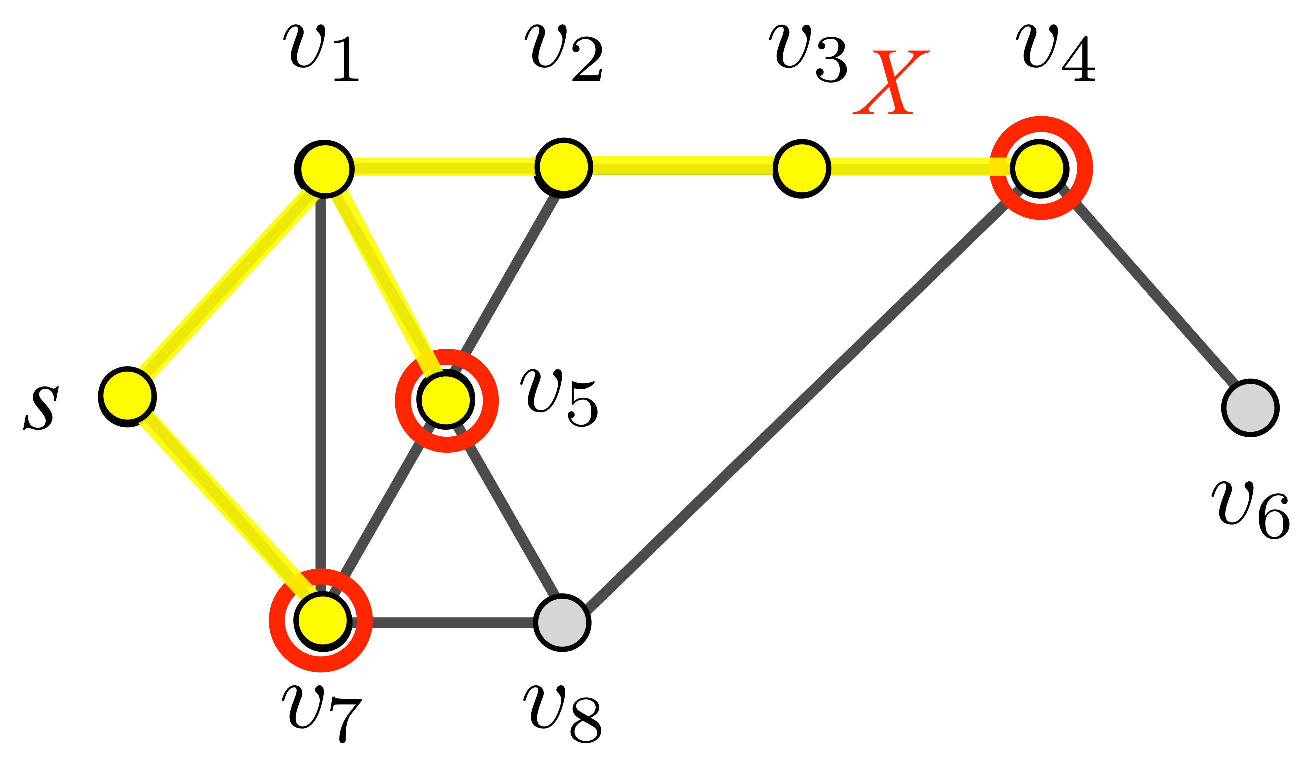
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

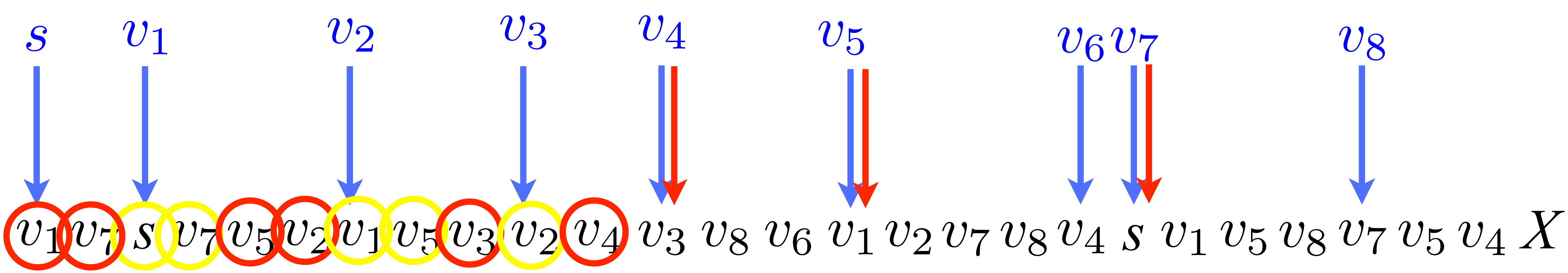
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





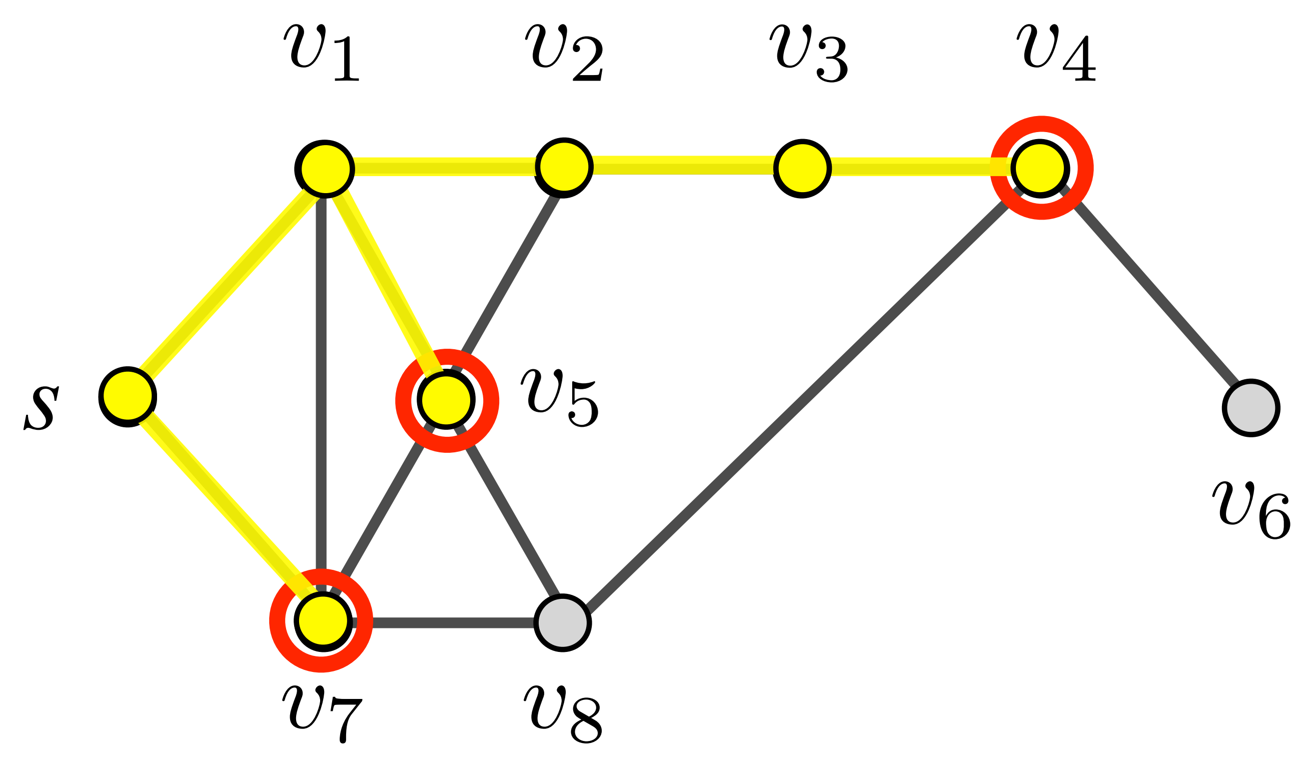
Algorithmus 3.7

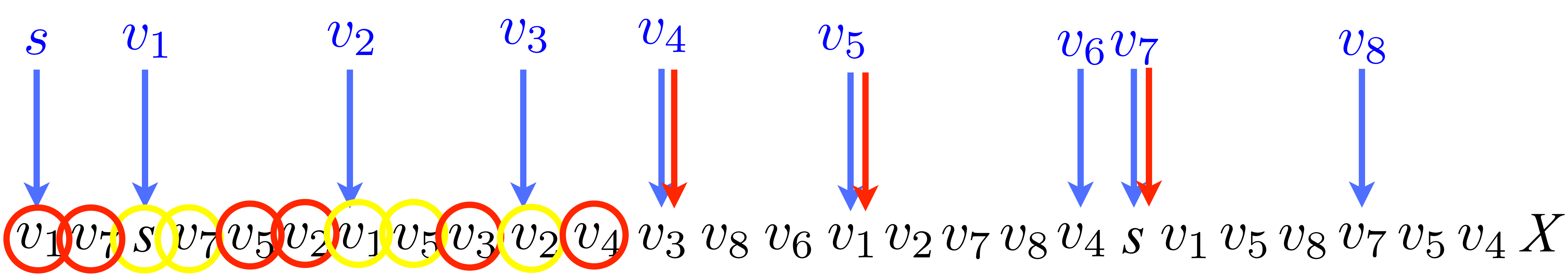
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP



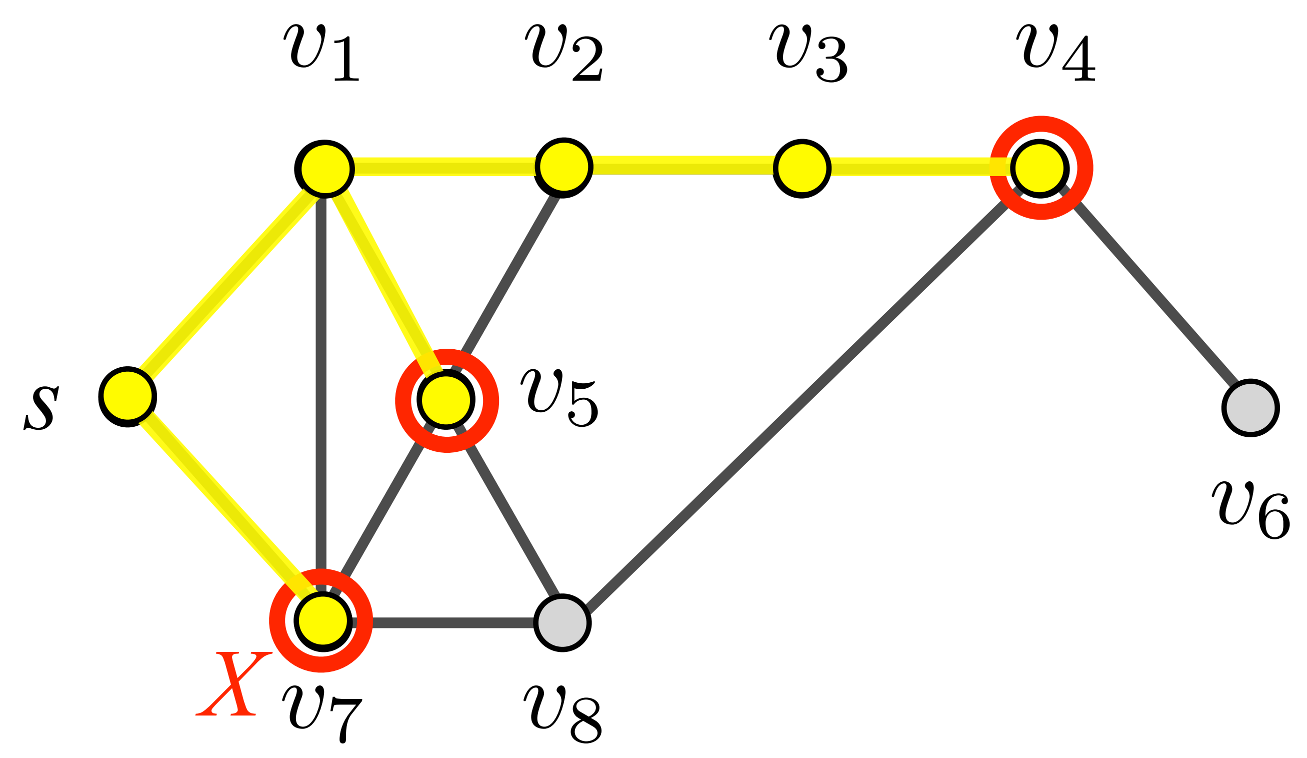


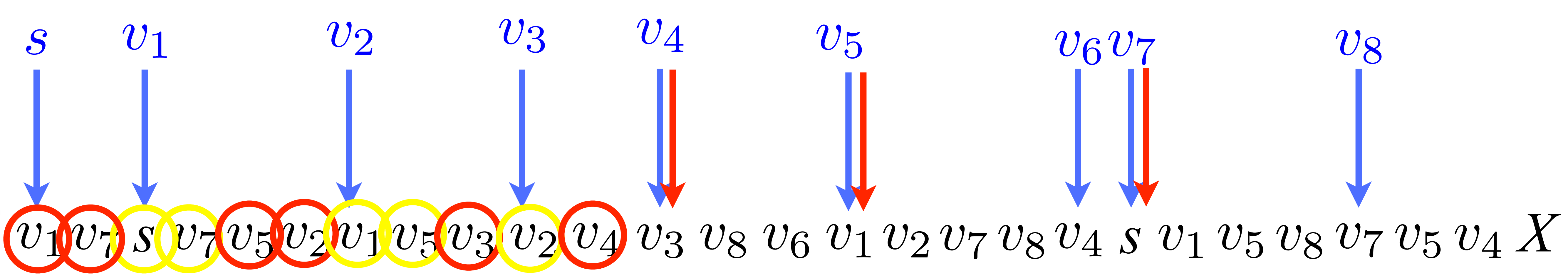
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }

   }
3. STOP



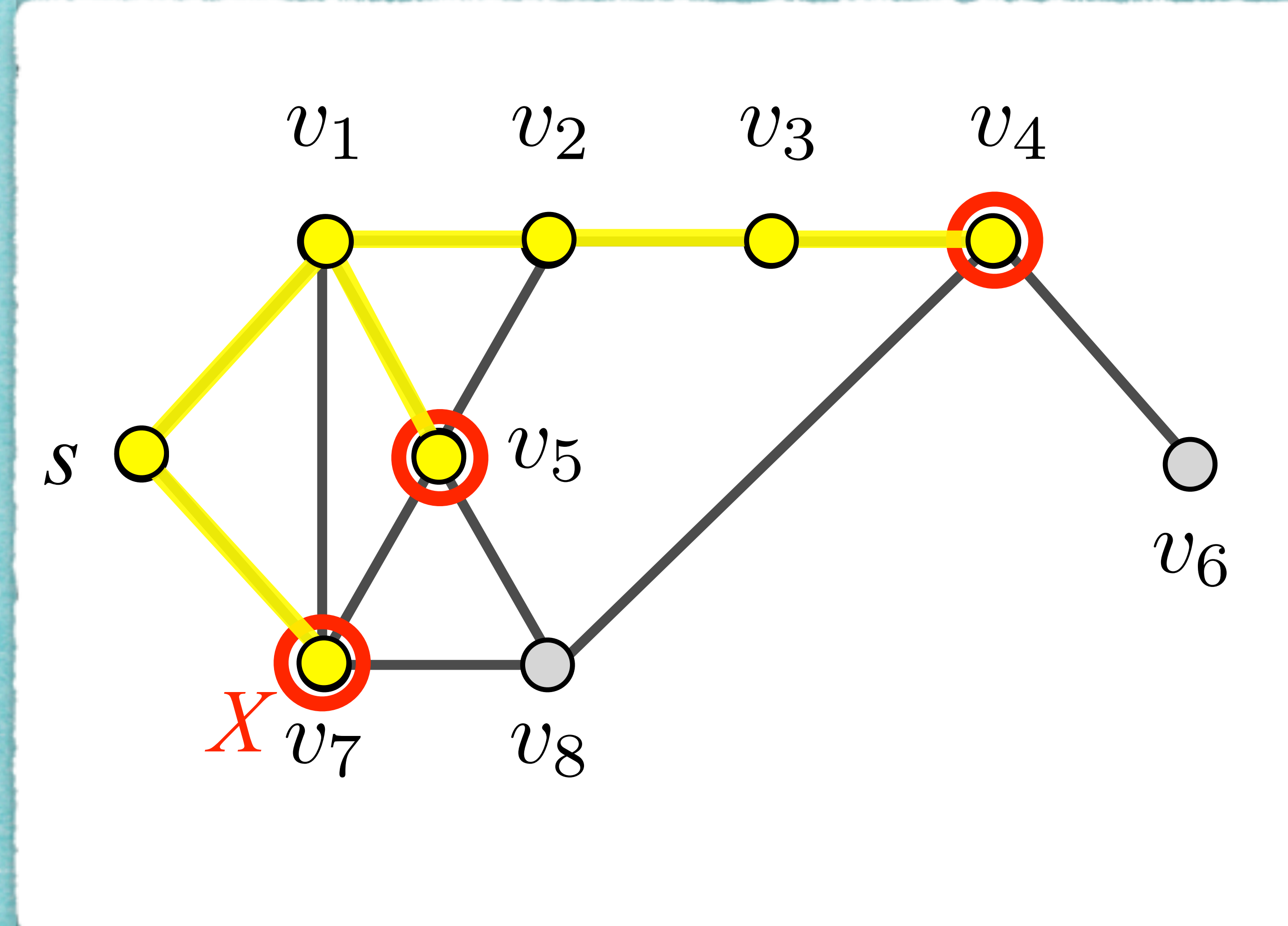


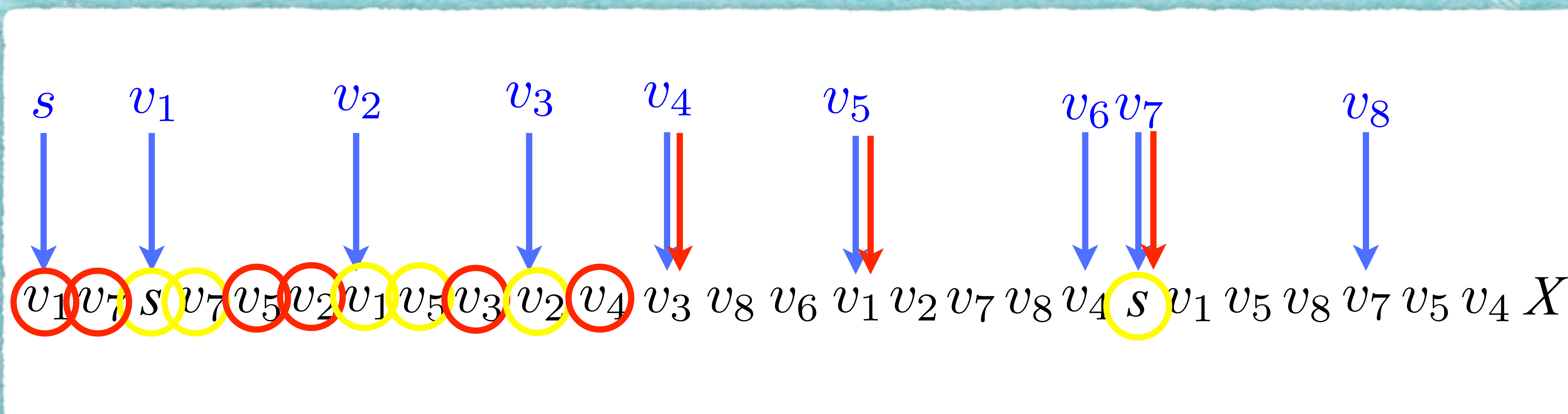
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP





Algorithmus 3.7

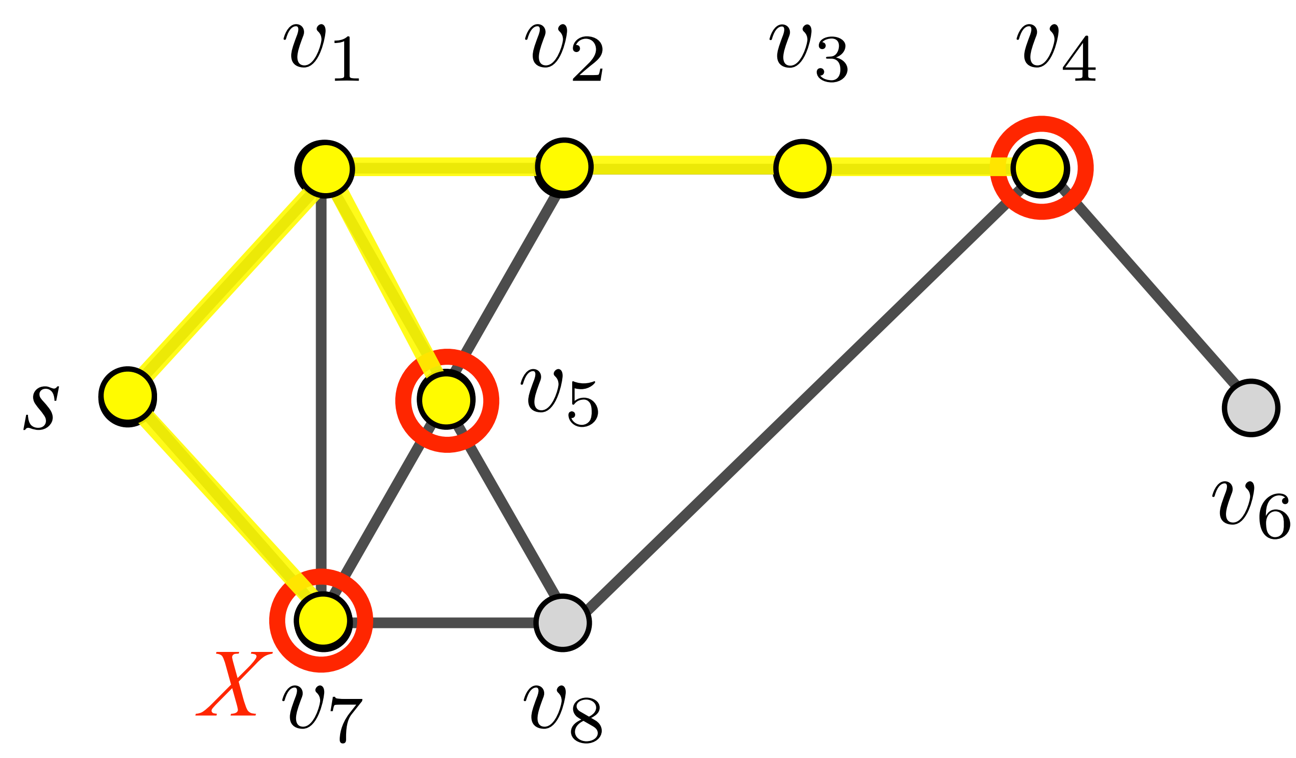
INPUT: Graph $G = (V,E)$, Knoten s

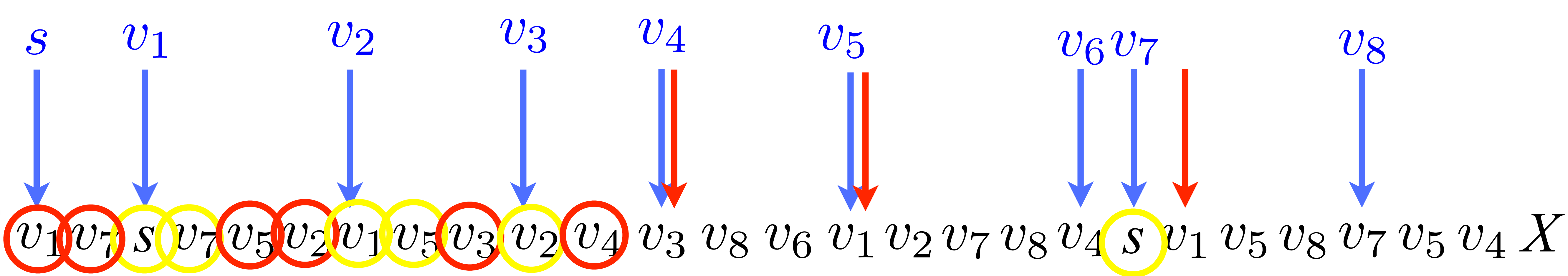
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

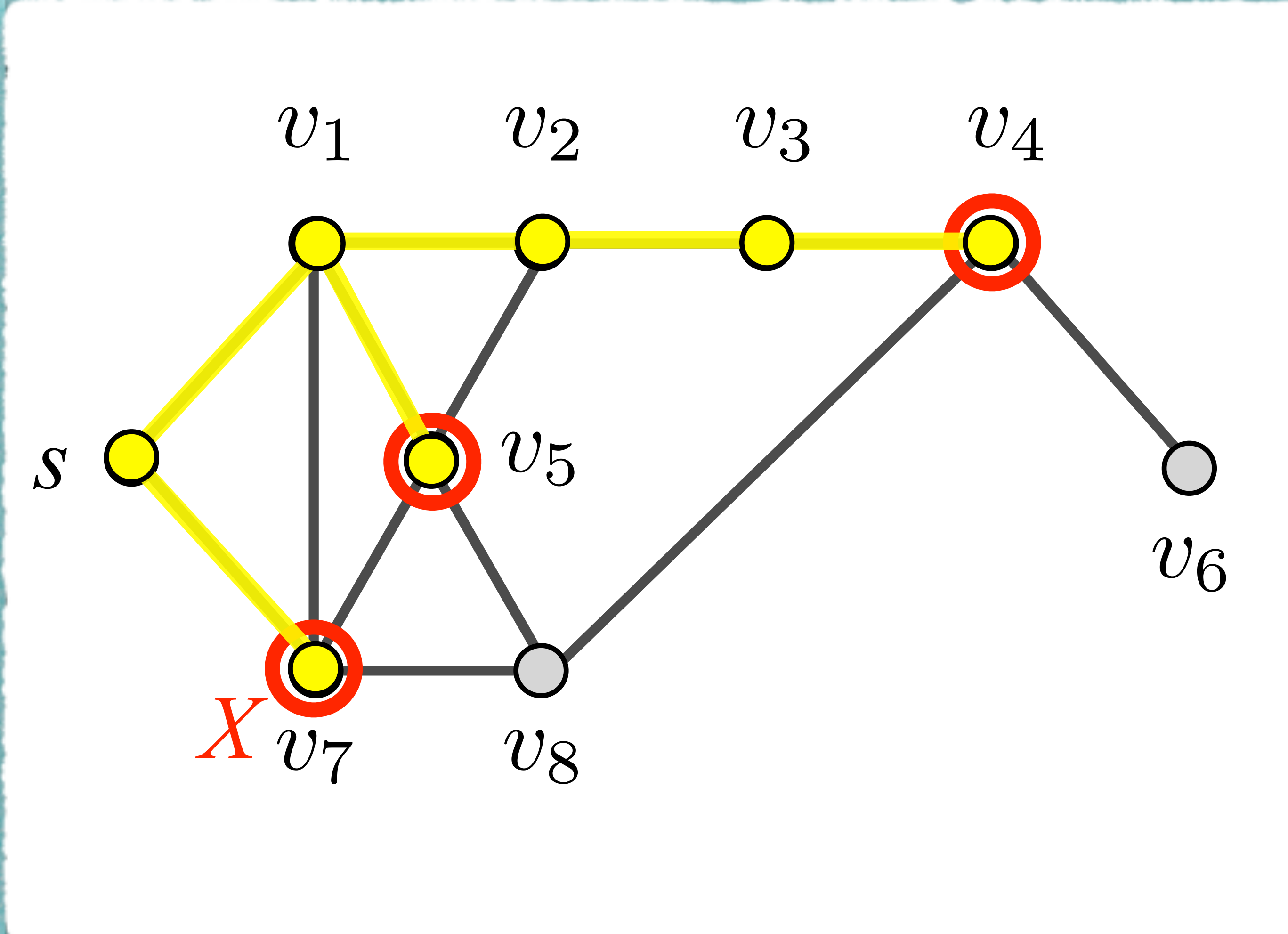
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

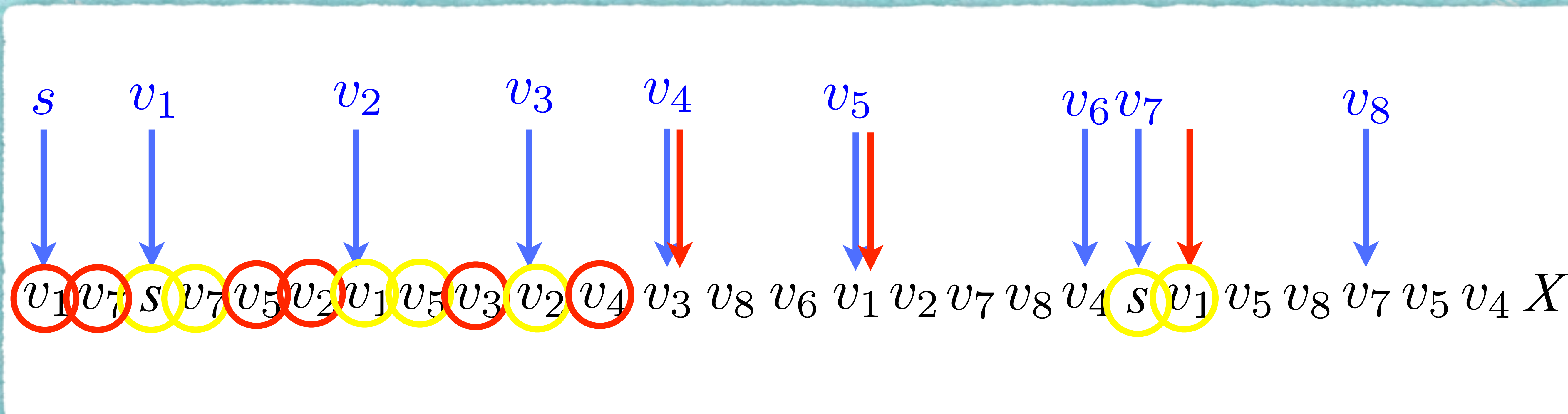
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

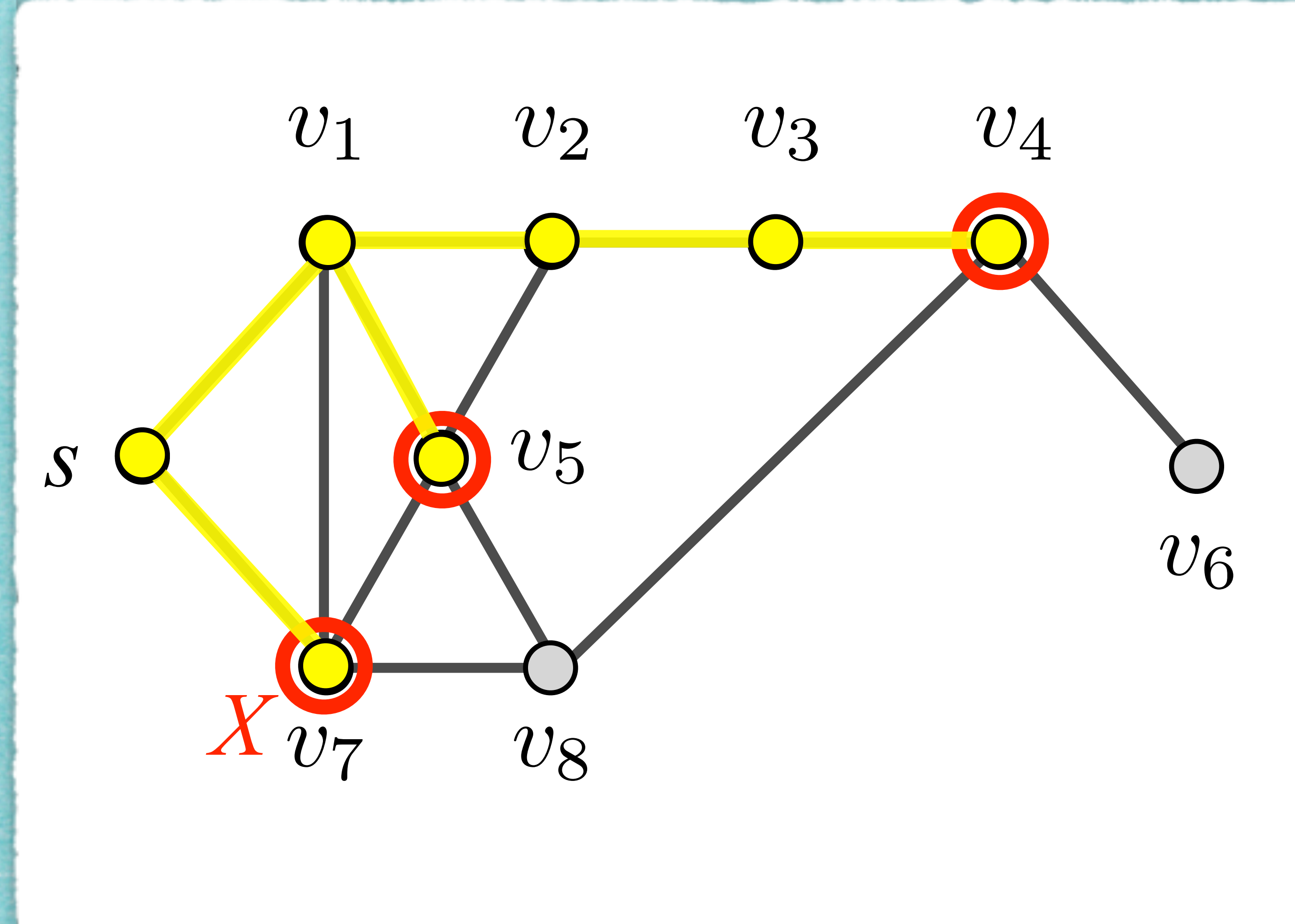


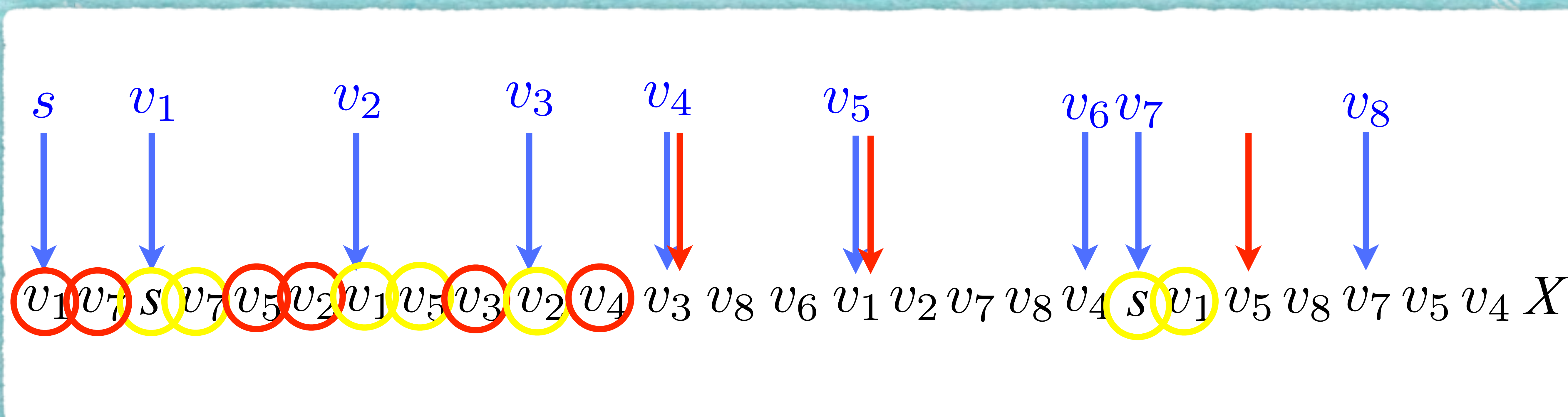


Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
         2.2.1. R:=R\{v}
      2.3. ELSE {
         2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
         2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
         }
    }
3.  STOP
```



Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

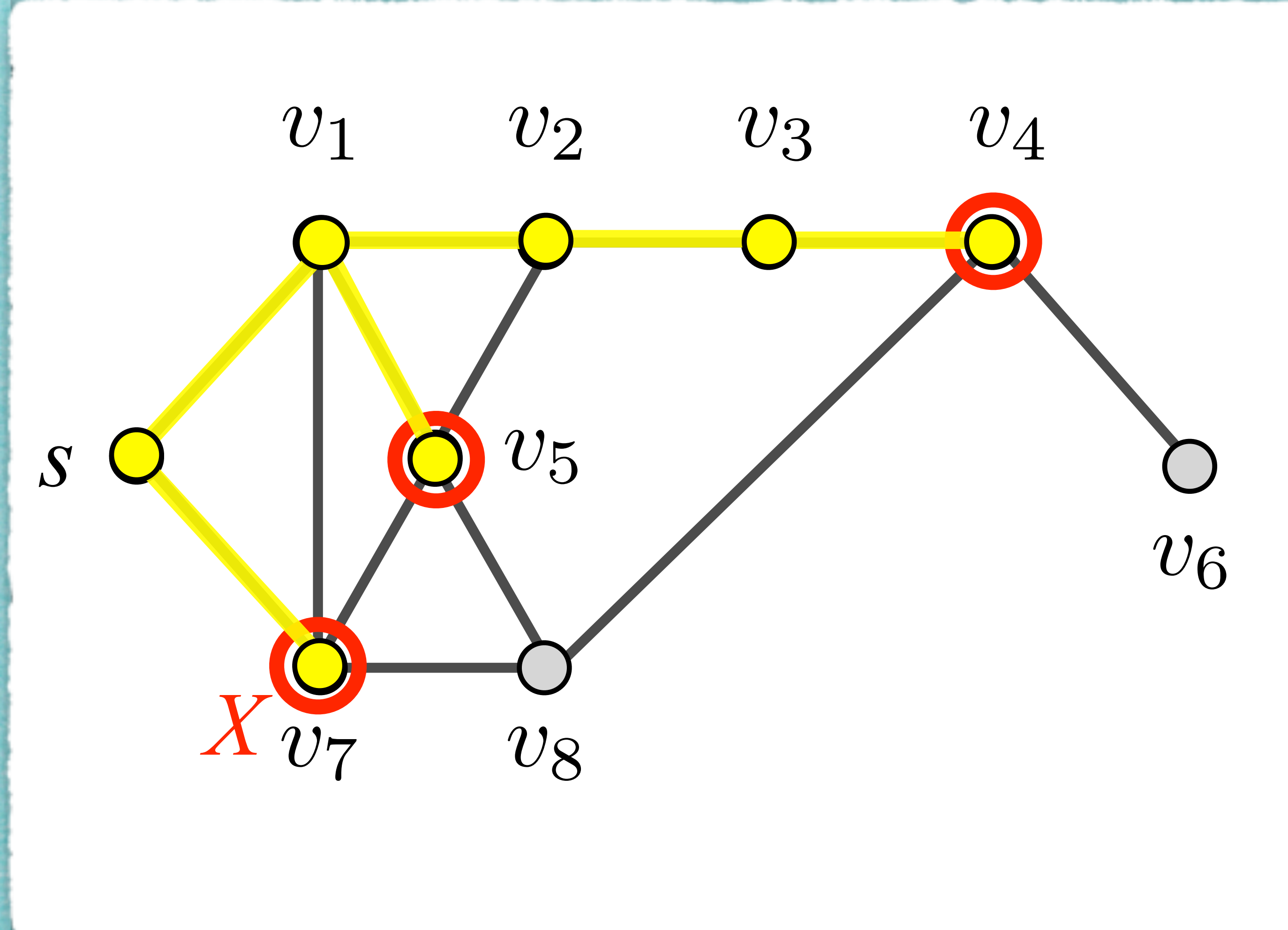
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

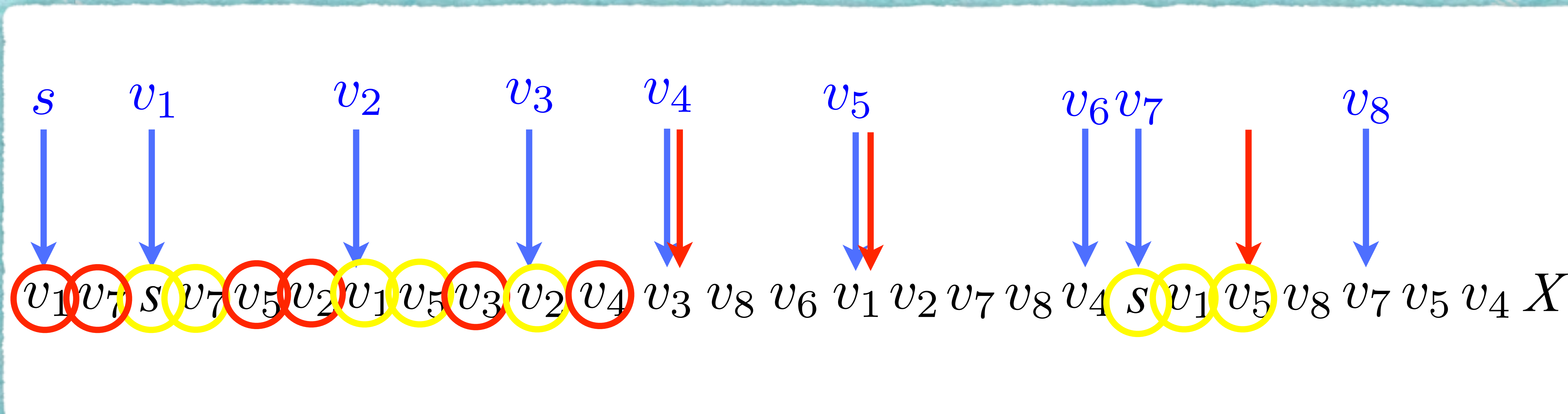
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP

2





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

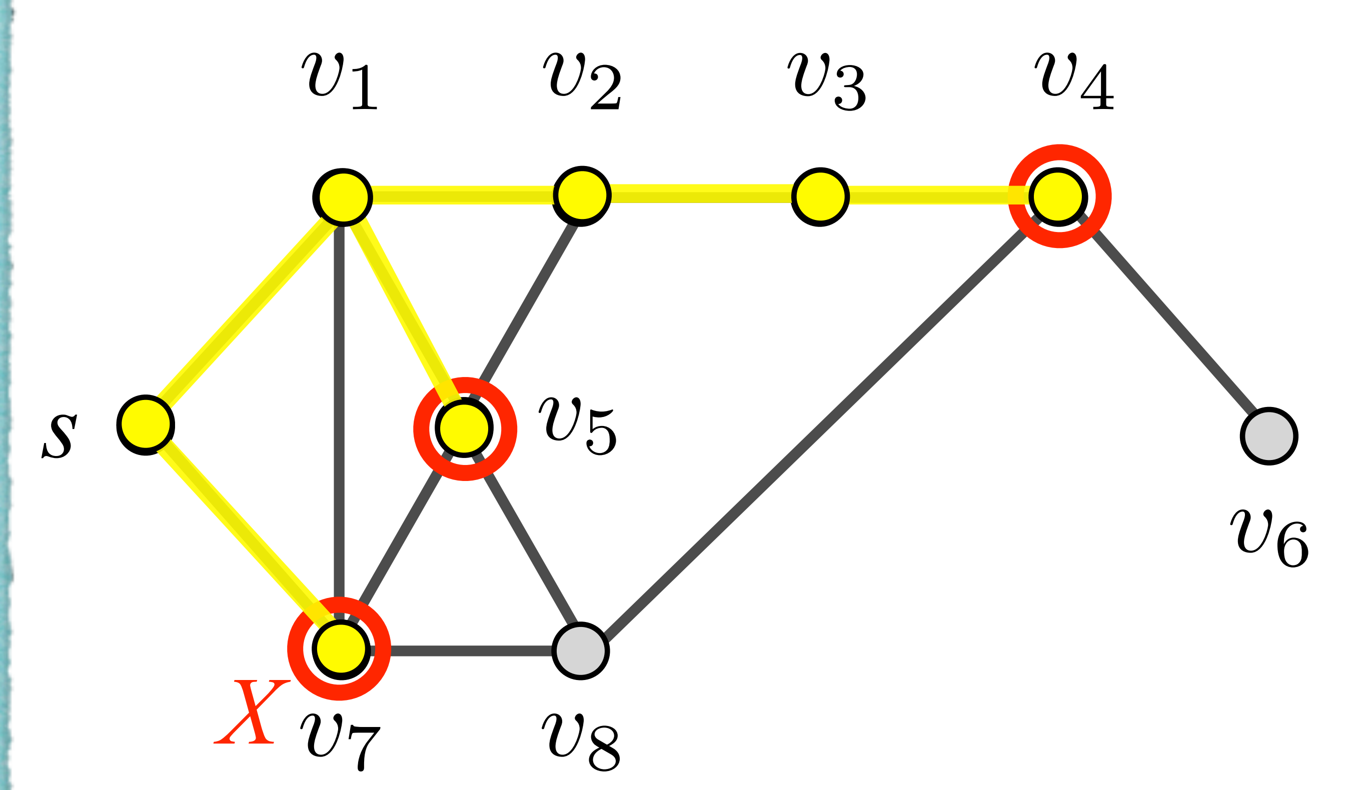
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

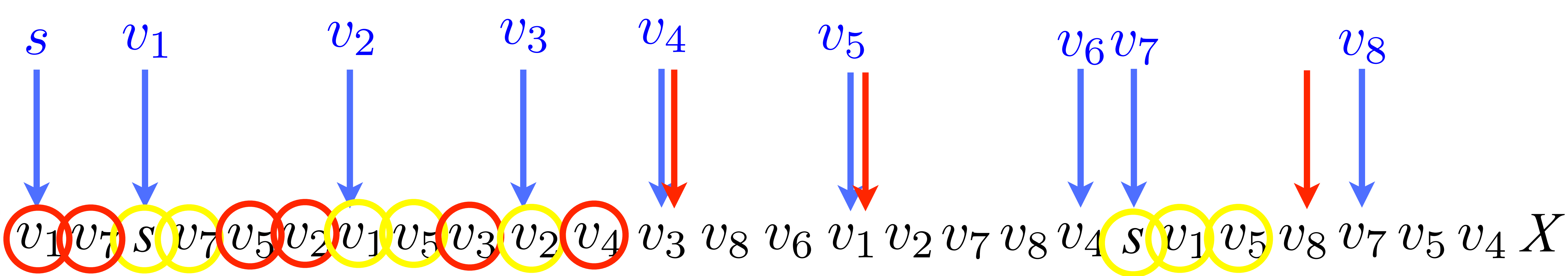
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }

3. STOP



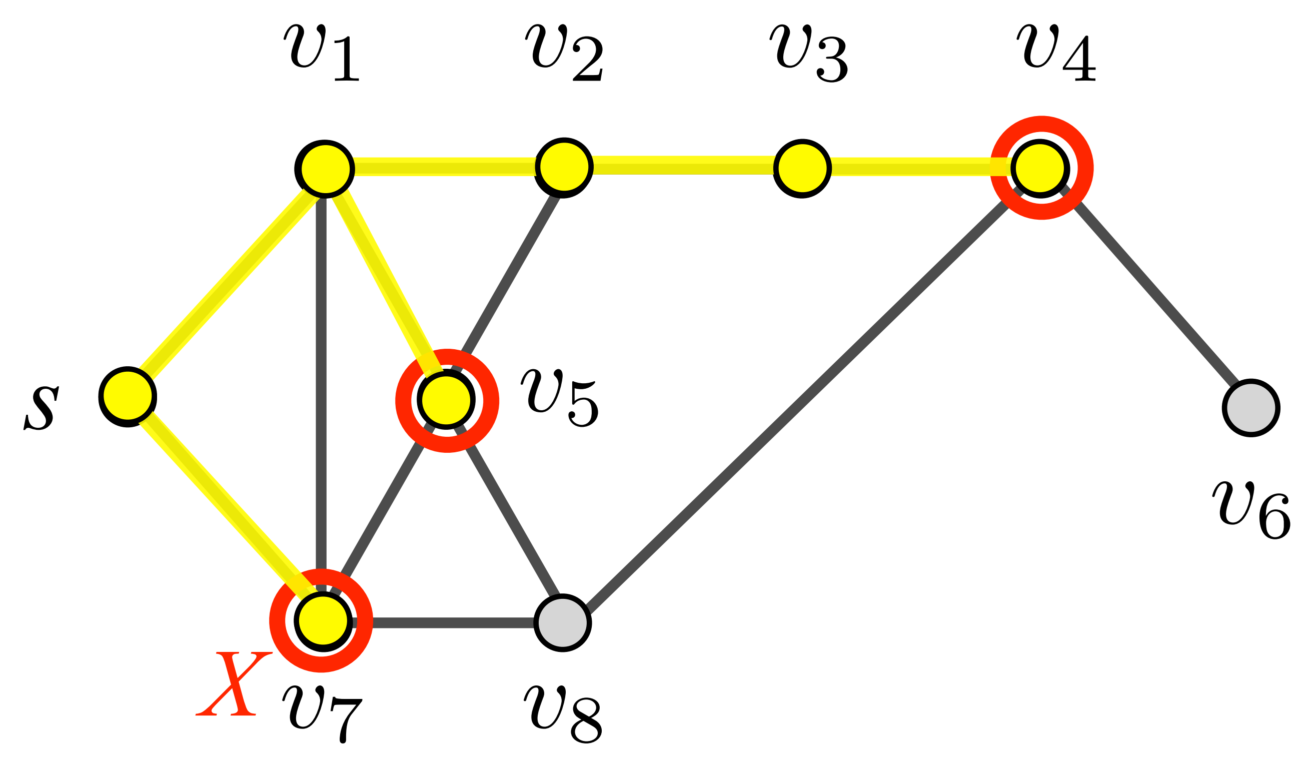


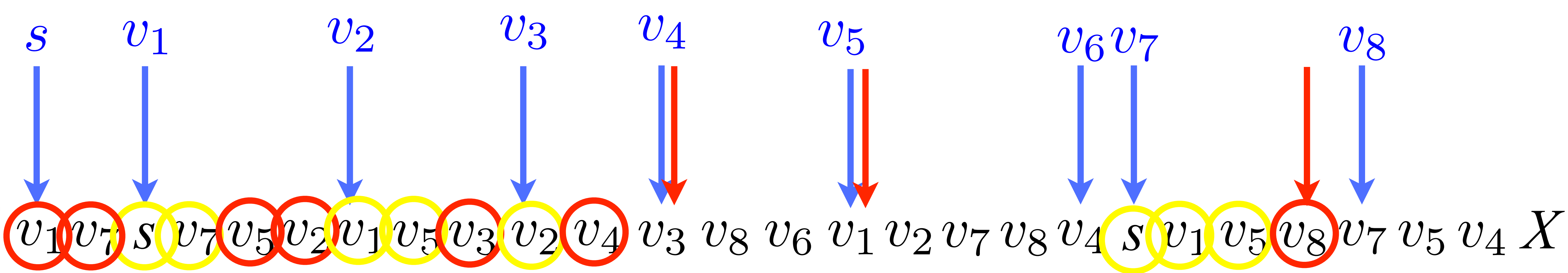
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }
   }
3. STOP

2



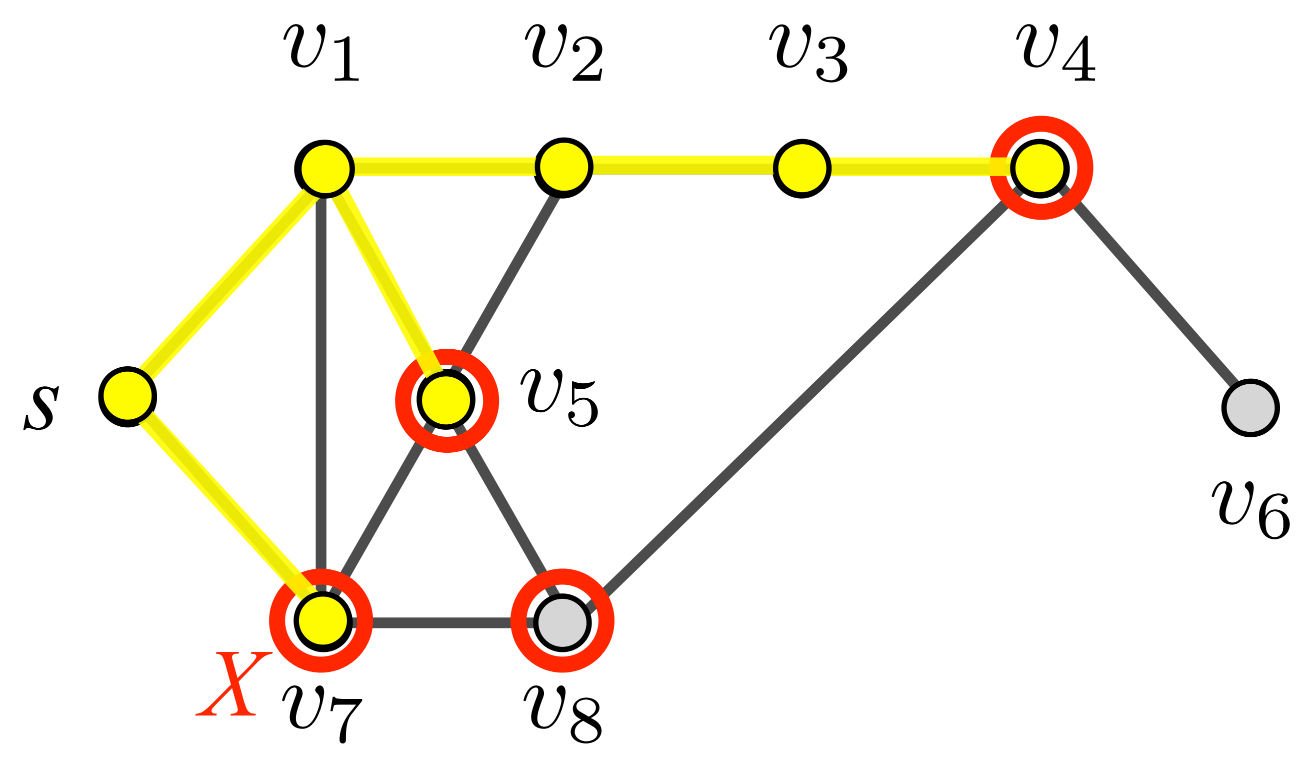


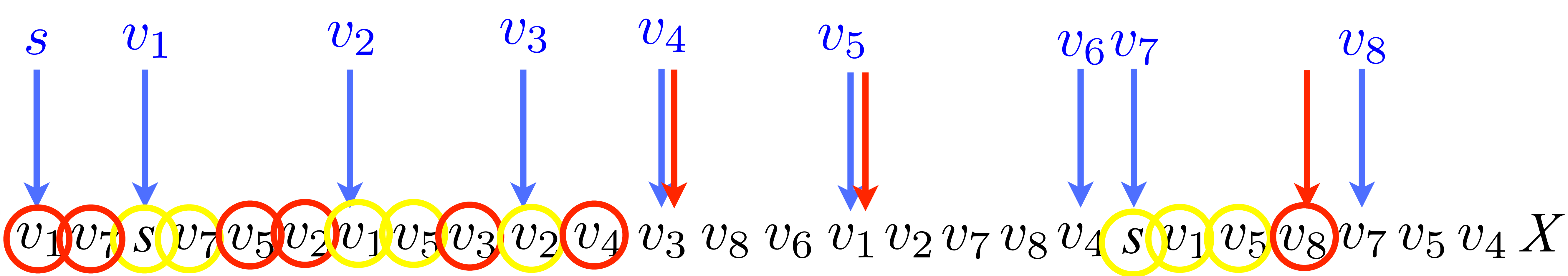
Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





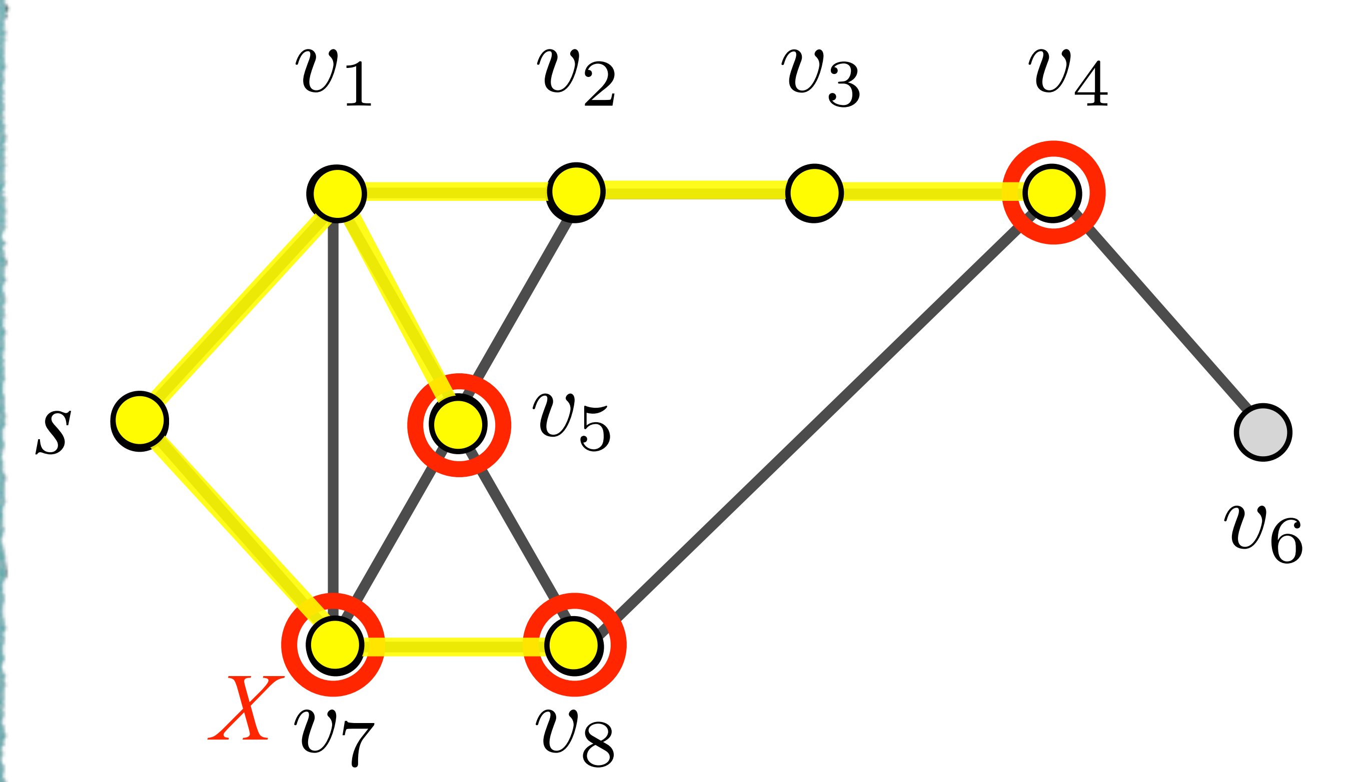
Algorithmus 3.7

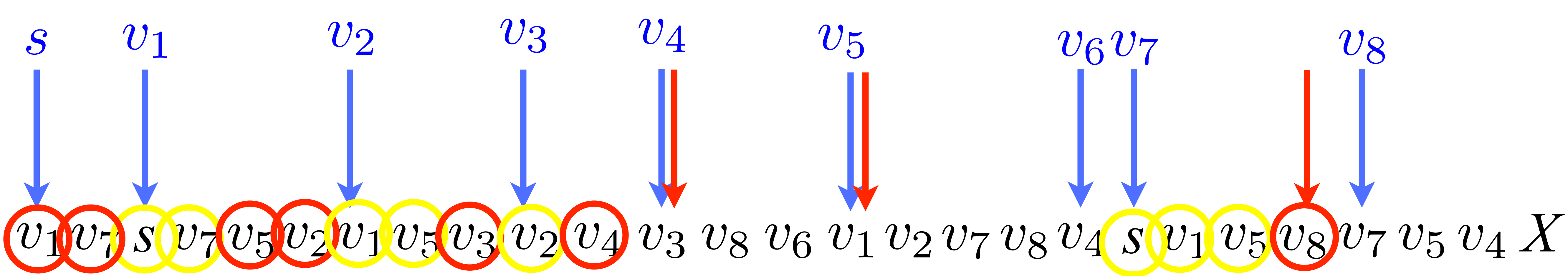
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





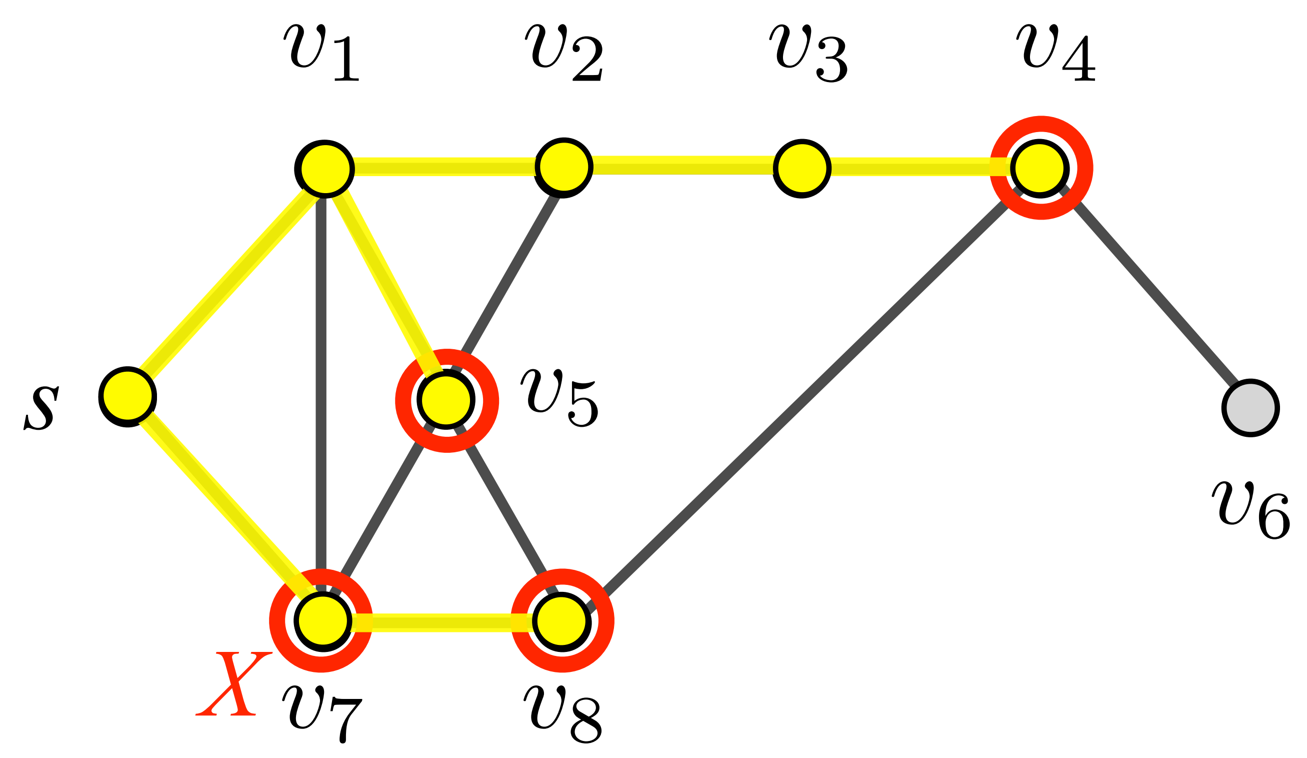
Algorithmus 3.7

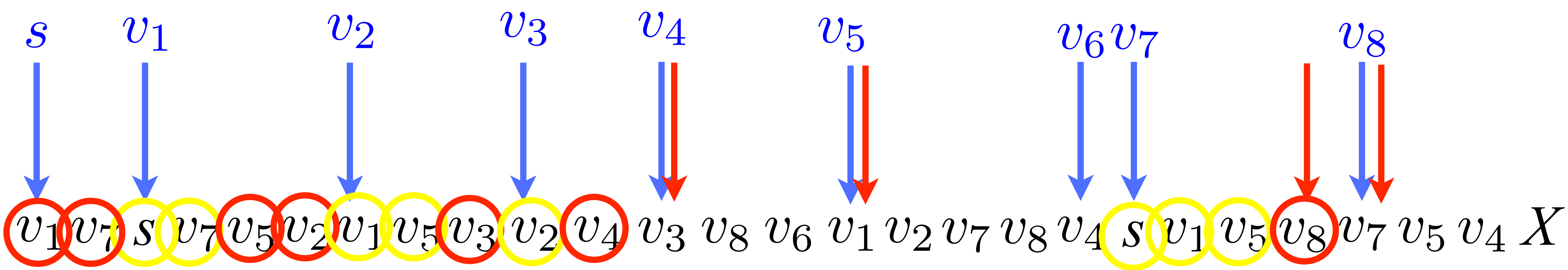
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

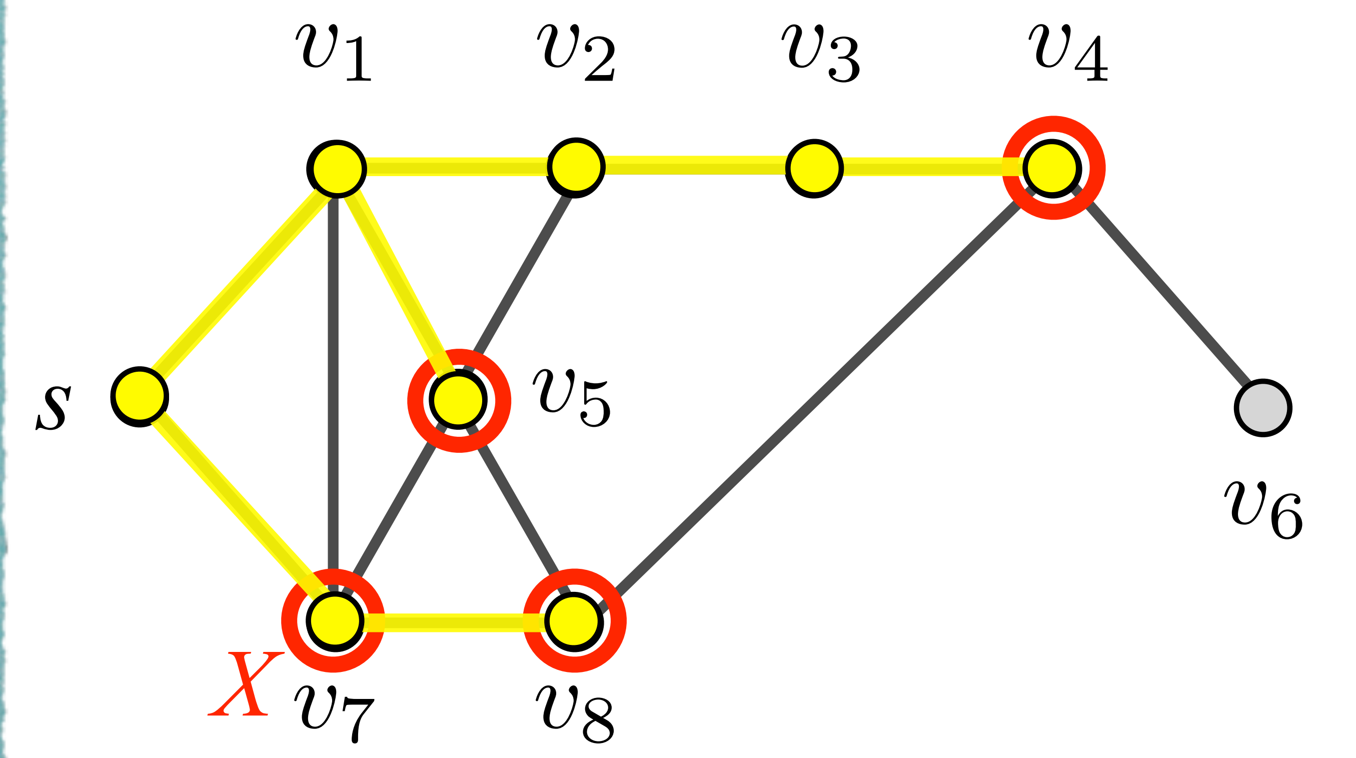
INPUT: Graph G = (V,E), Knoten s

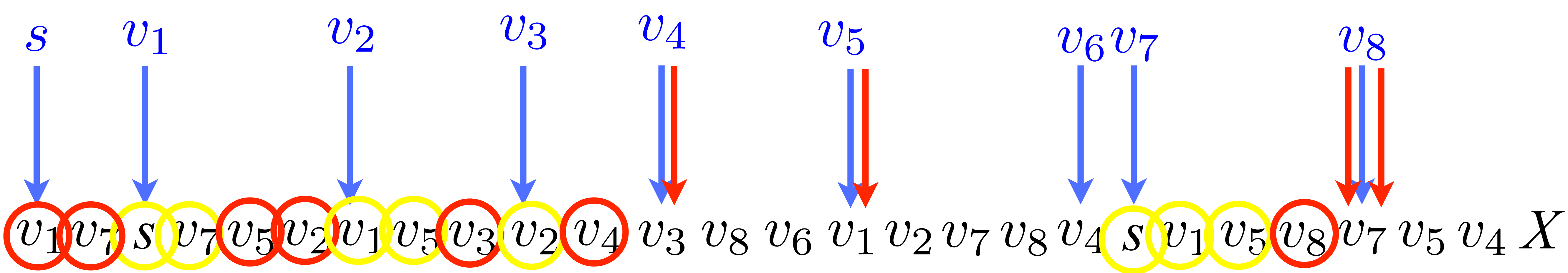
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }
   - }
3. STOP

2





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

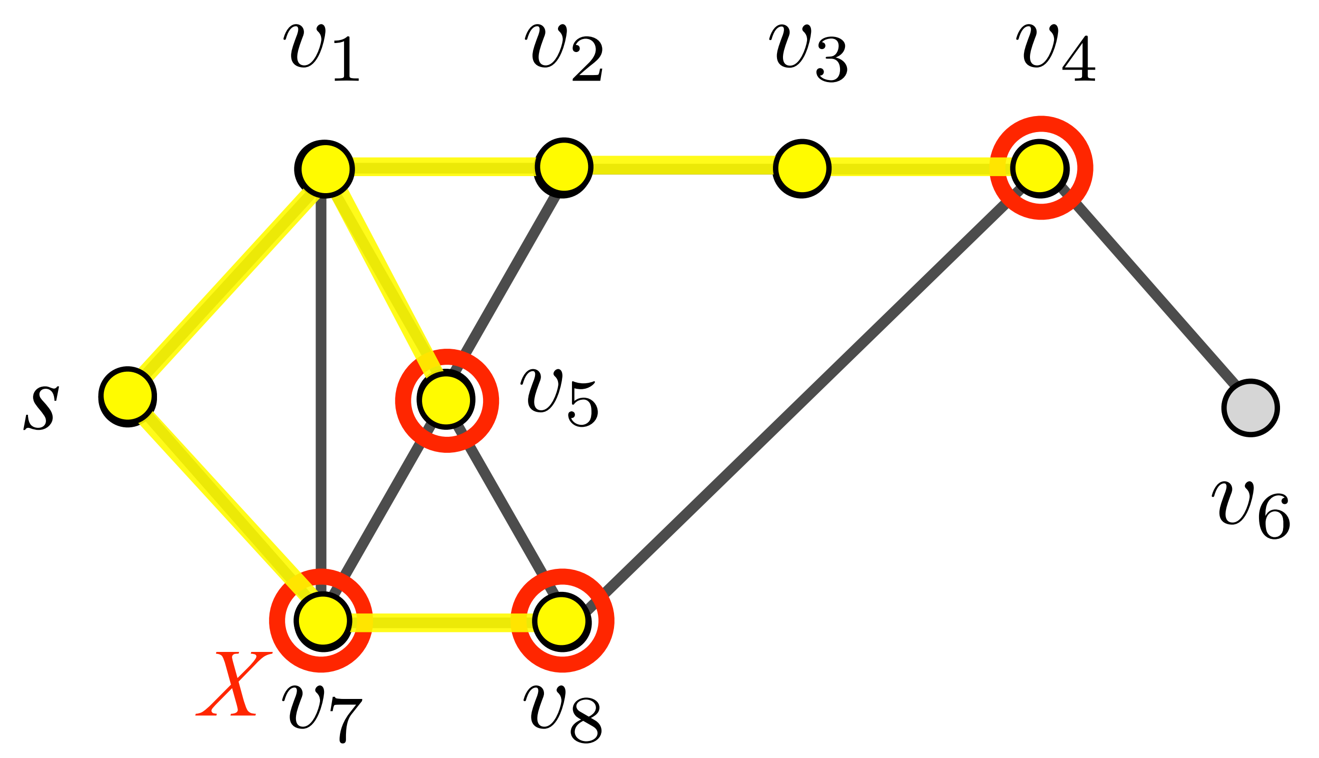
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

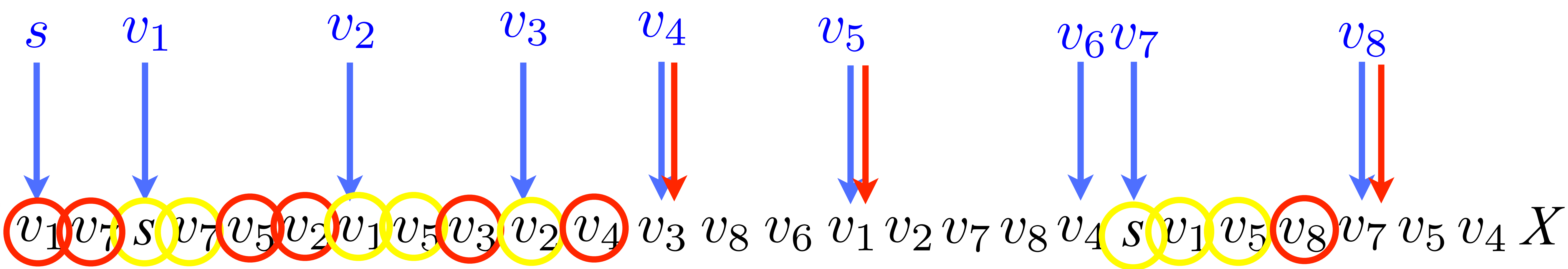
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

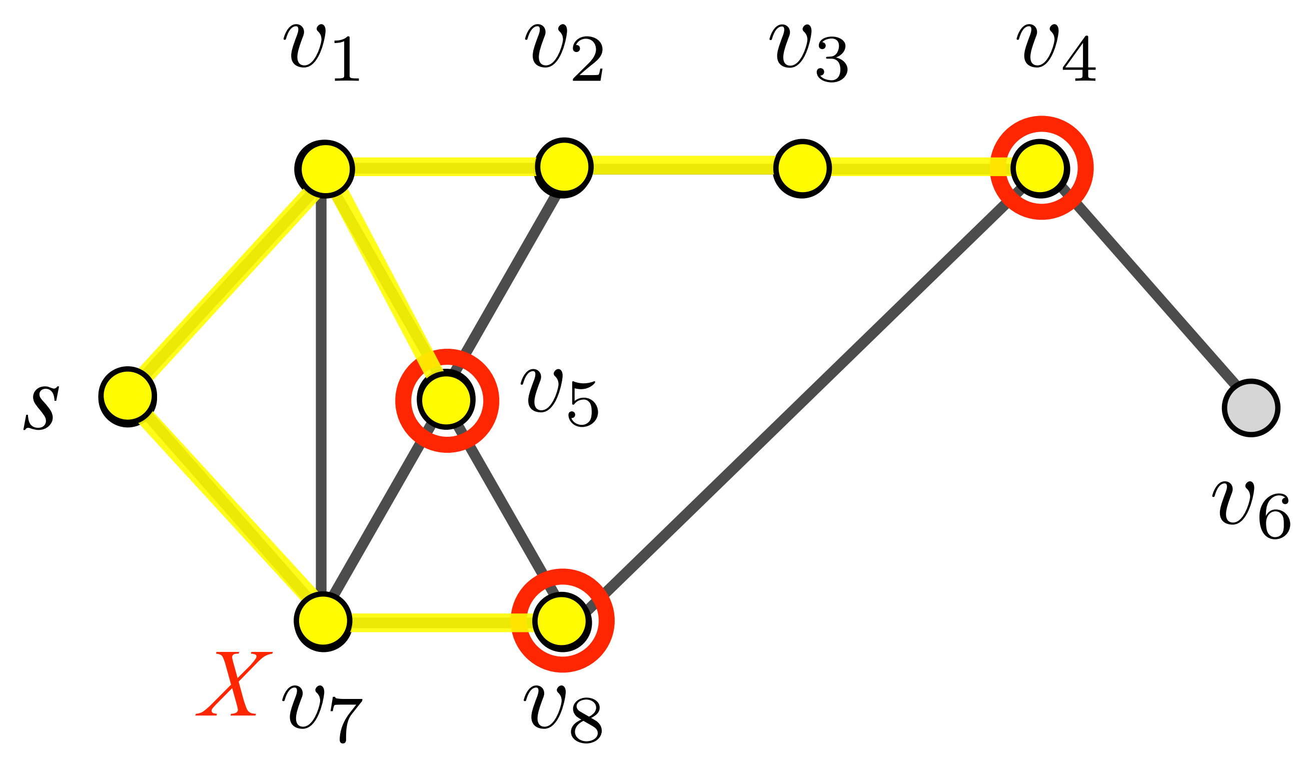
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

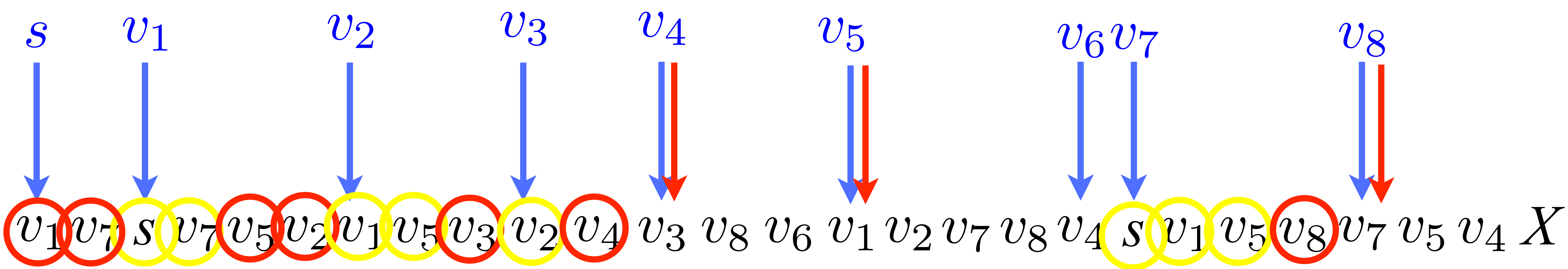
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

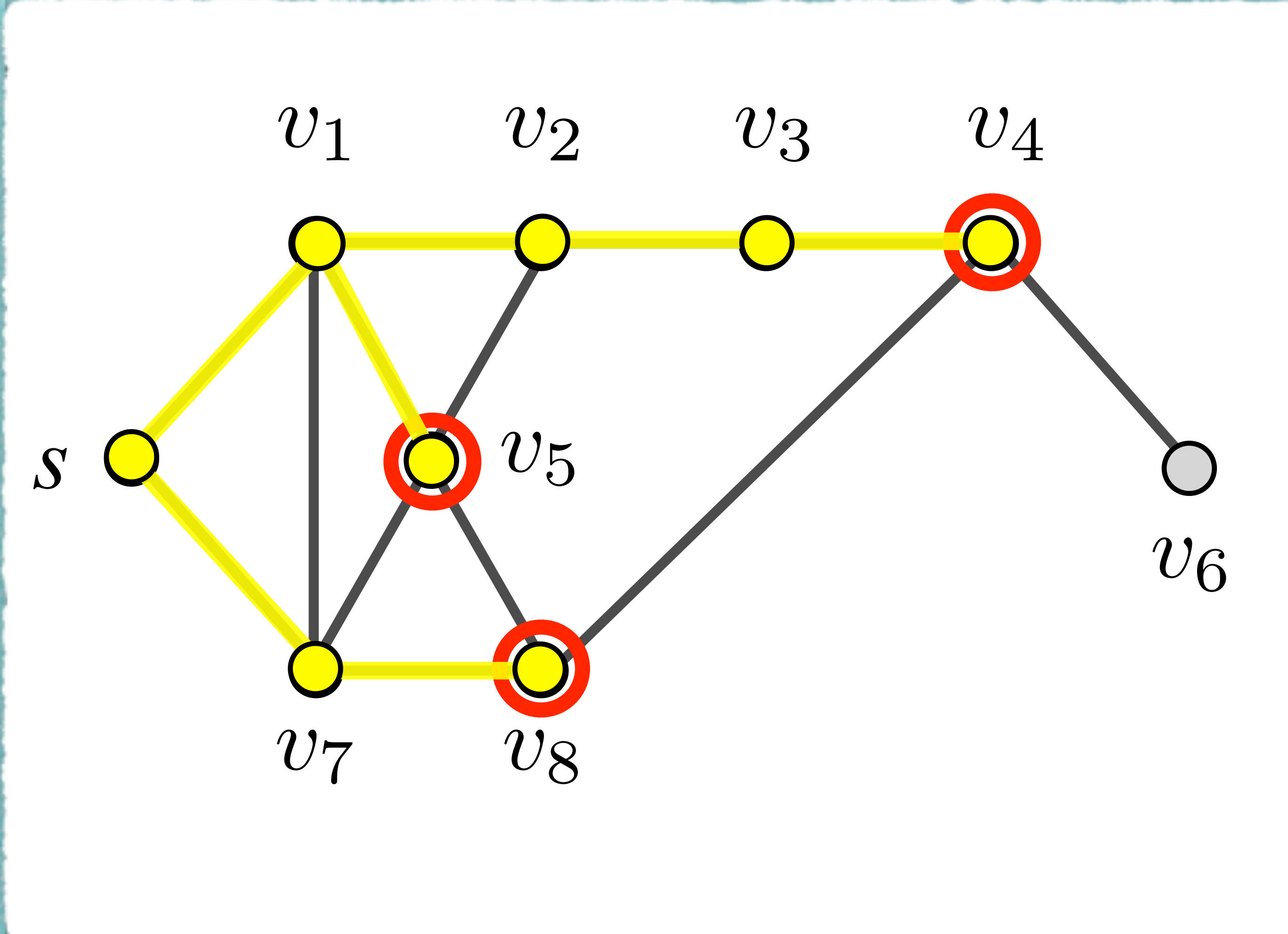


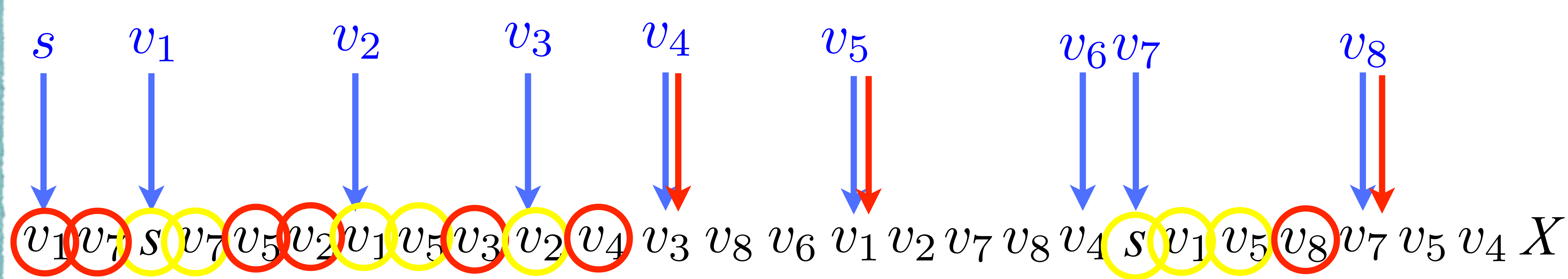


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





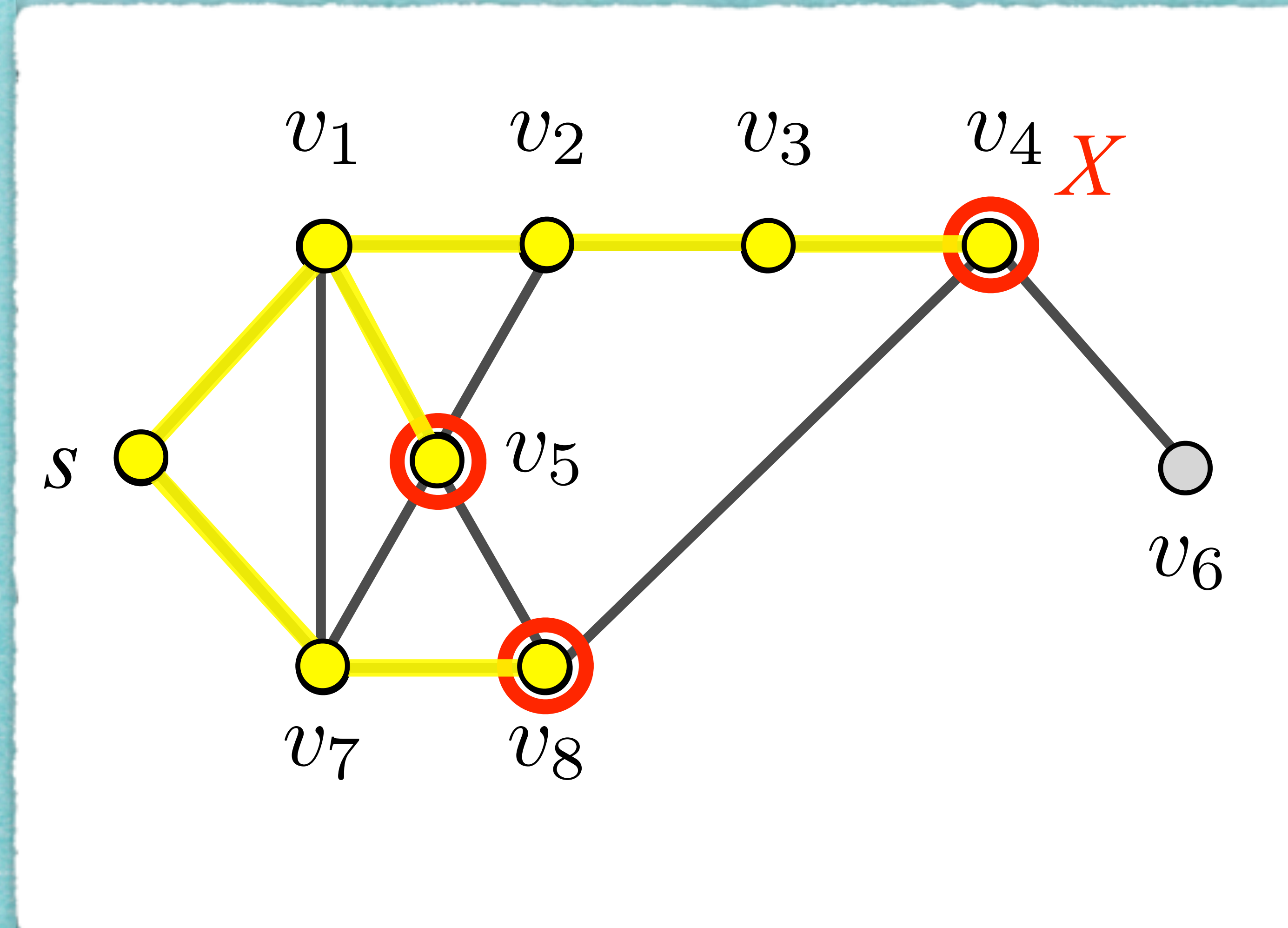
Algorithmus 3.7

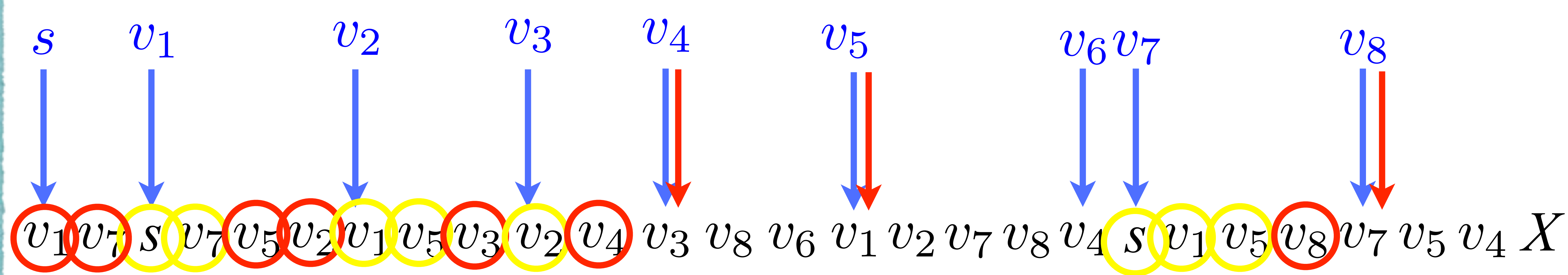
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





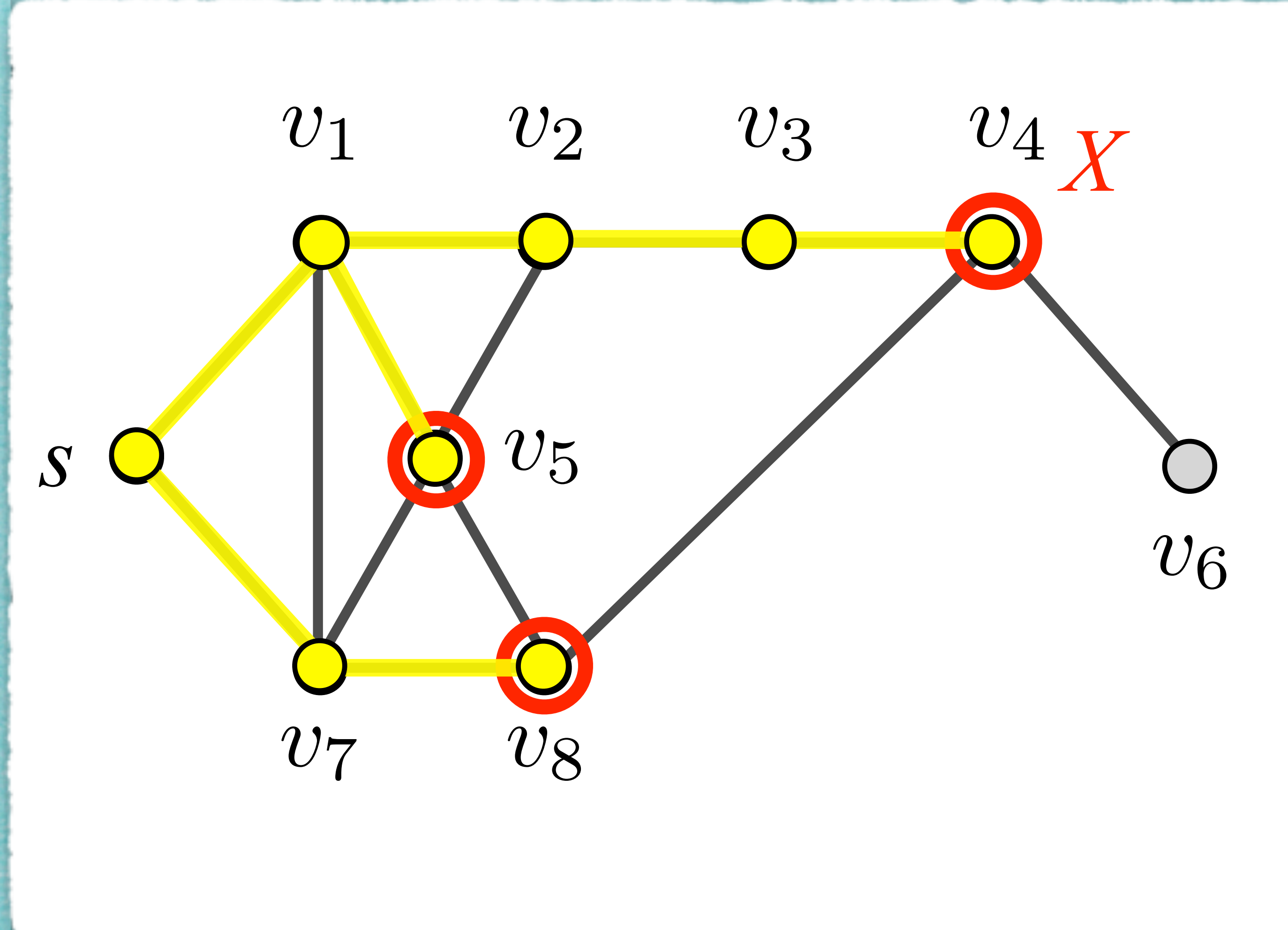
Algorithmus 3.7

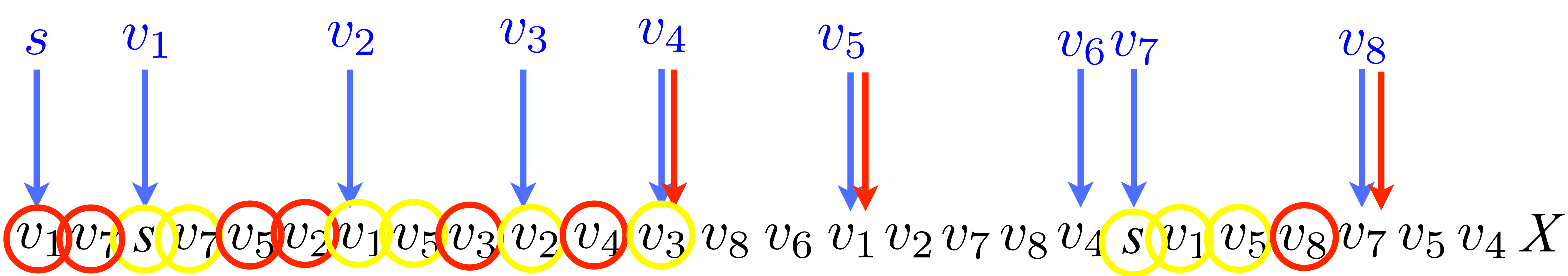
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





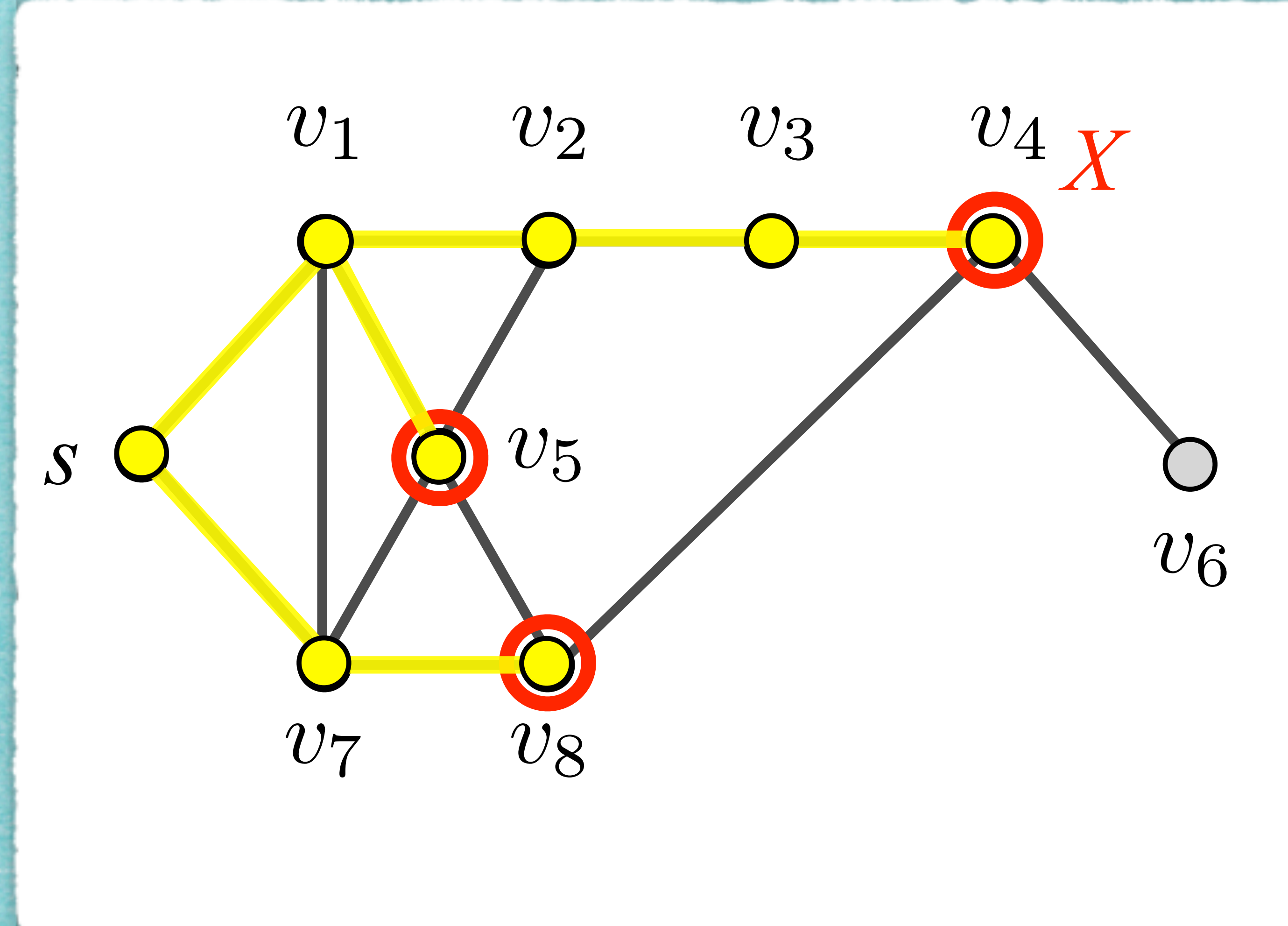
Algorithmus 3.7

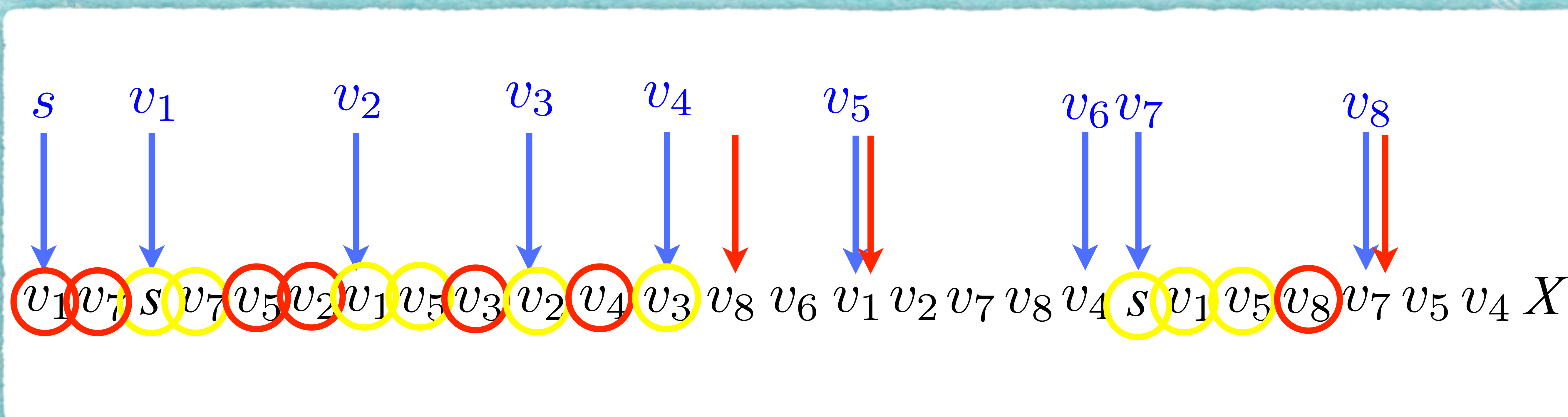
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



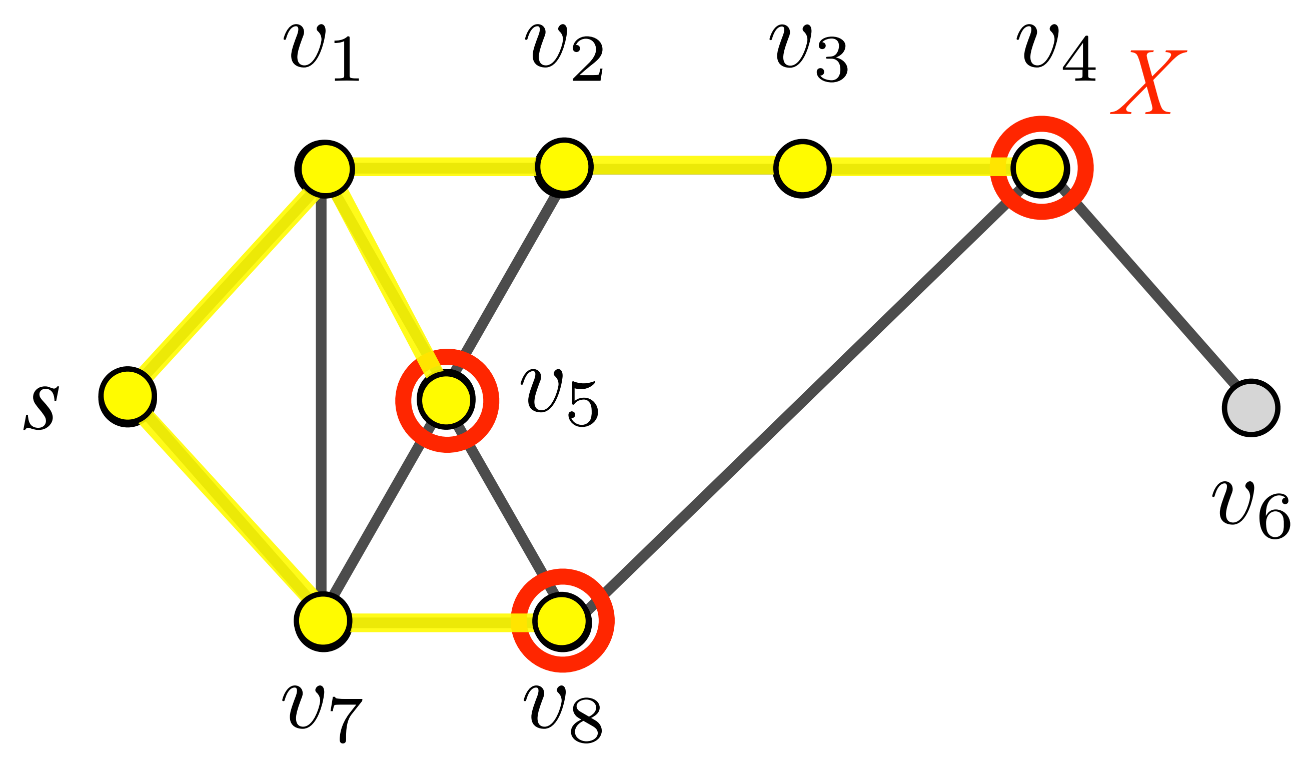


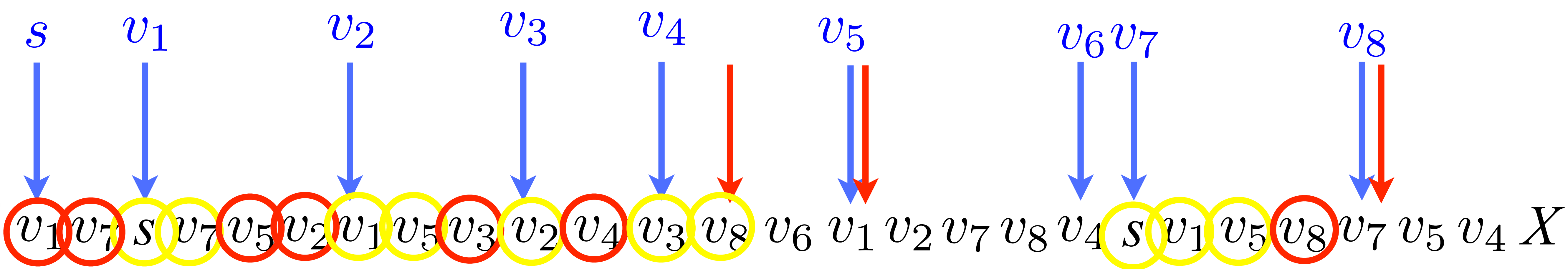
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

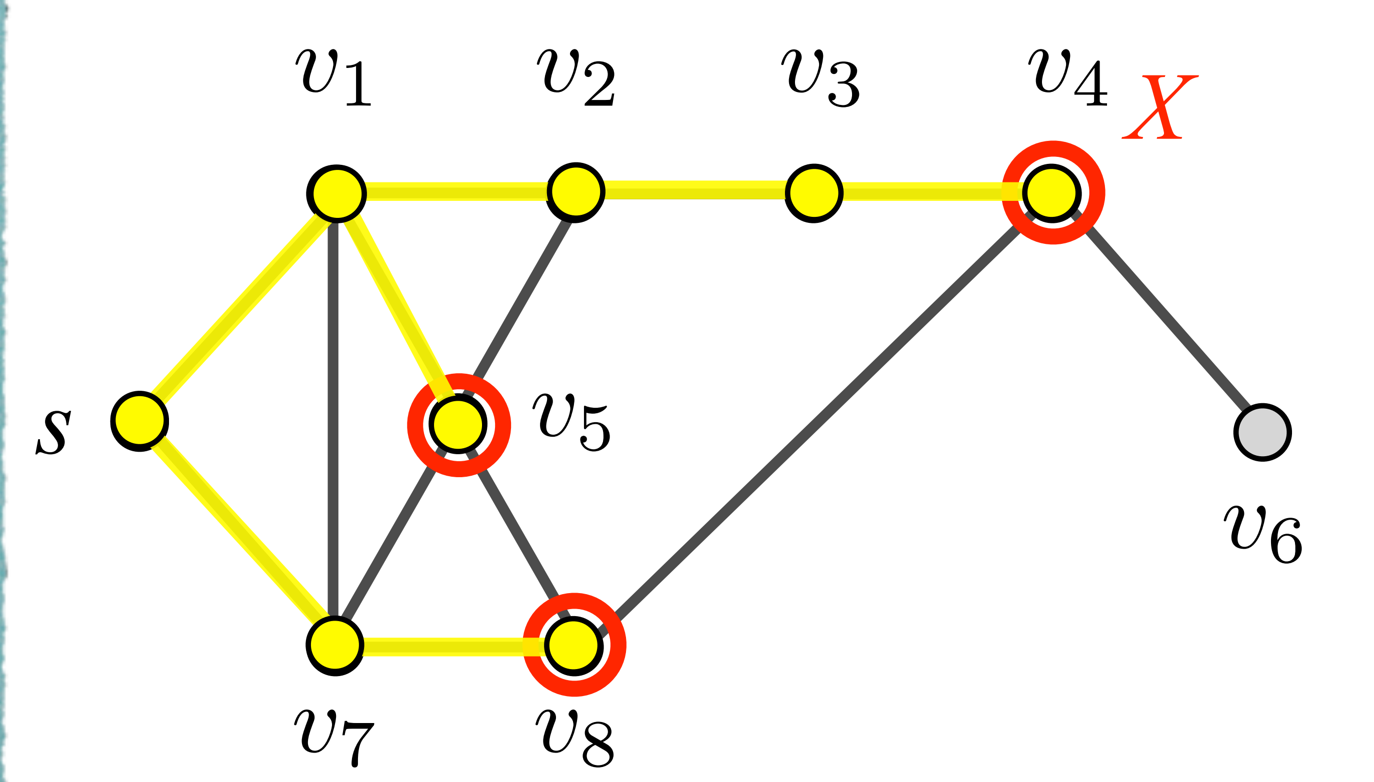
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

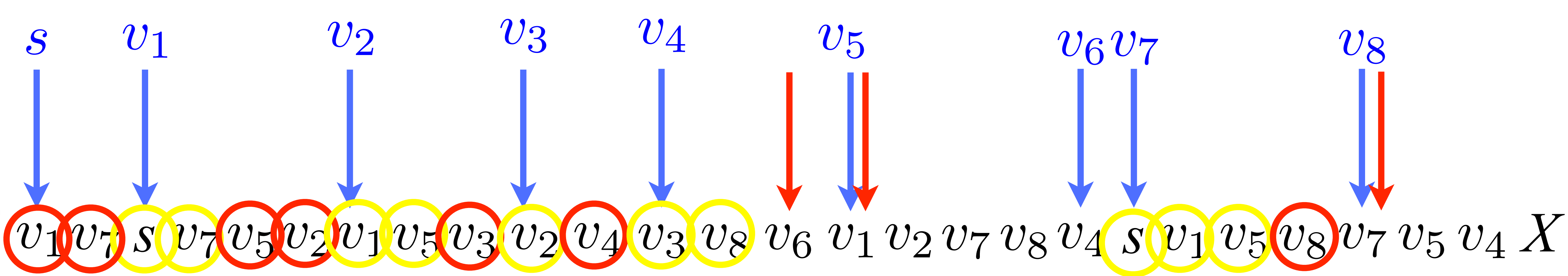
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

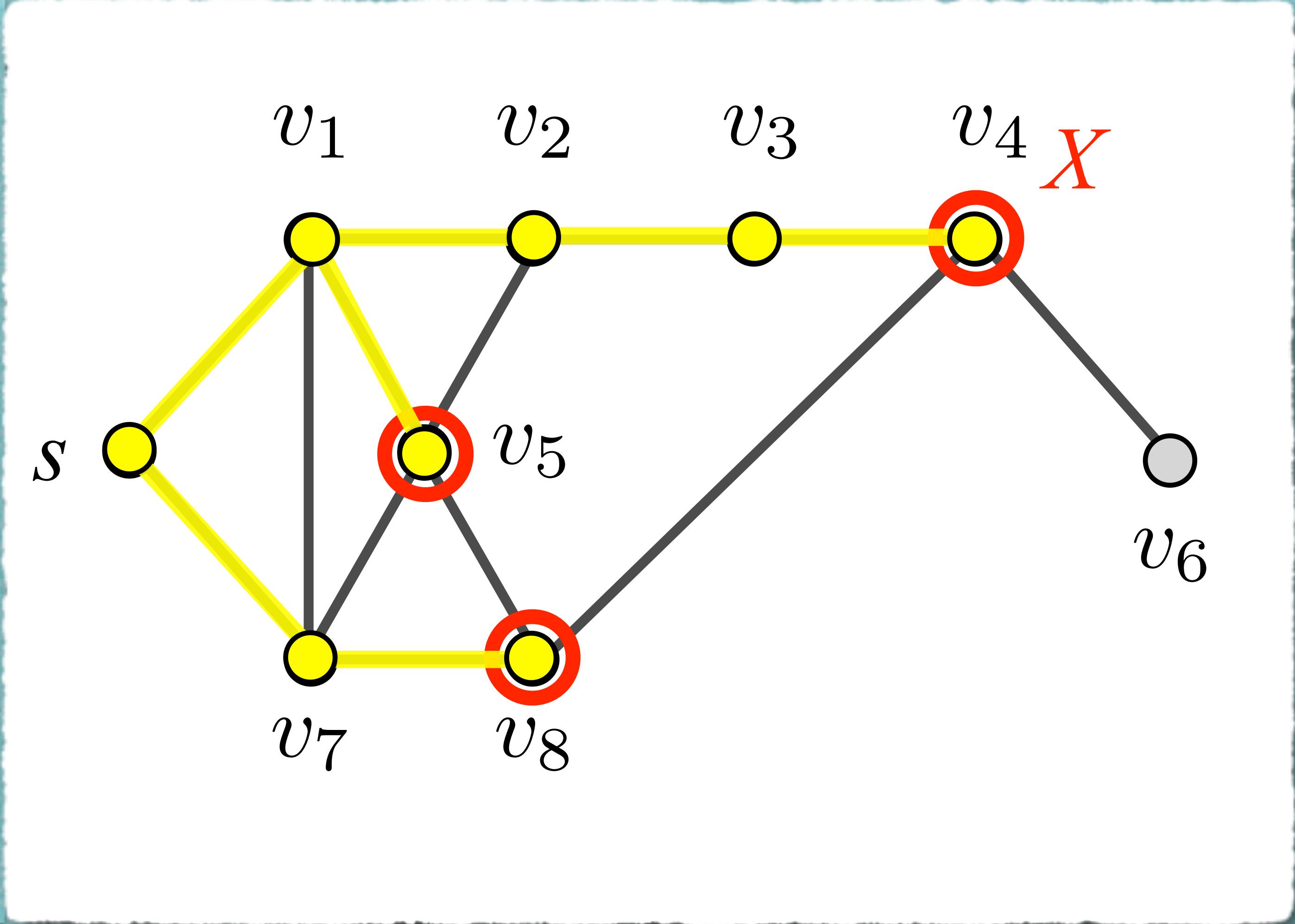
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

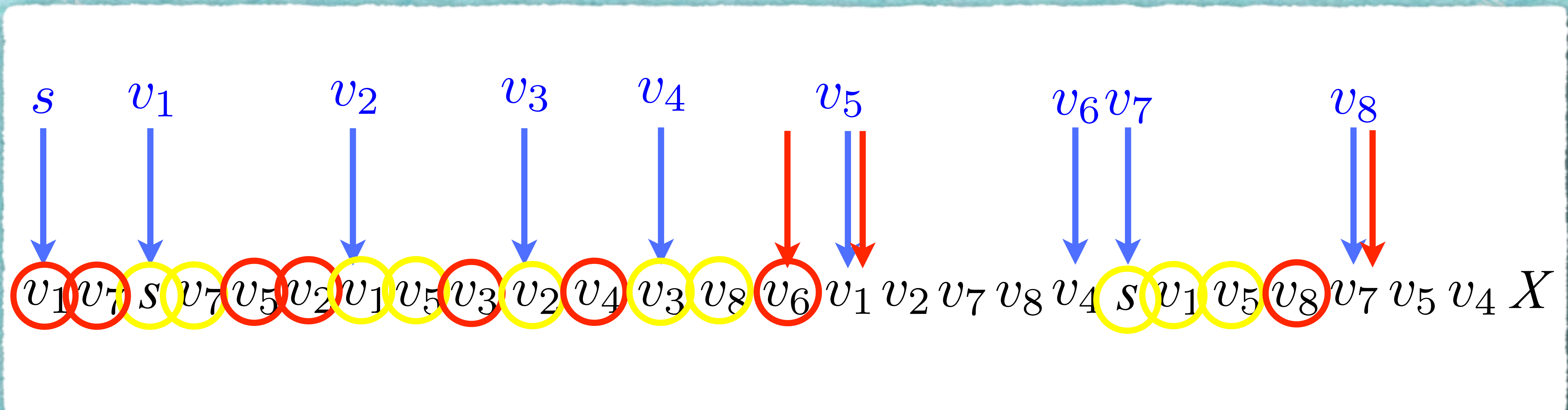
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP

2





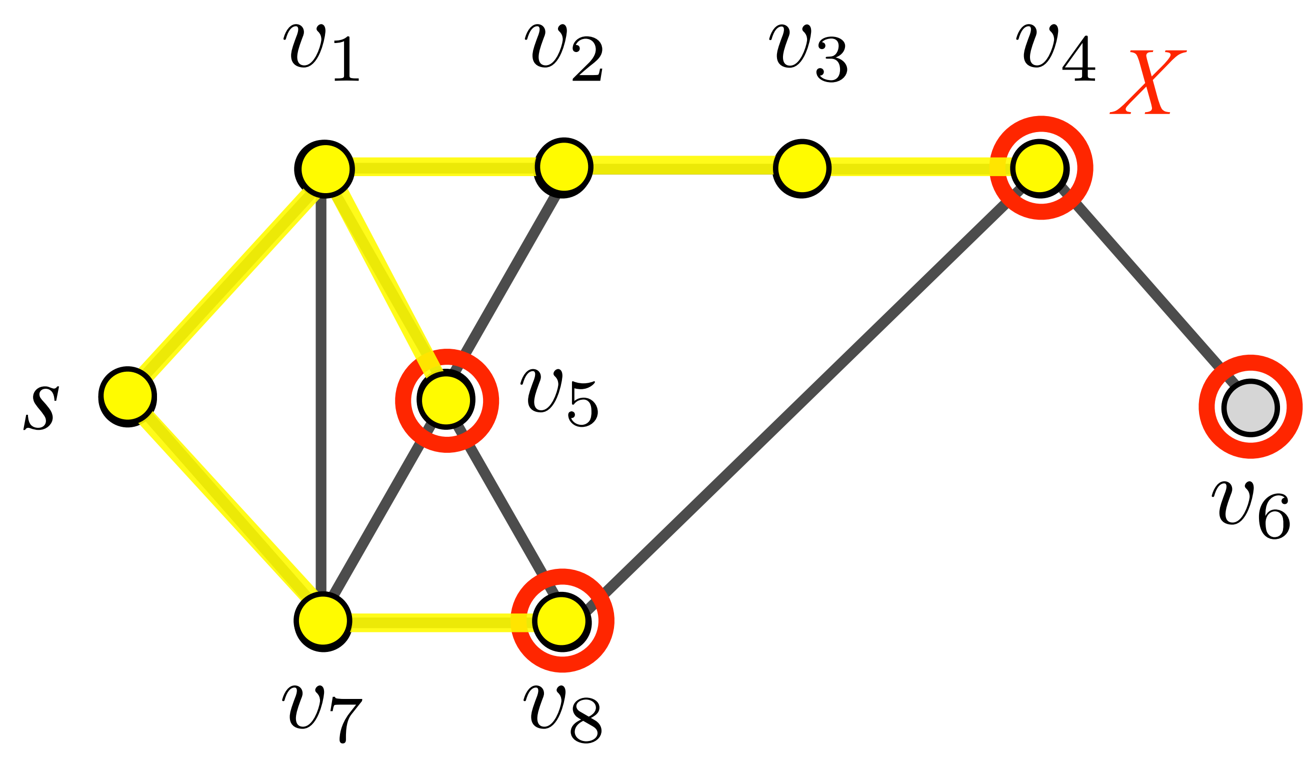
Algorithmus 3.7

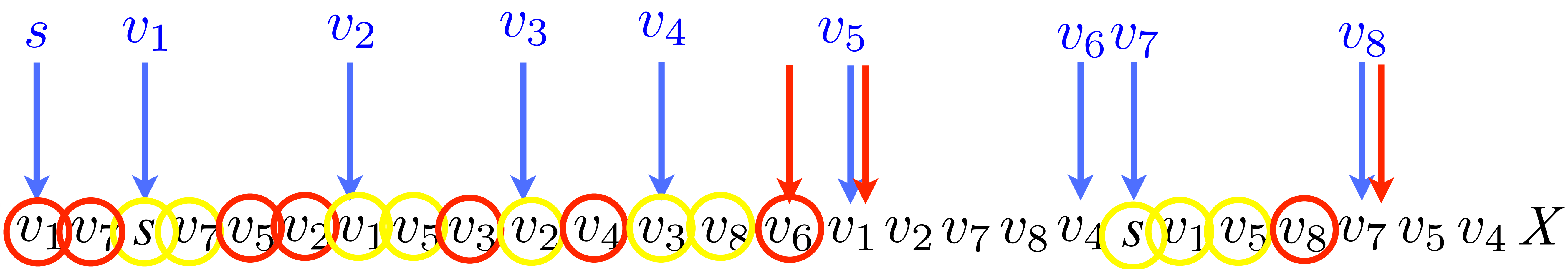
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





Algorithmus 3.7

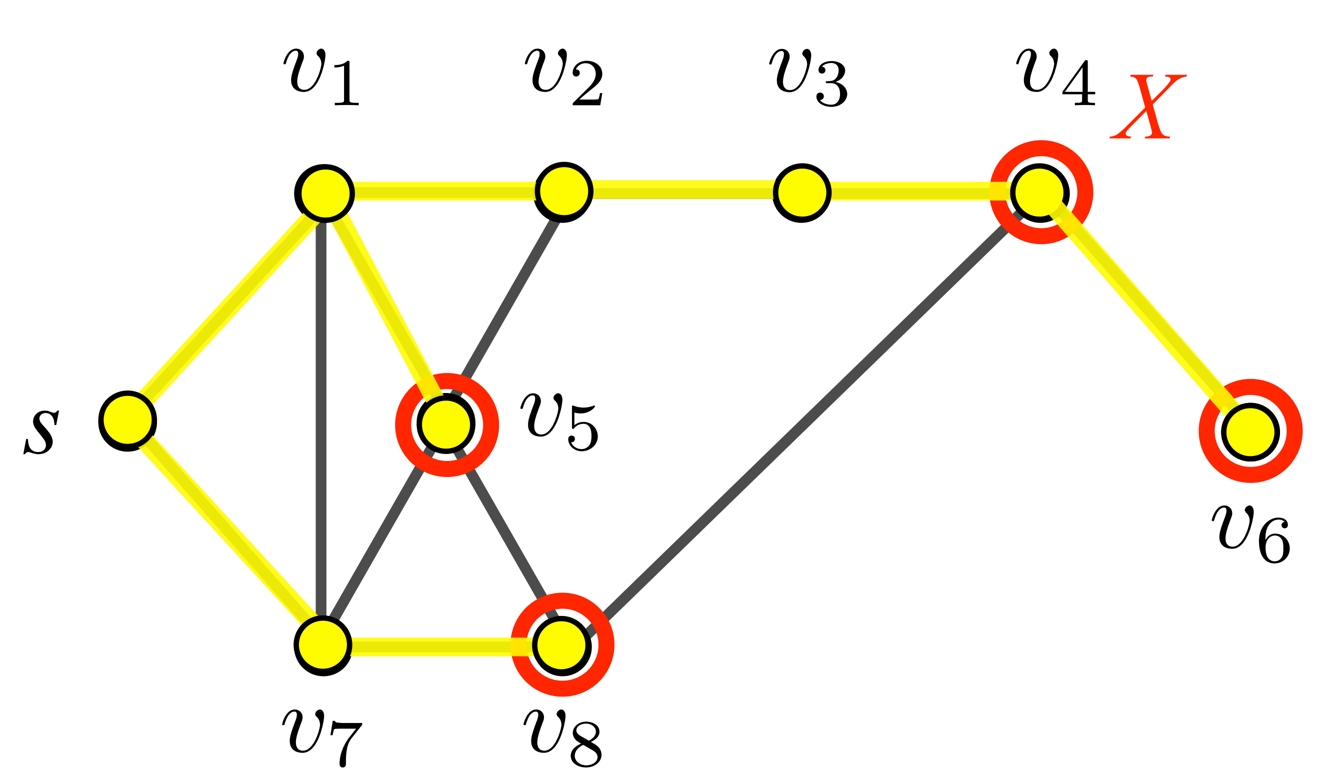
INPUT: Graph $G = (V,E)$, Knoten s

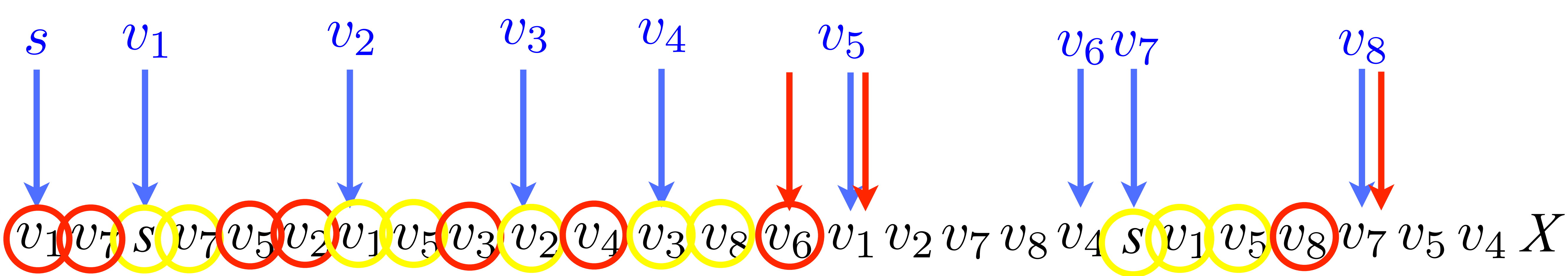
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

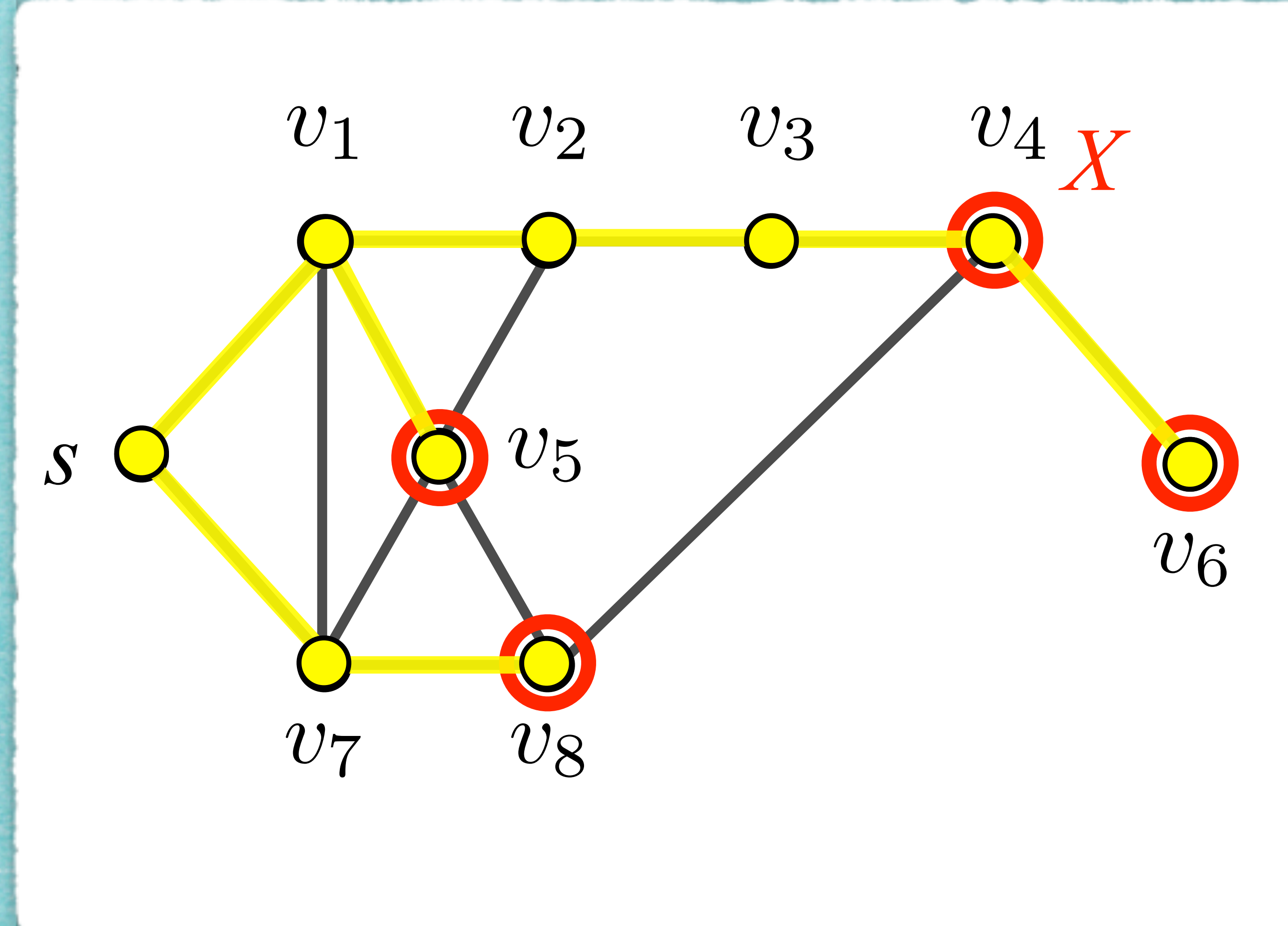
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

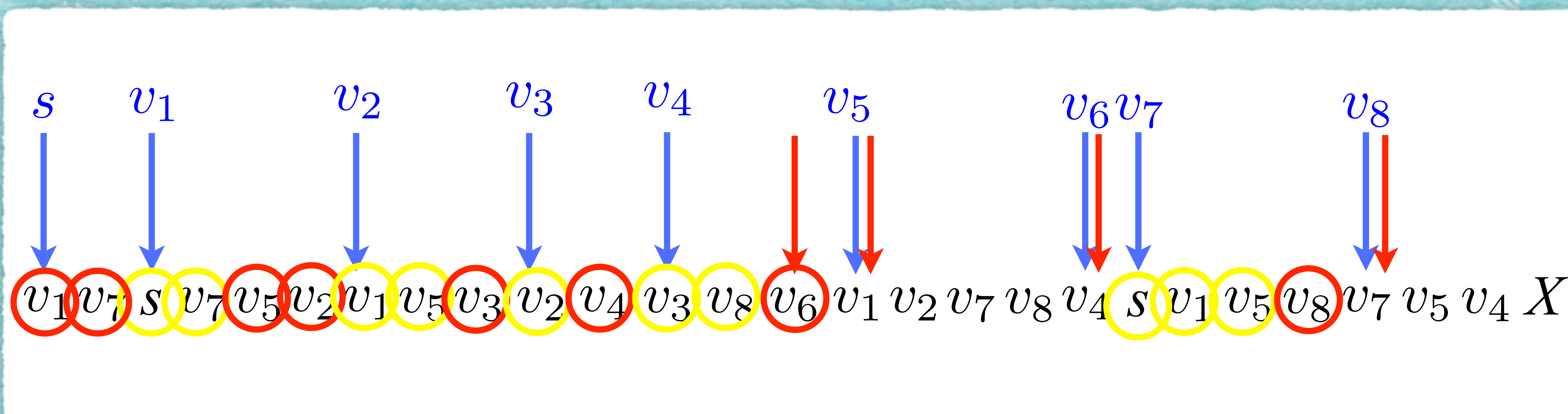
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {

    2.1. Wähle $v \in R$

    2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN

        2.2.1. $R:=R\setminus\{v\}$

    2.3. ELSE {

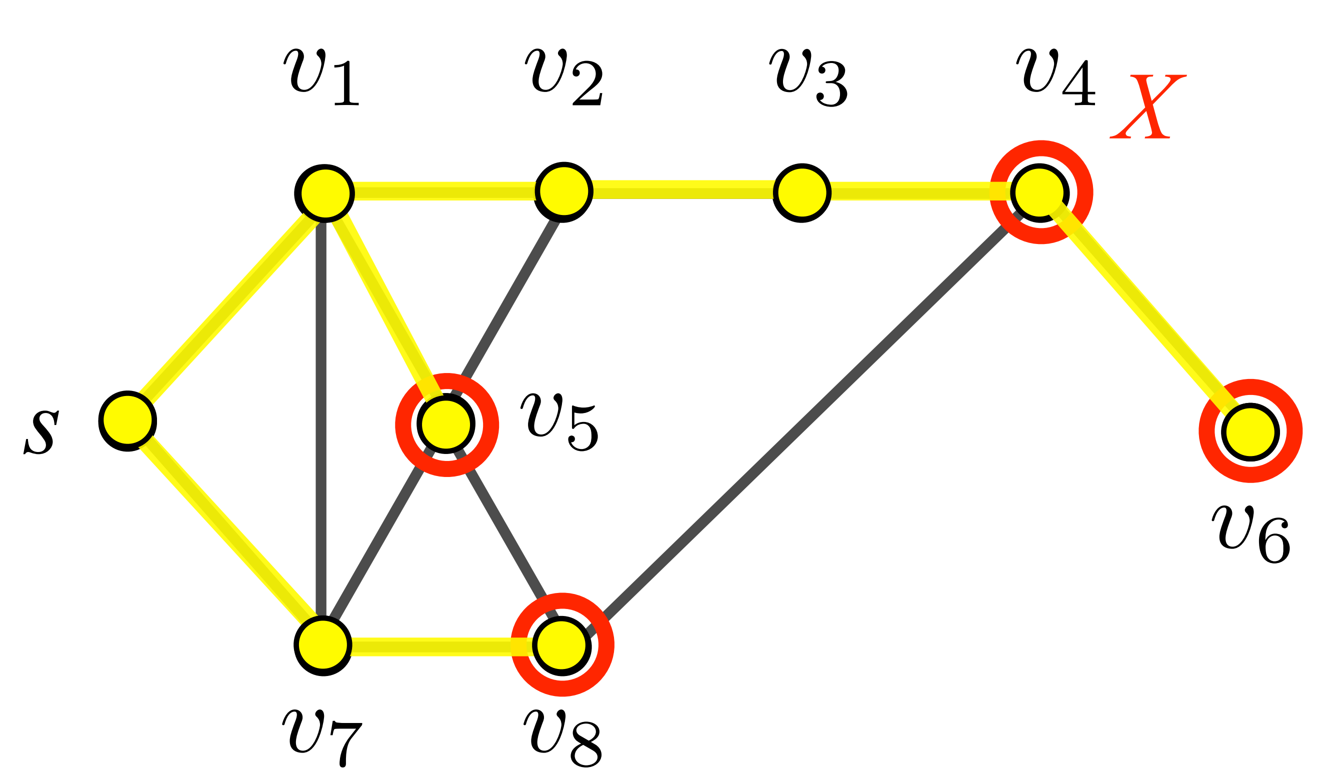
        2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$

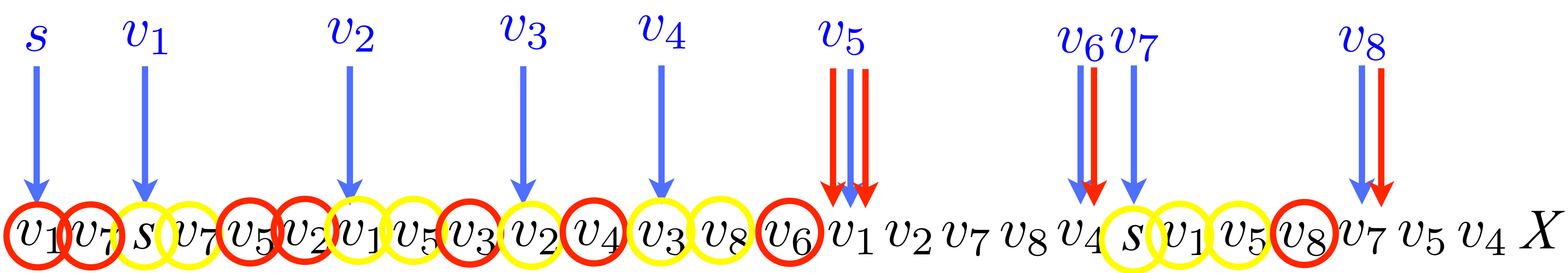
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }

3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R\neq\emptyset)$ DO {

    2.1. Wähle $v \in R$

    2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

    2.2.1. $R:=R\setminus\{v\}$

    2.3. ELSE {

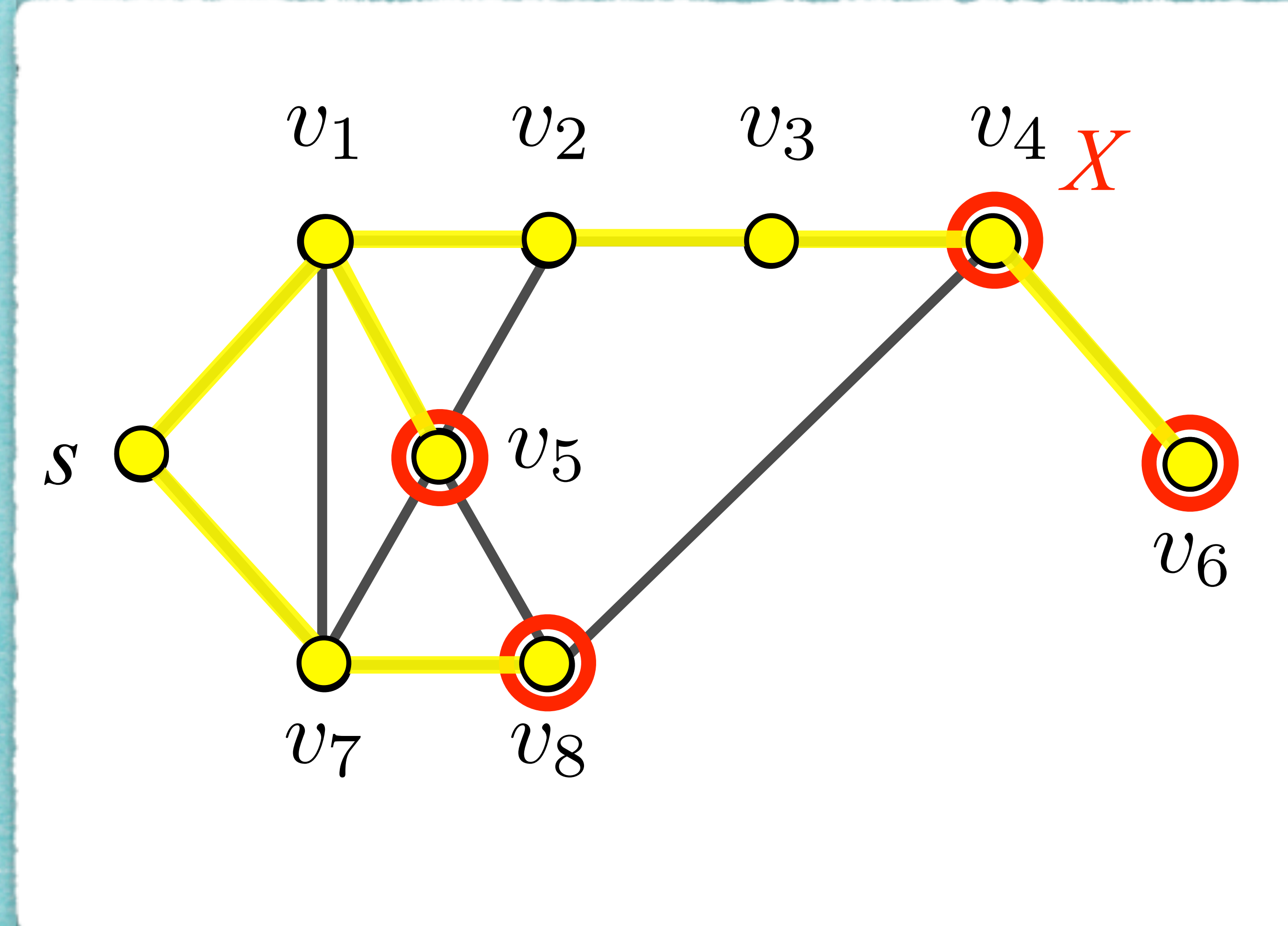
    2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$

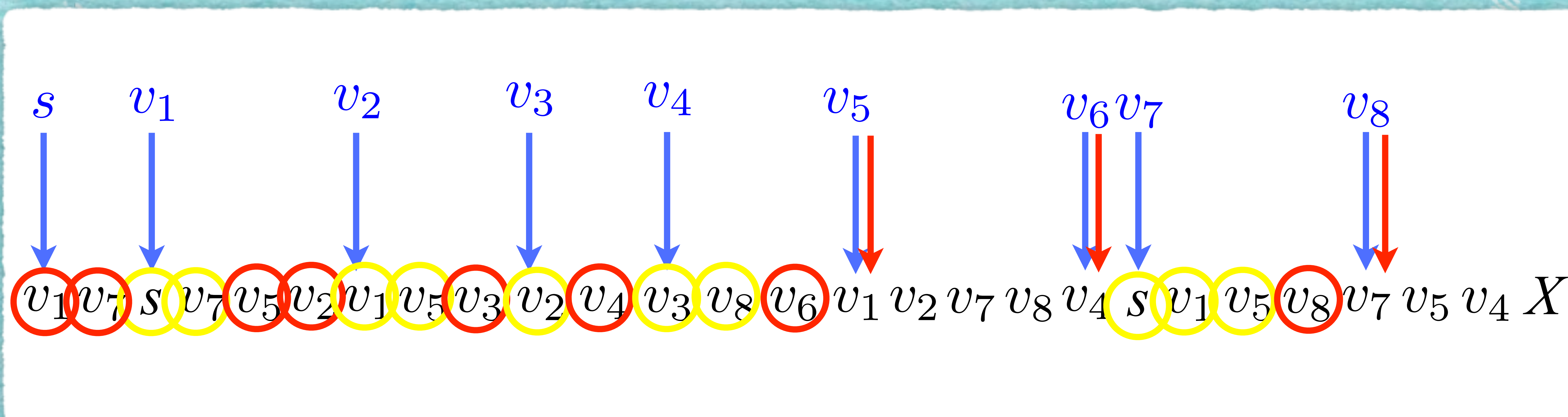
    2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

    }

   }

3. STOP





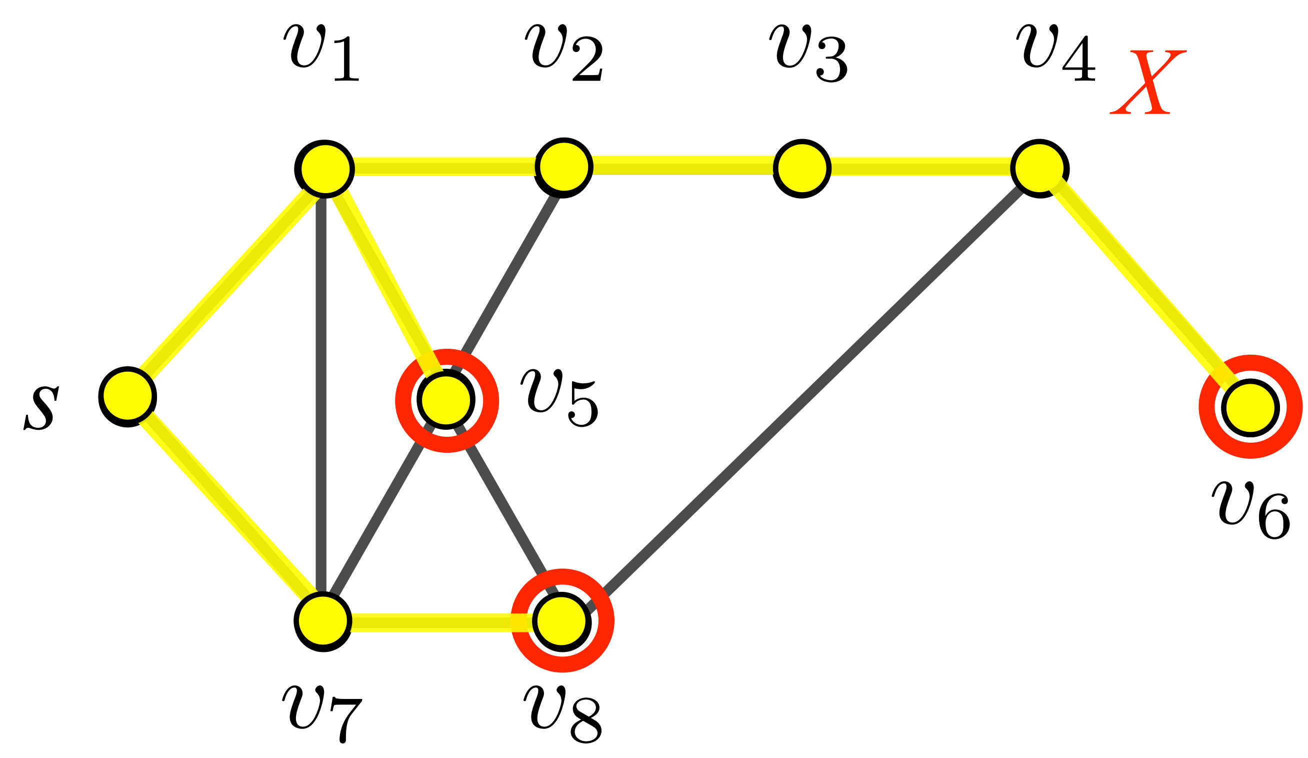
Algorithmus 3.7

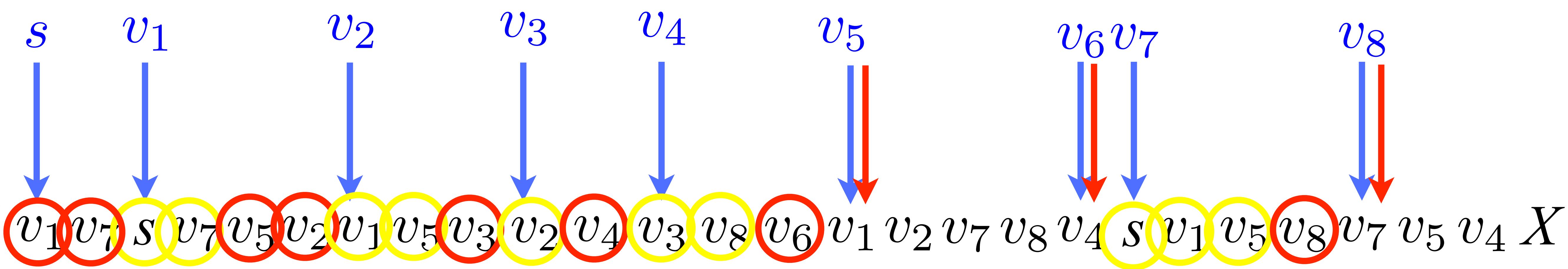
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
           2.2.1. R:=R\{v}
      2.3. ELSE {
           2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
           2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
           }
    }
3.  STOP
```

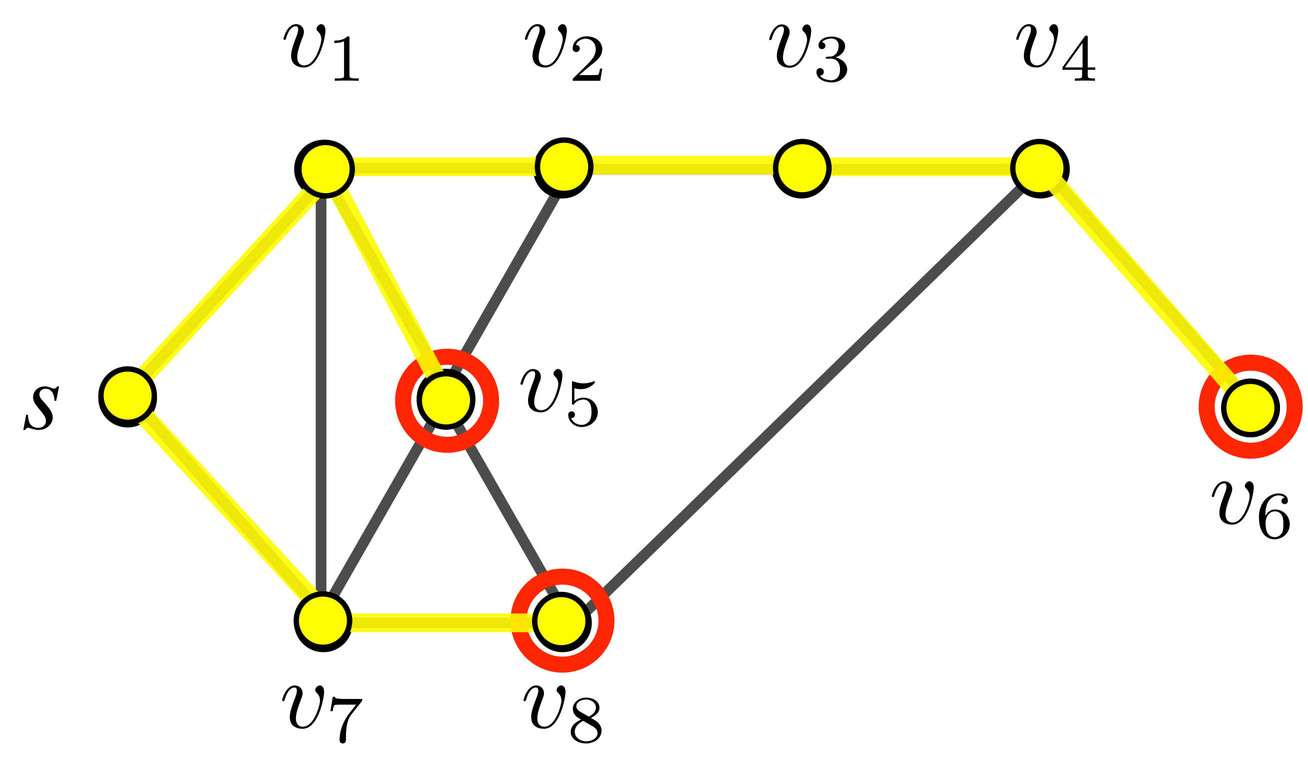



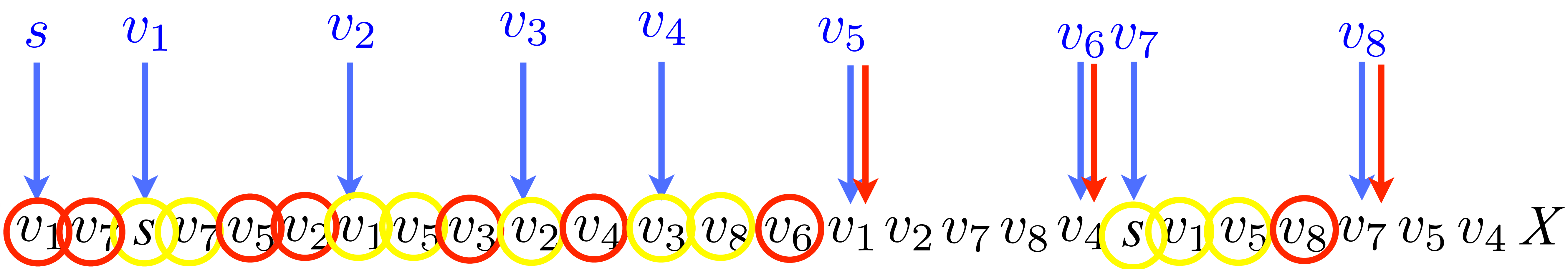


Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   2.1. Wähle $v \in R$
   2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
      2.2.1. $R:=R \setminus \{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP

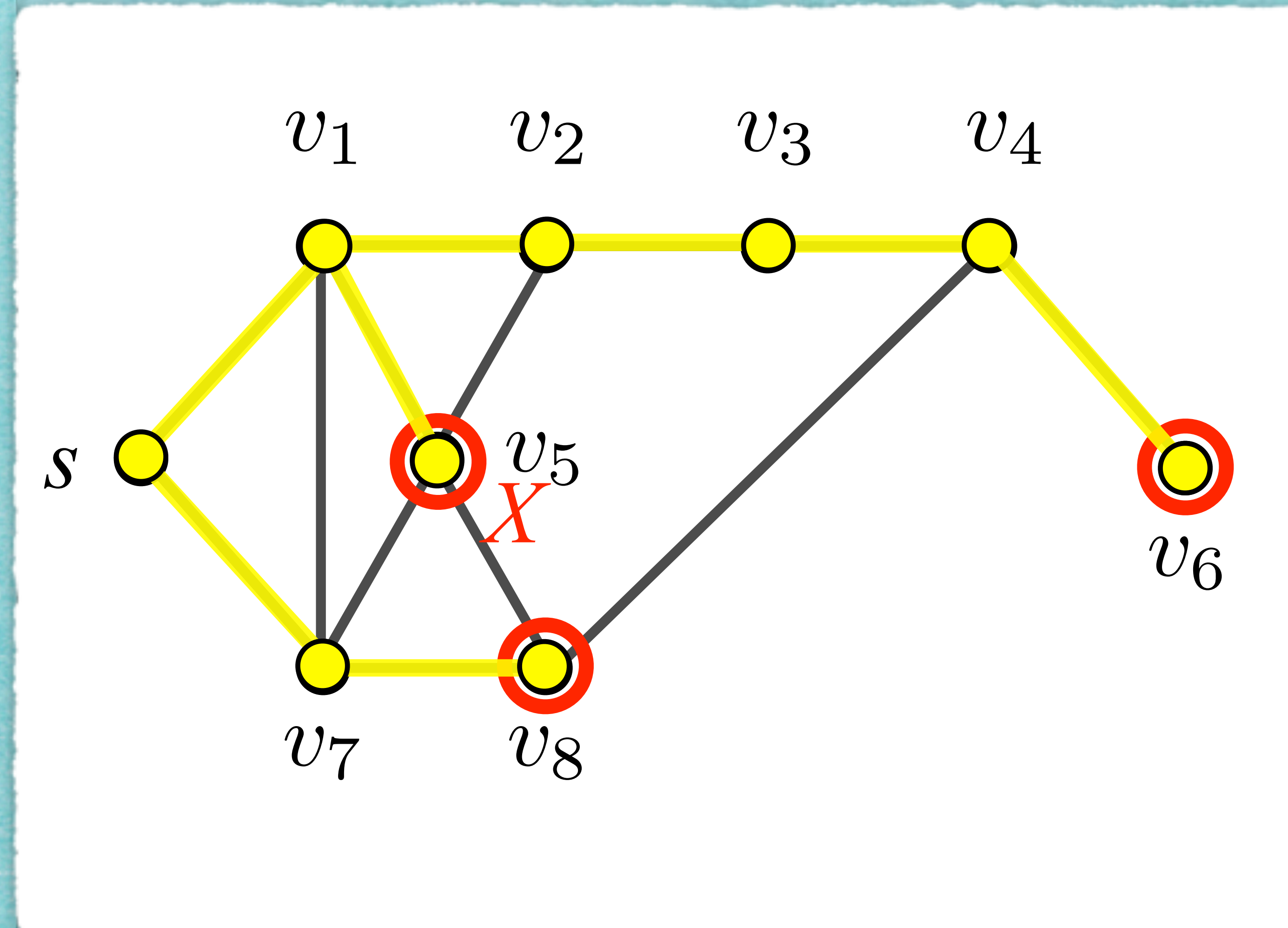


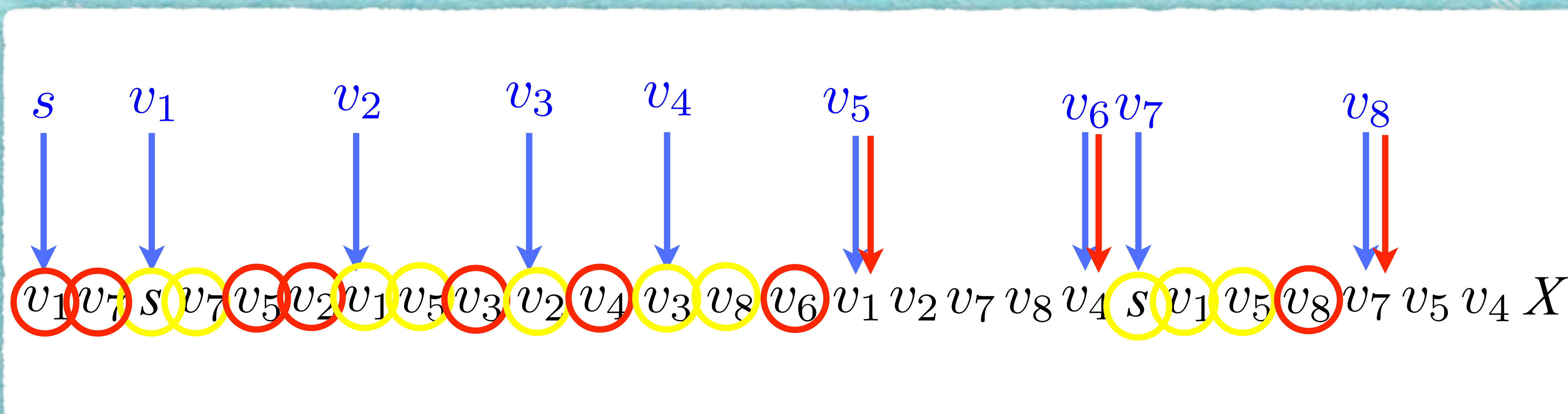


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    2.1. Wähle $v\in R$
    2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP



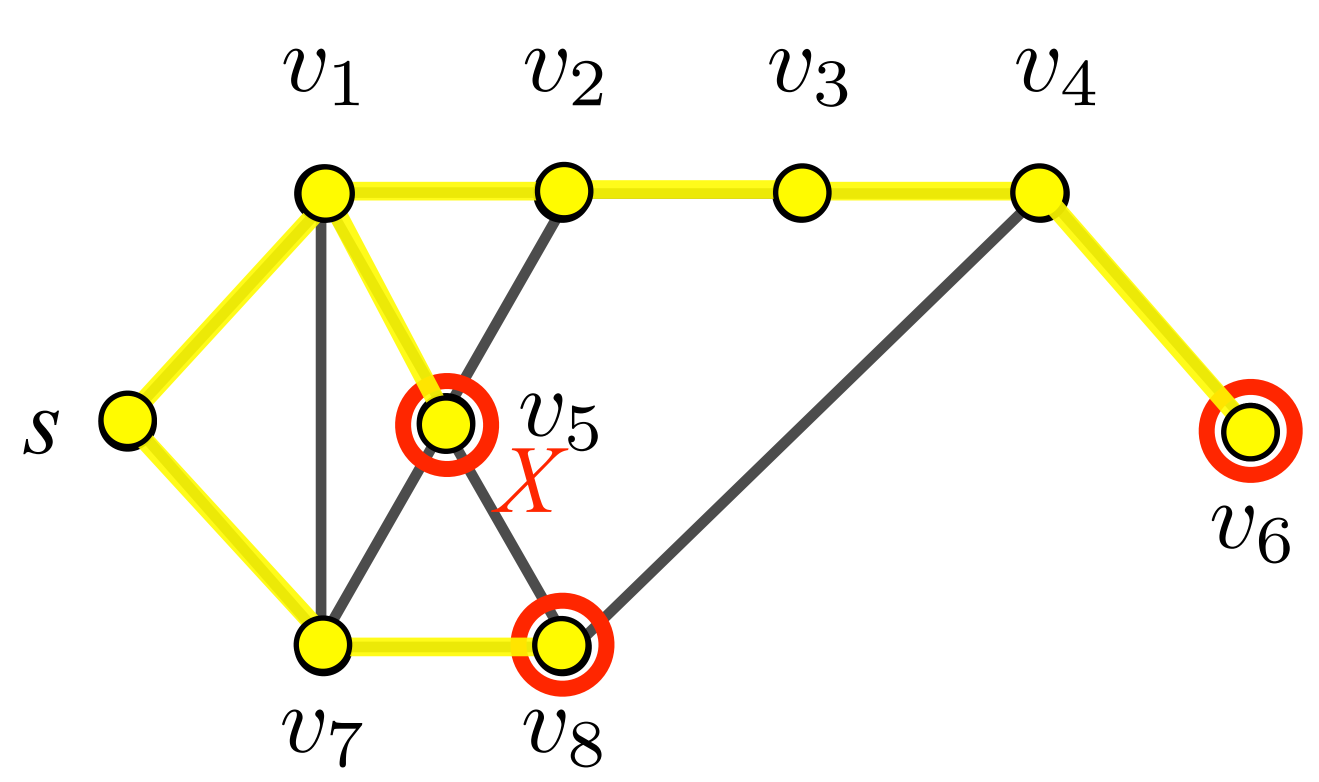


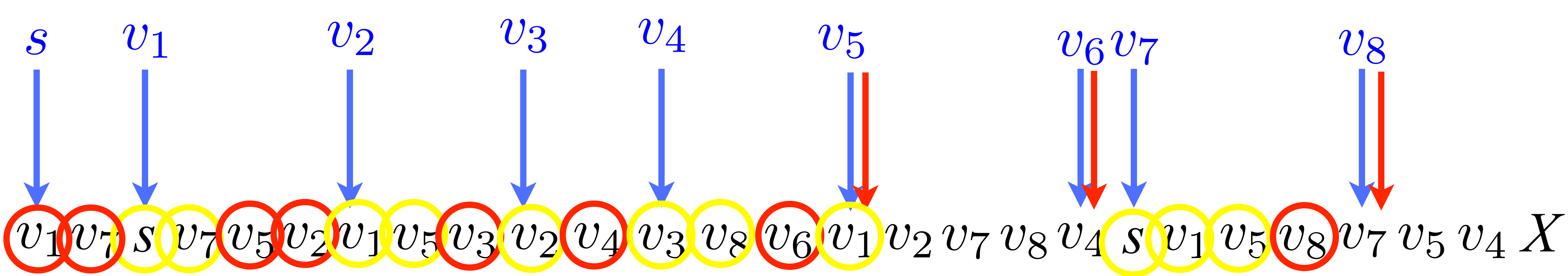
Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        - }
    - }
3. STOP





Algorithmus 3.7

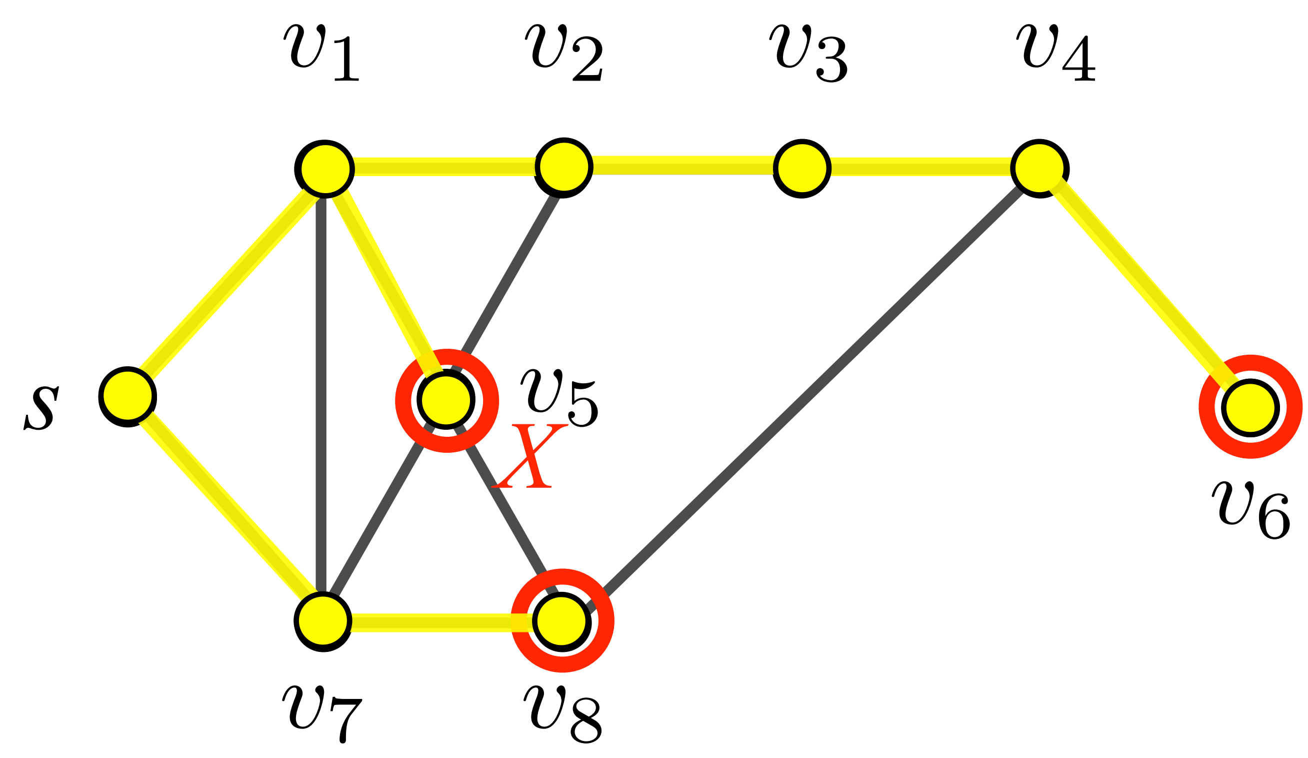
INPUT: Graph G = (V,E), Knoten s

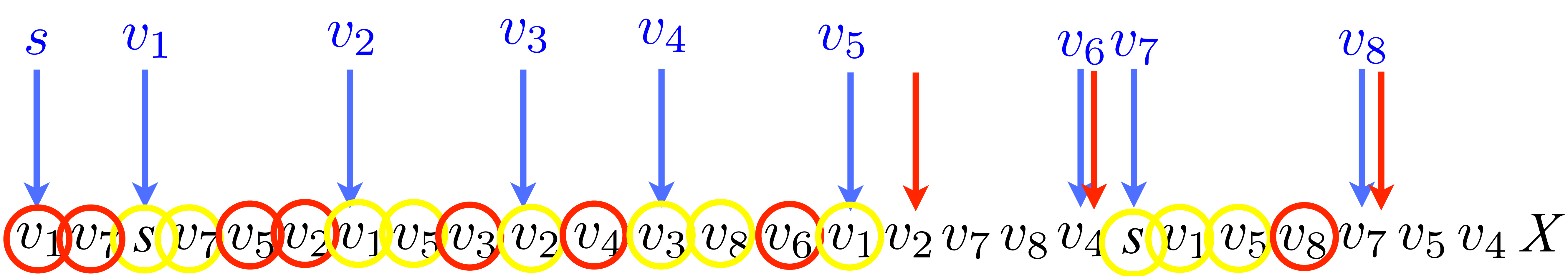
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

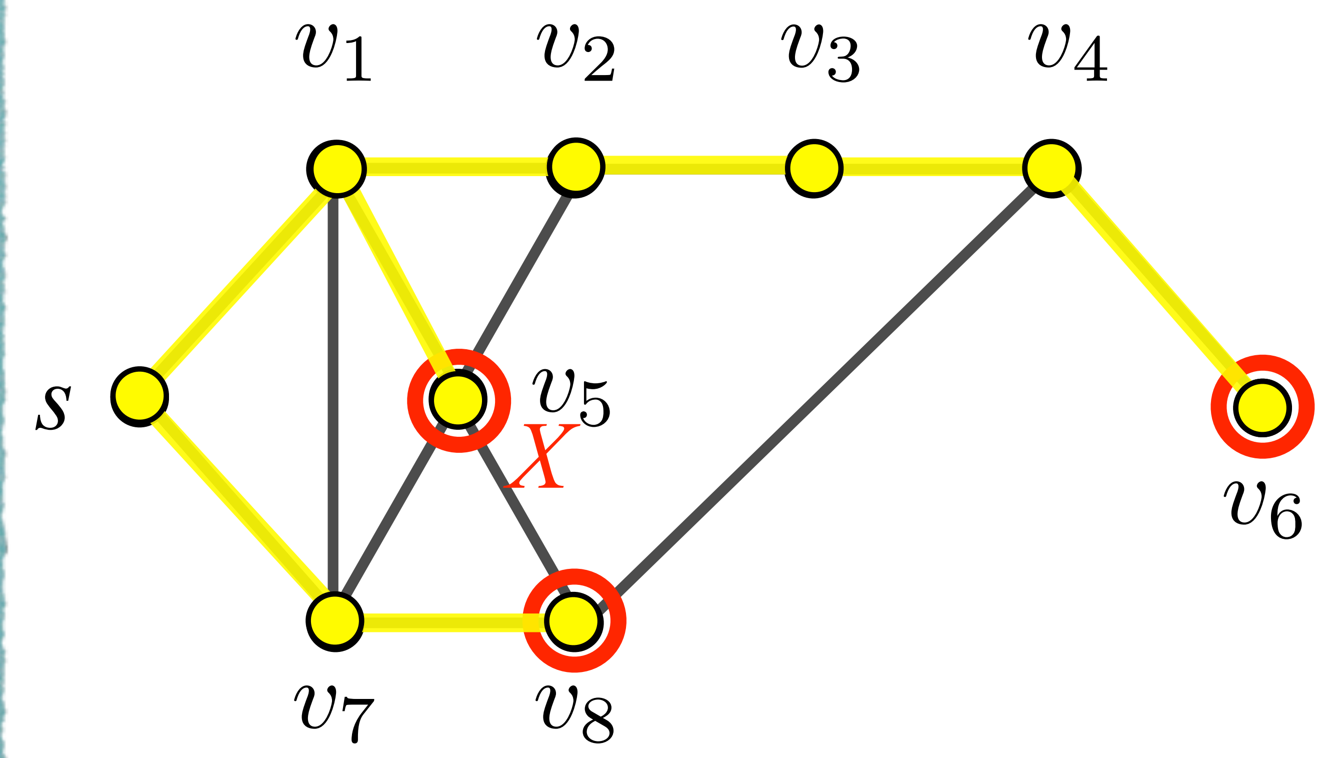
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

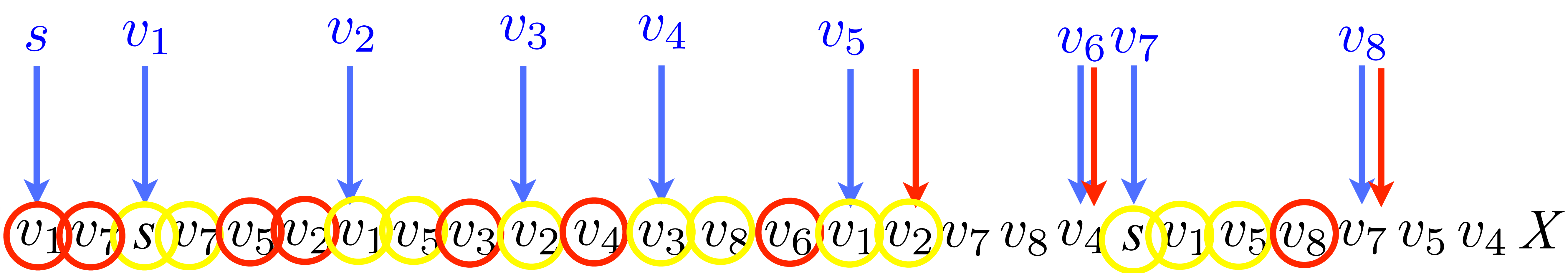
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





Algorithmus 3.7

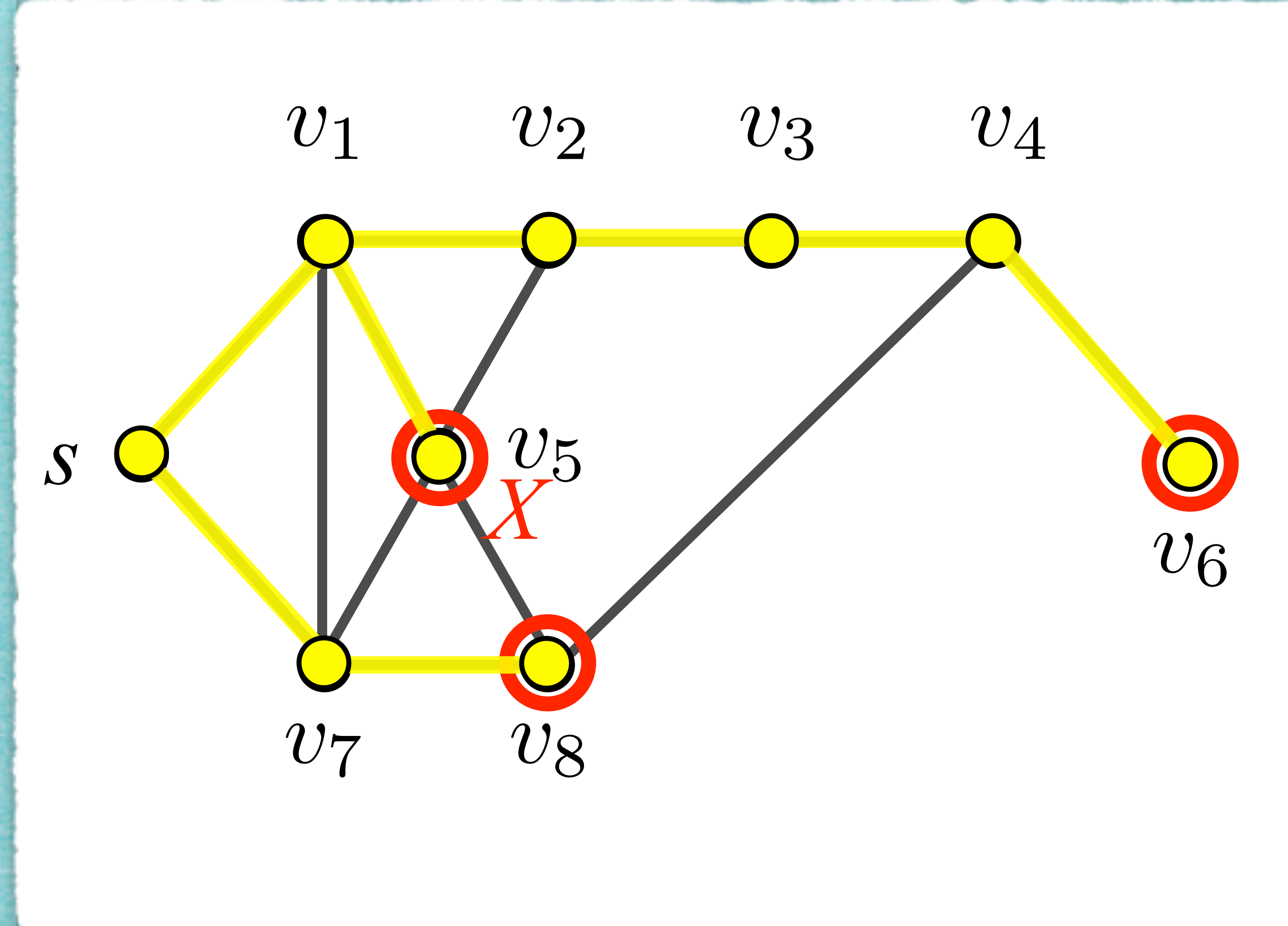
INPUT: Graph G = (V,E), Knoten s

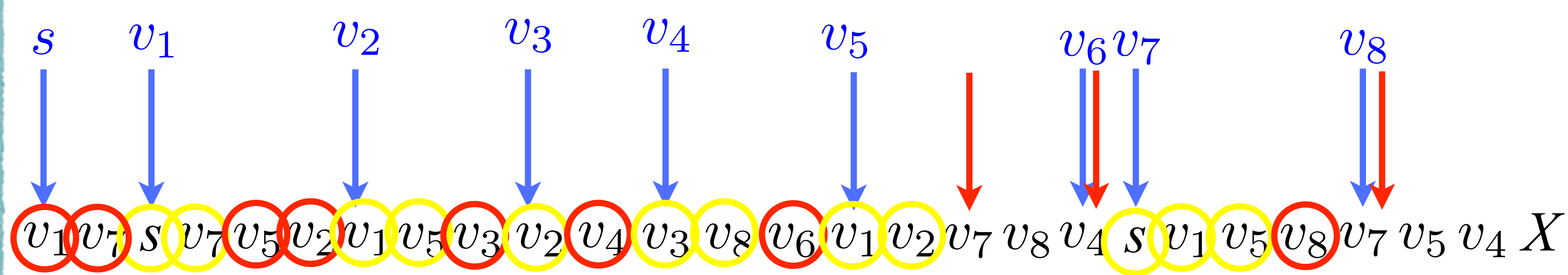
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





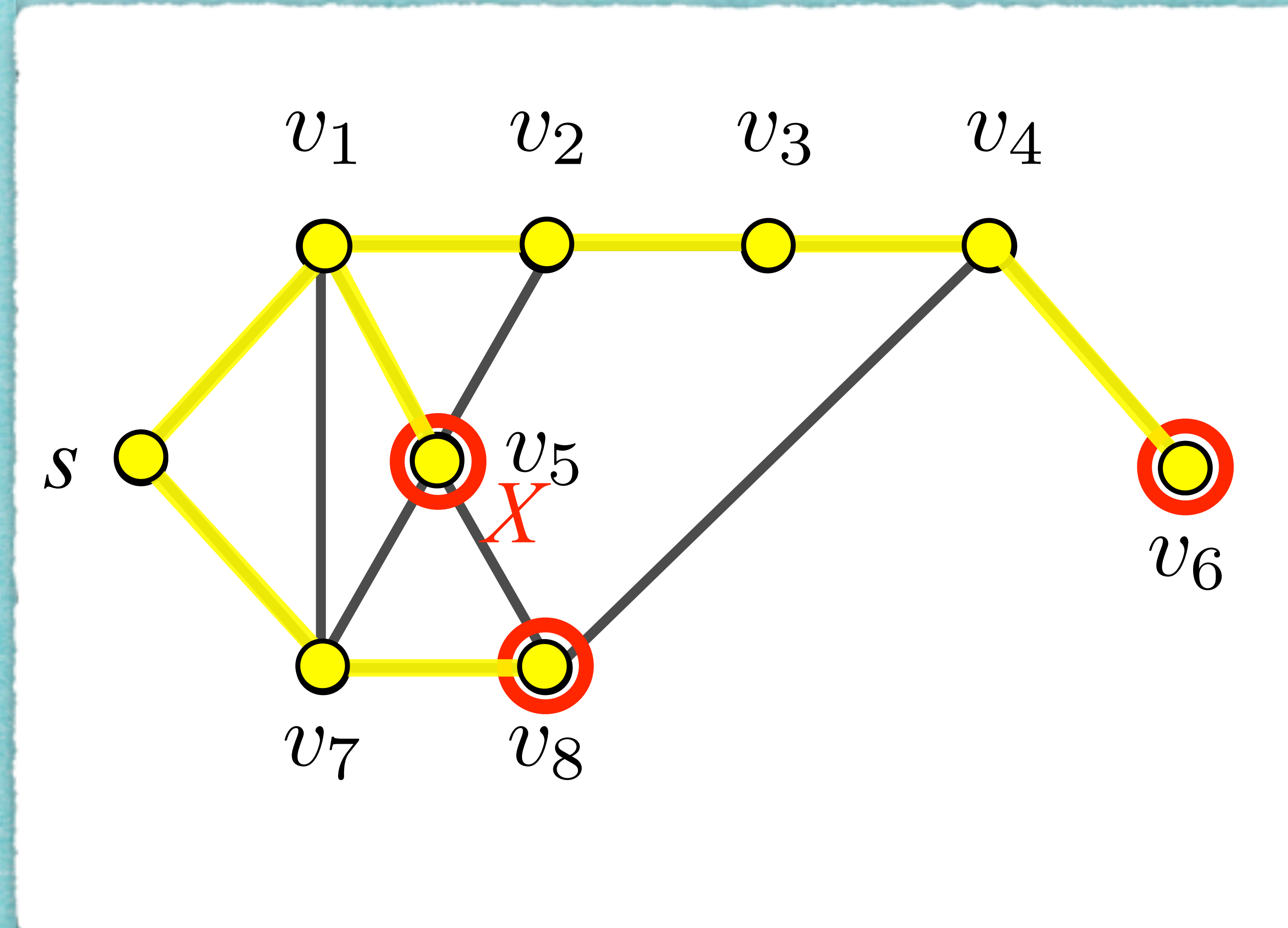
Algorithmus 3.7

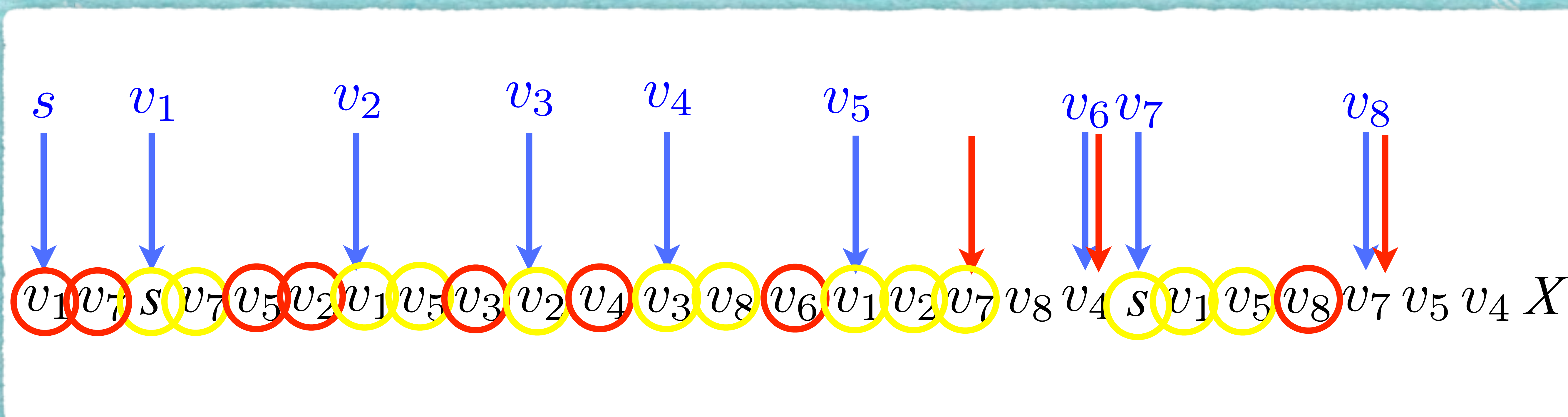
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





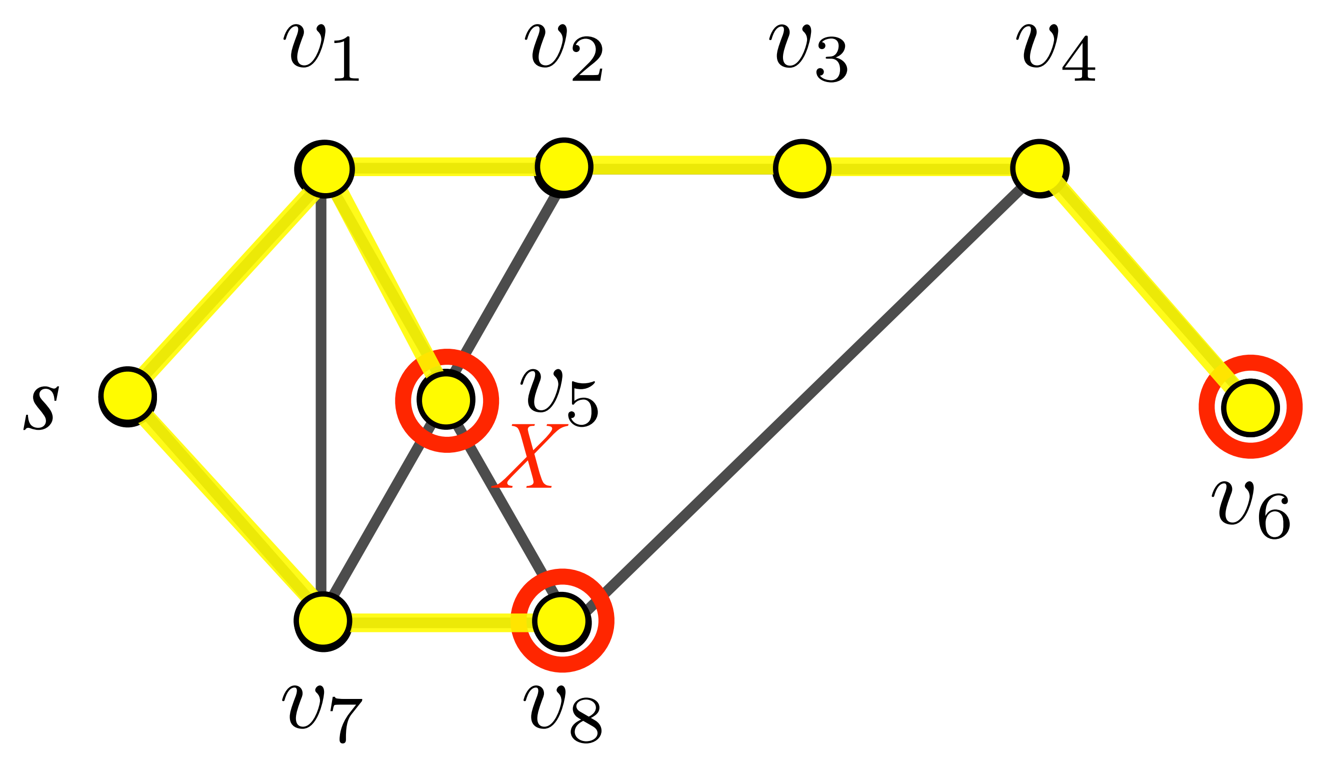
Algorithmus 3.7

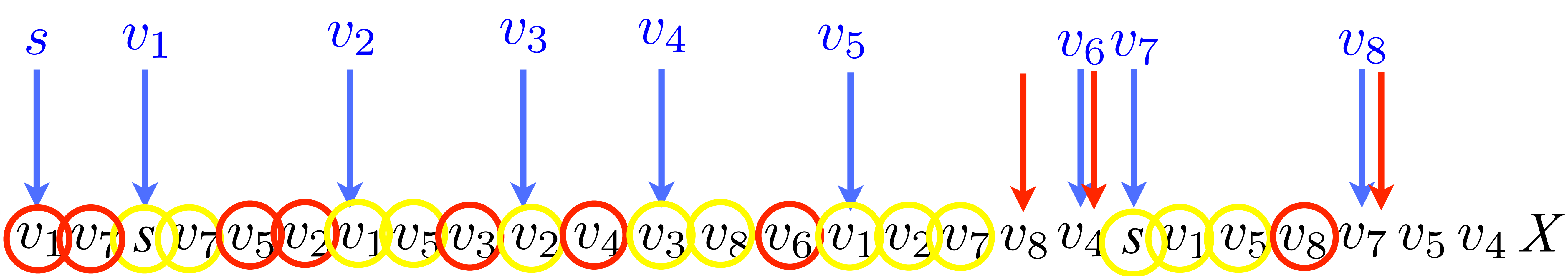
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP



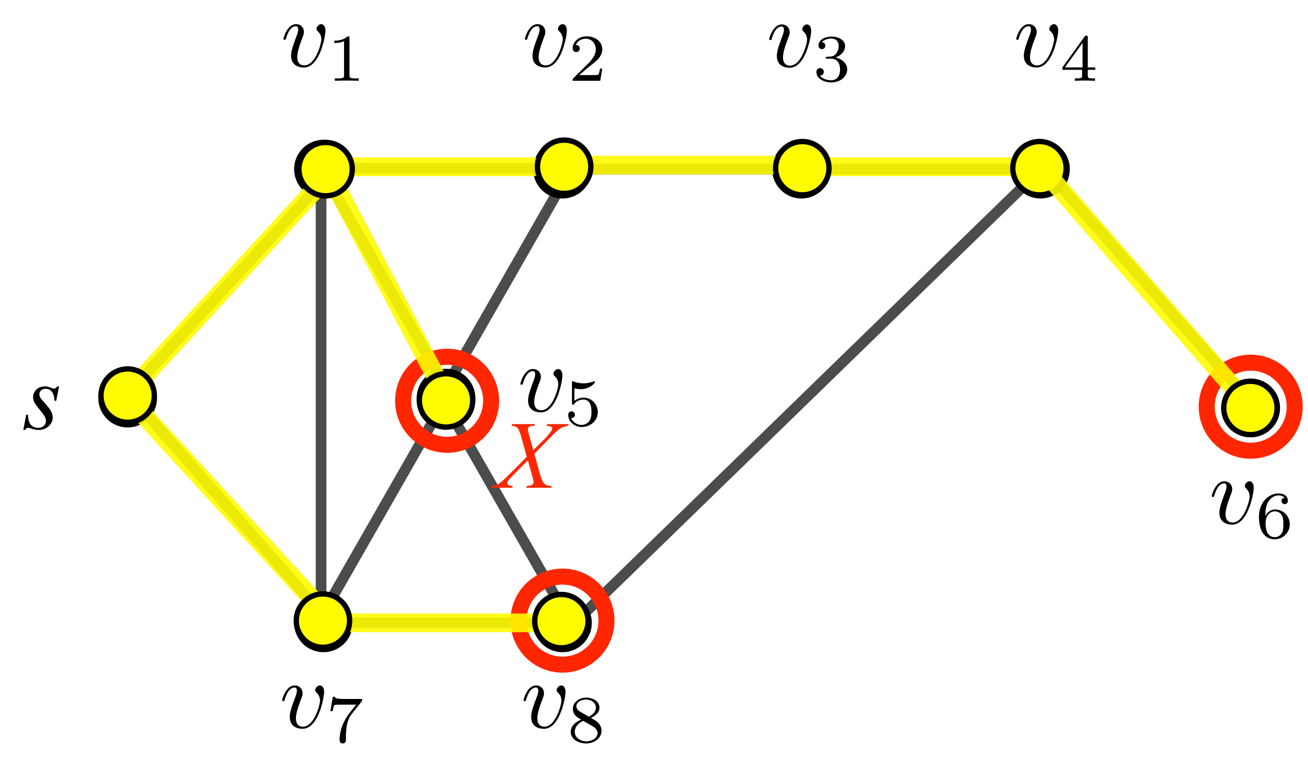


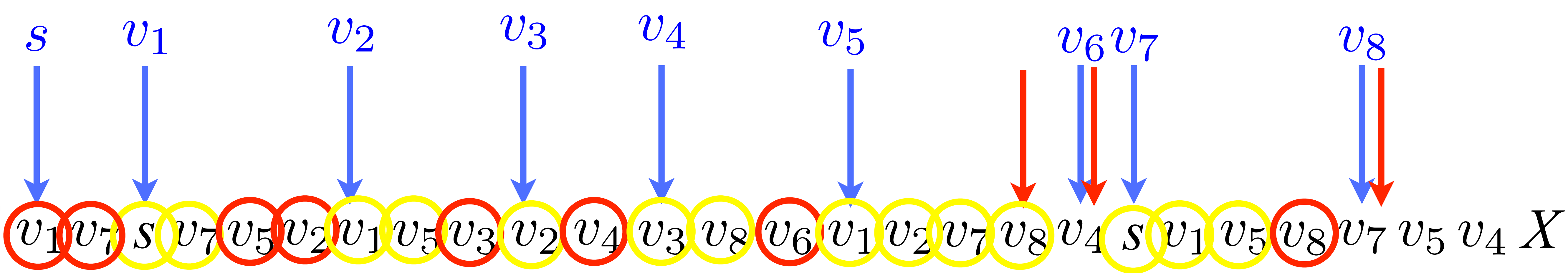
Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   2.1. Wähle $v \in R$
   2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
      2.2.1. $R:=R\setminus\{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP





Algorithmus 3.7

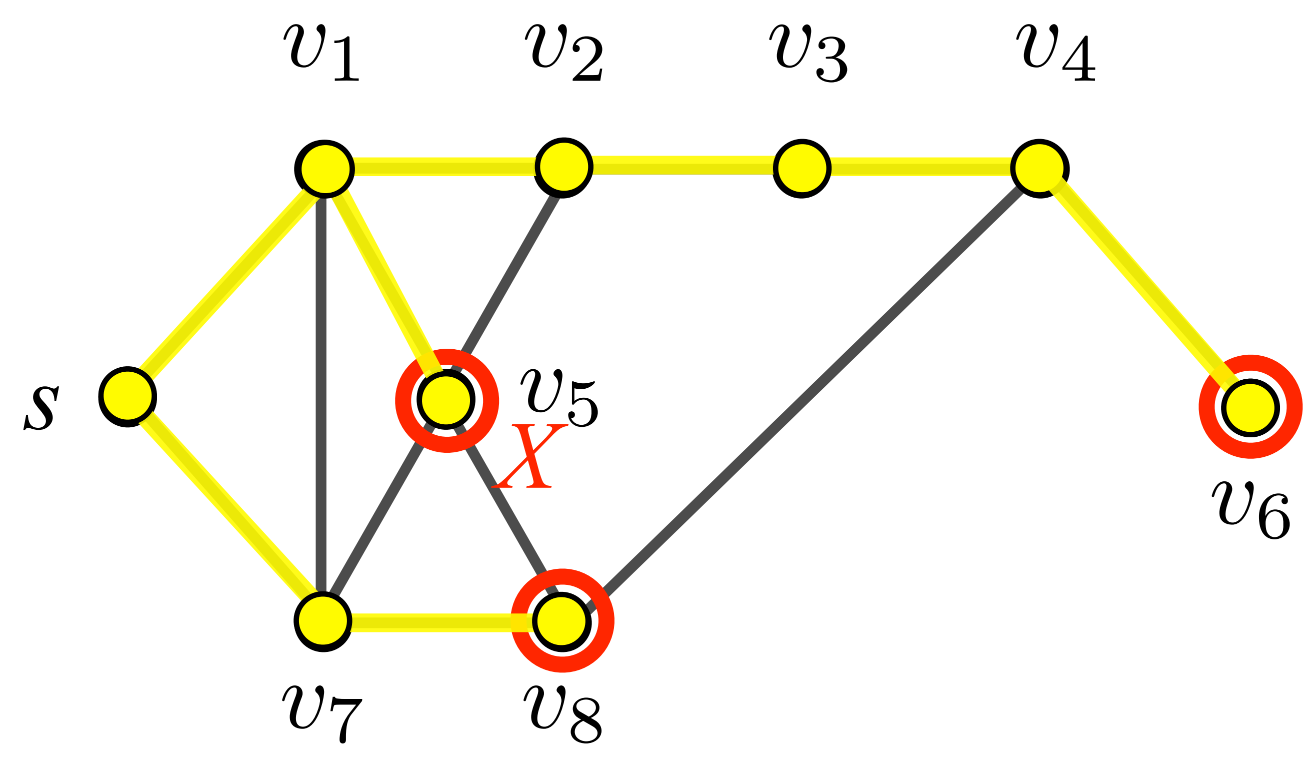
INPUT: Graph G = (V,E), Knoten s

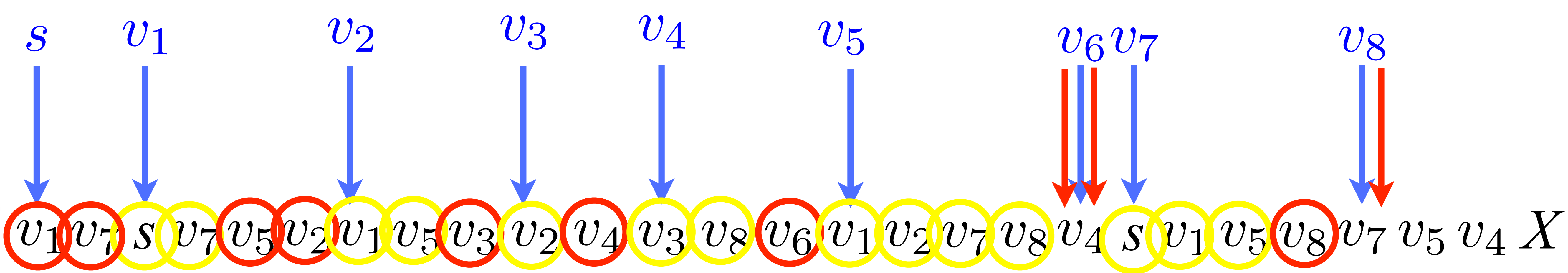
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
     - }
   - }
3. STOP

2





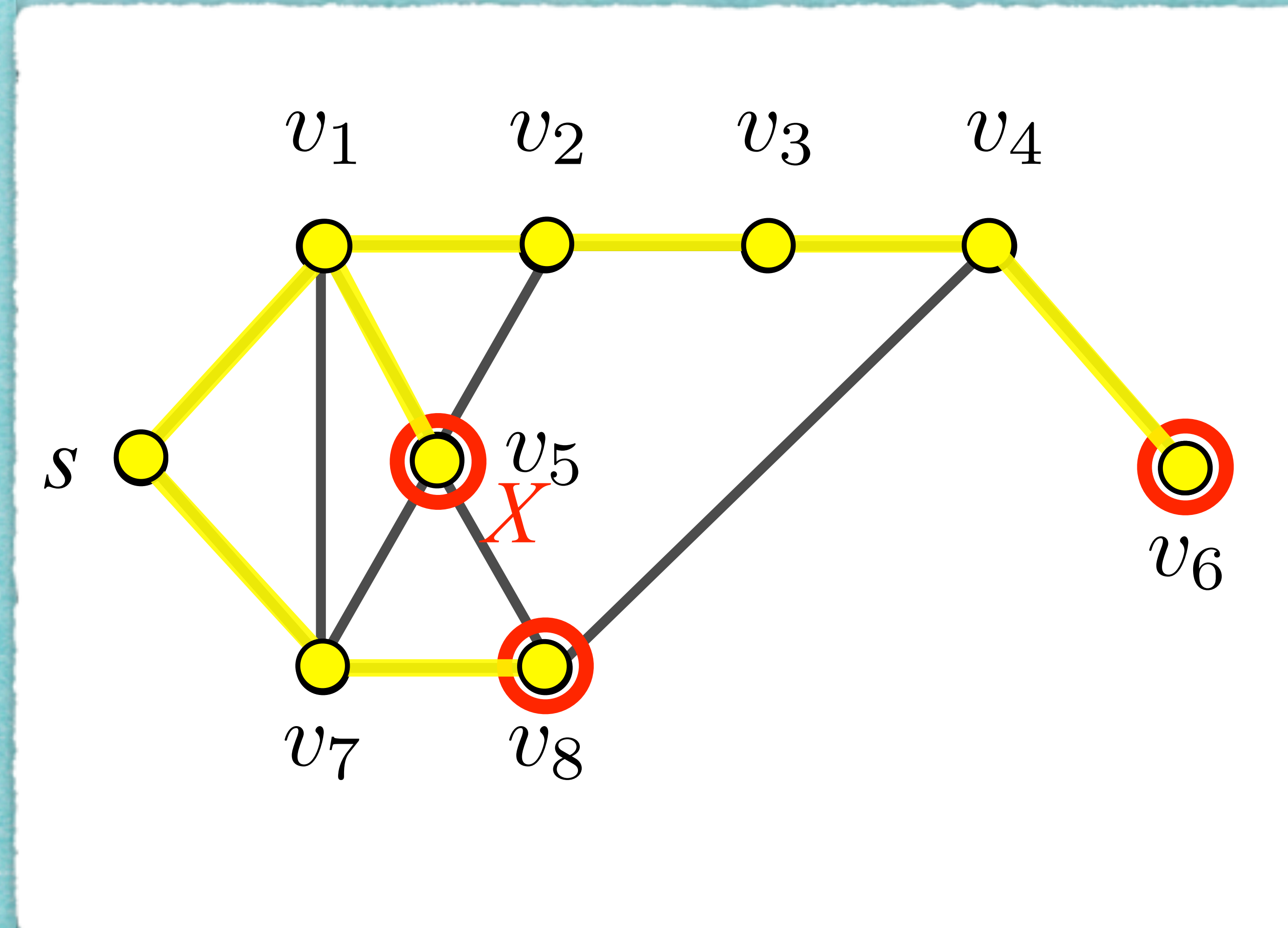
Algorithmus 3.7

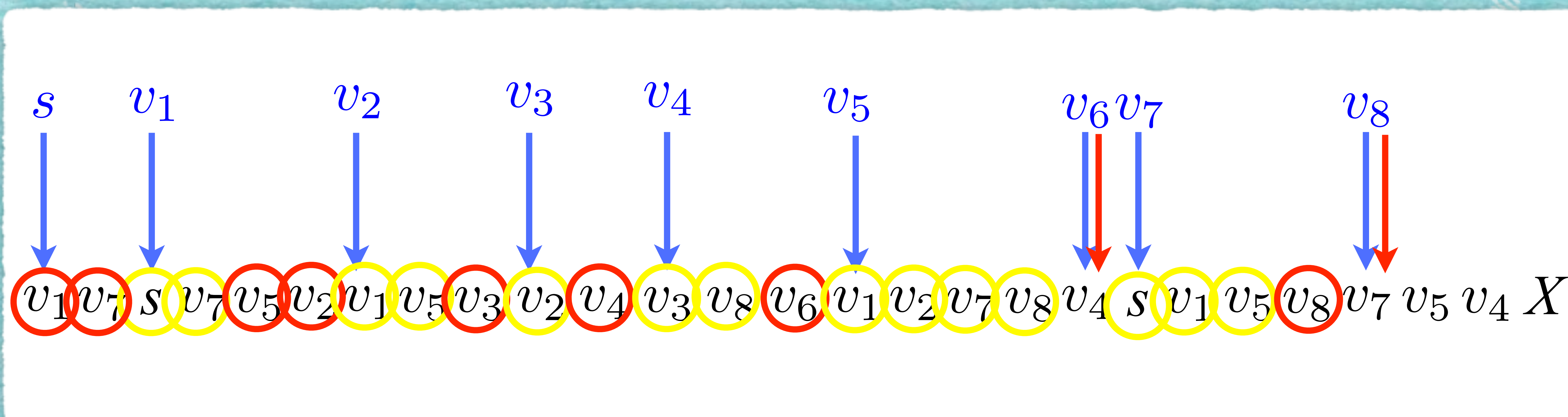
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

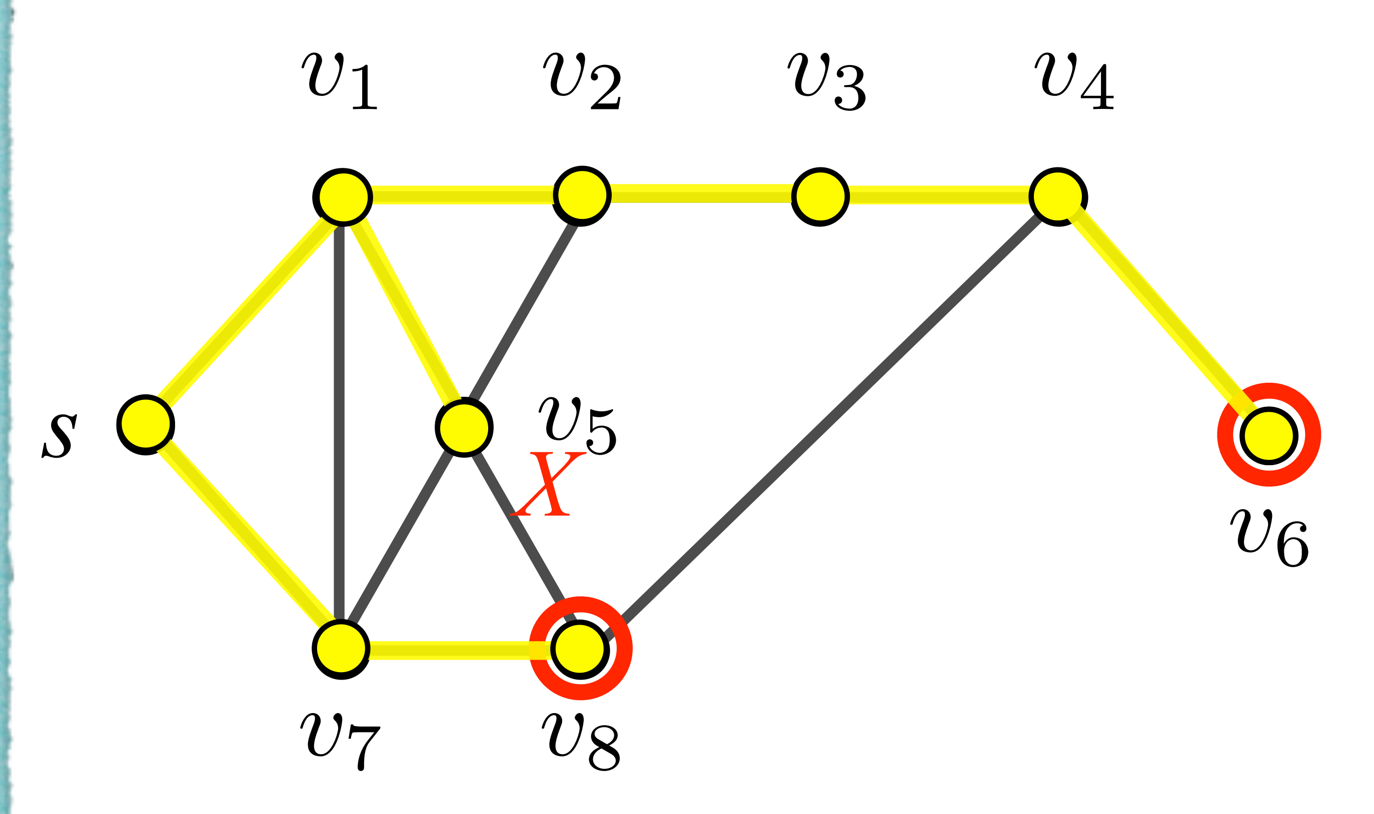


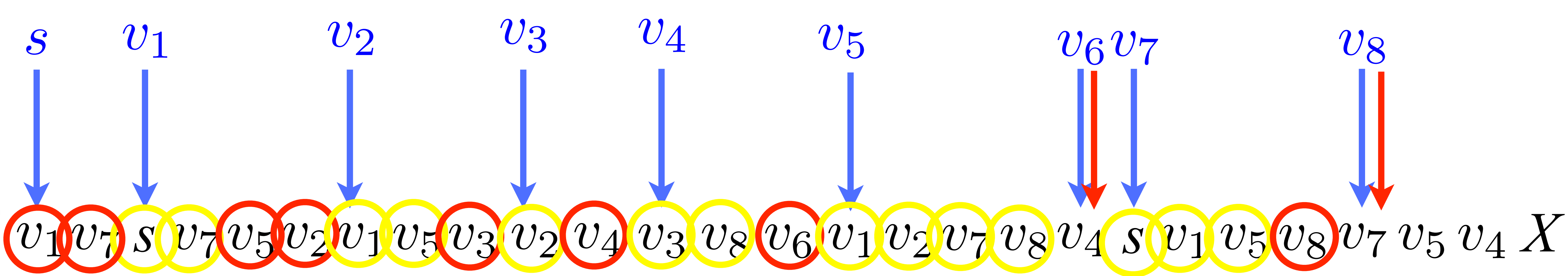


Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1. Wähle v∈R
      2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
          2.2.1. R:=R\{v}
      2.3. ELSE {
          2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
          2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
          }
    }
3.  STOP
```



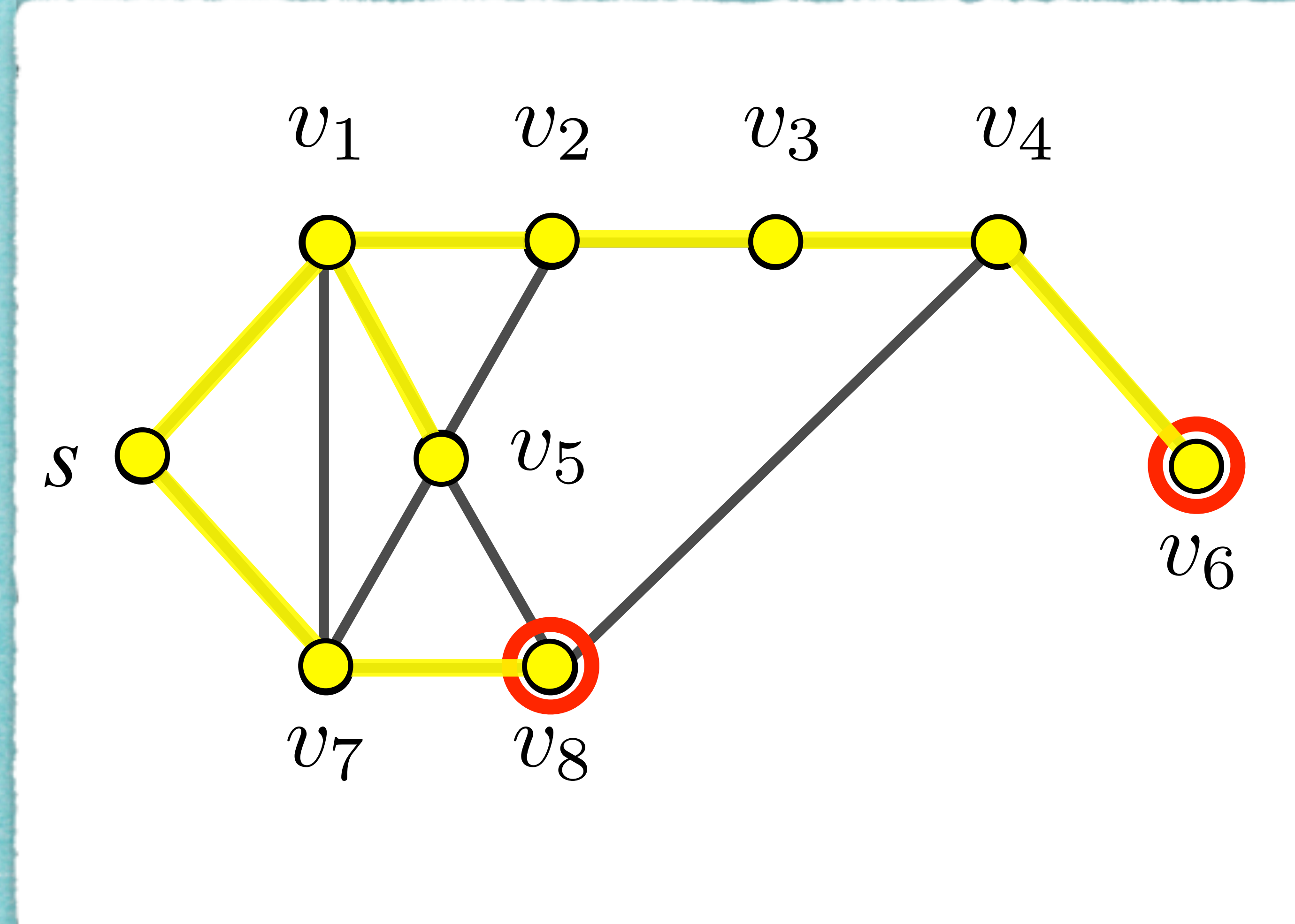
Algorithmus 3.7

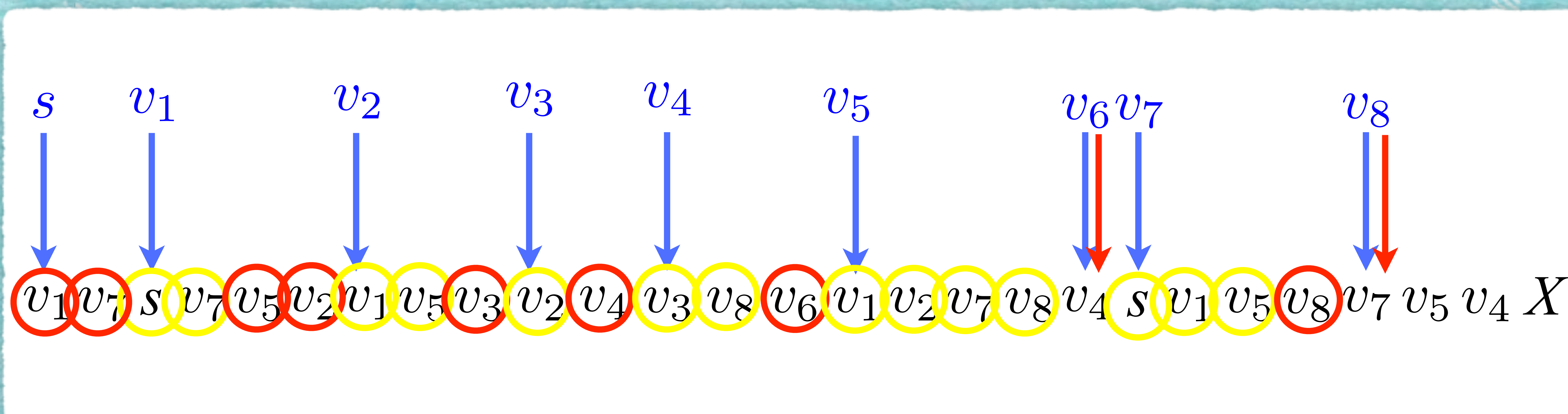
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





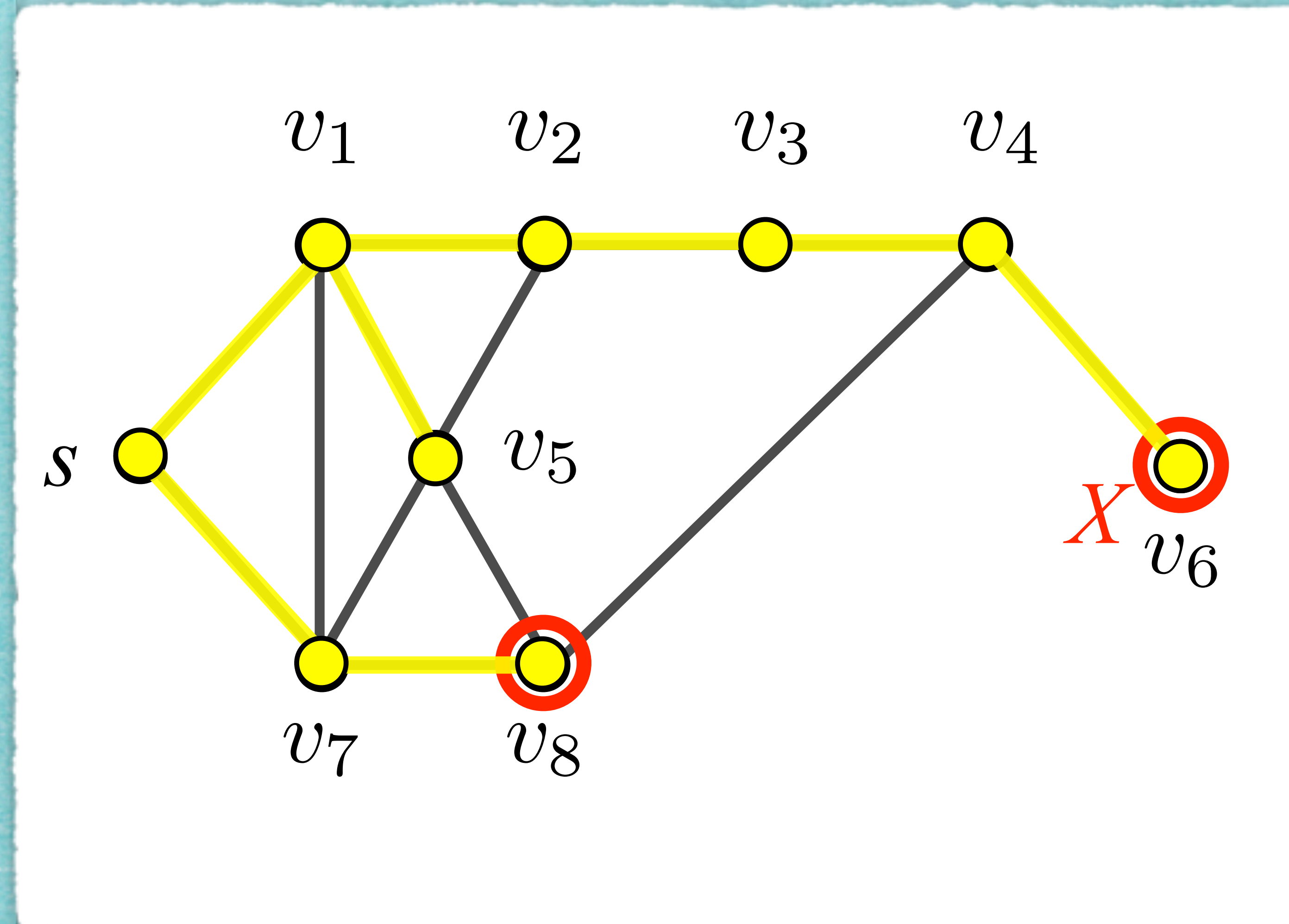
Algorithmus 3.7

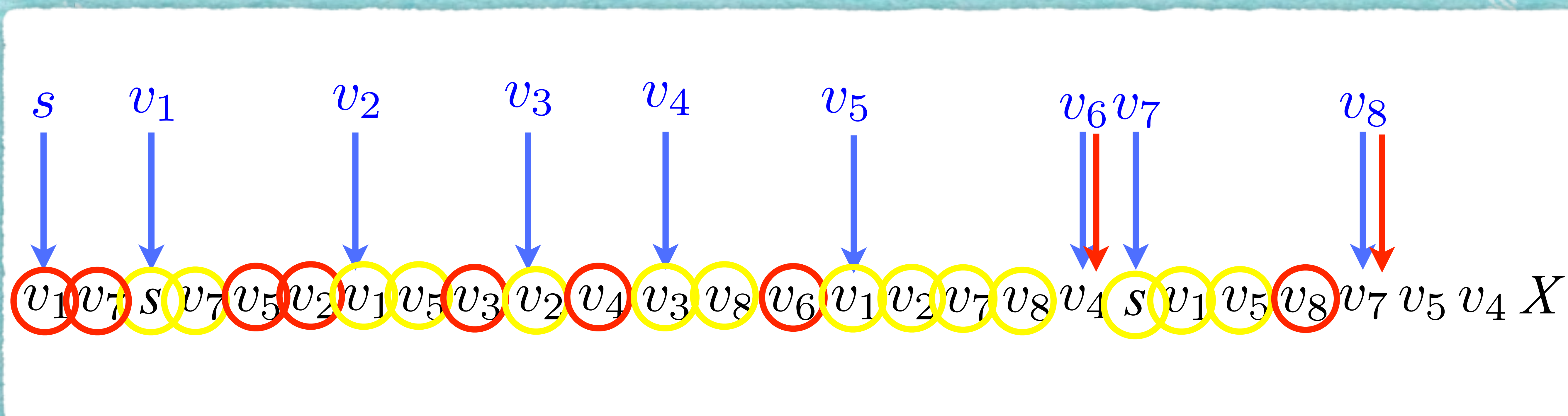
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

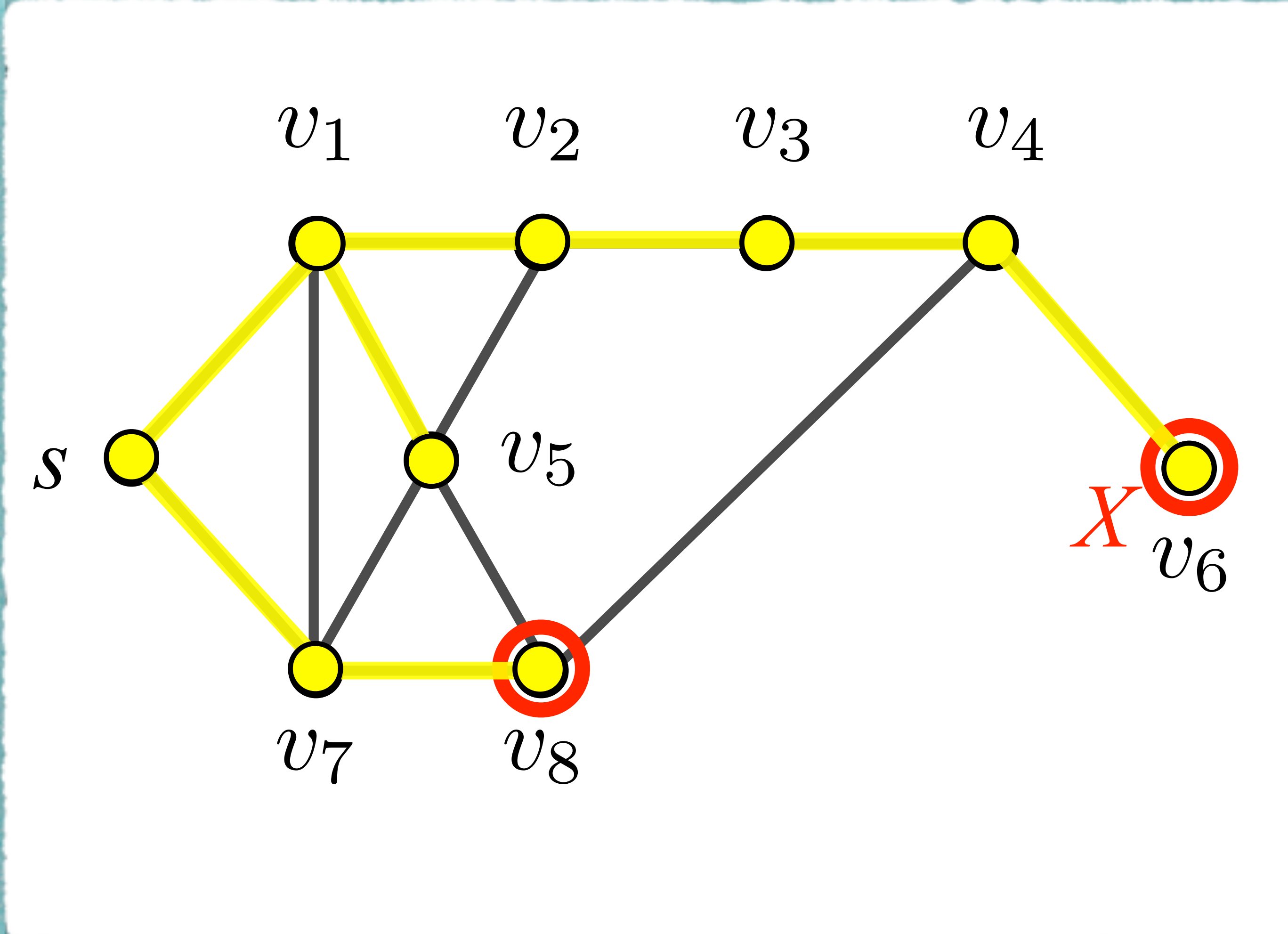
INPUT: Graph $G = (V,E)$, Knoten s

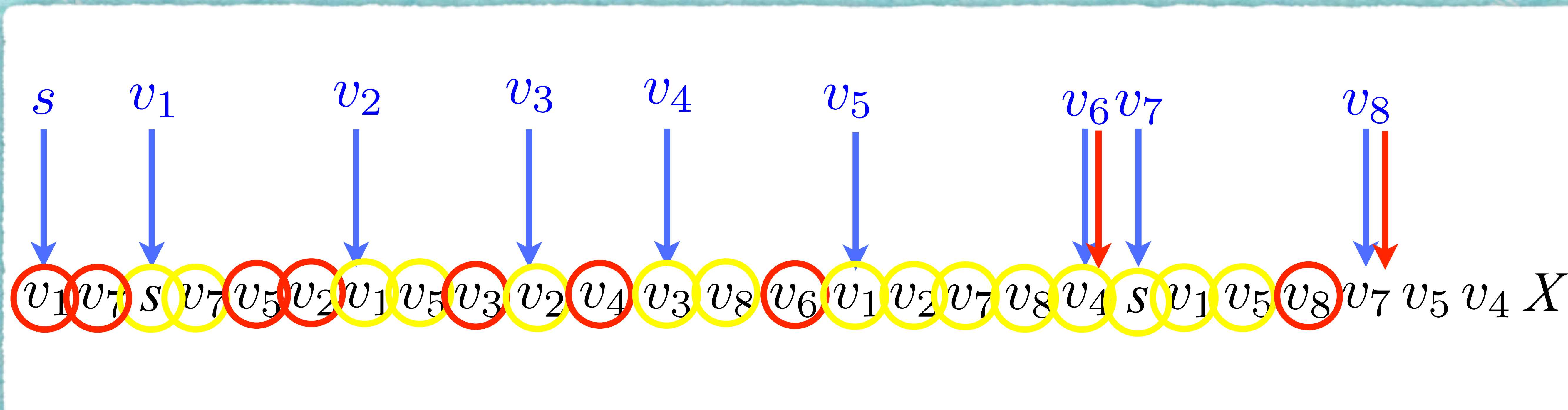
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP



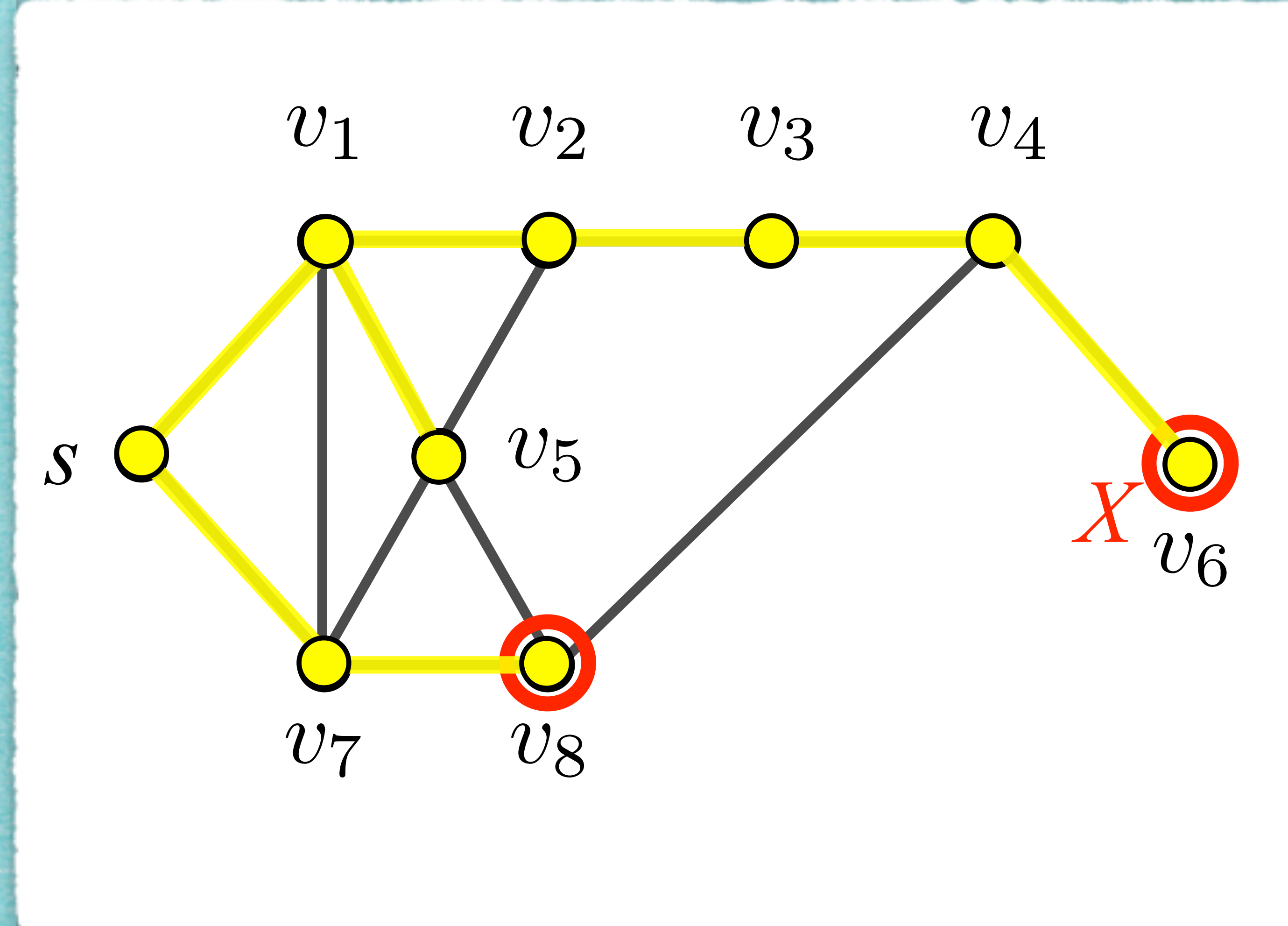


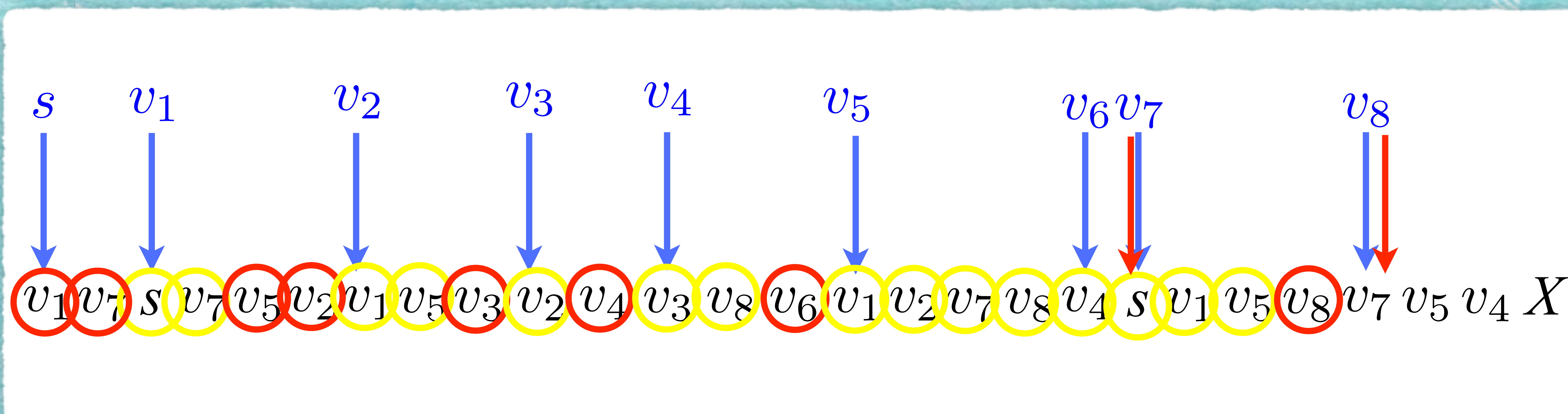
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
     - }

   }
3. STOP





Algorithmus 3.7

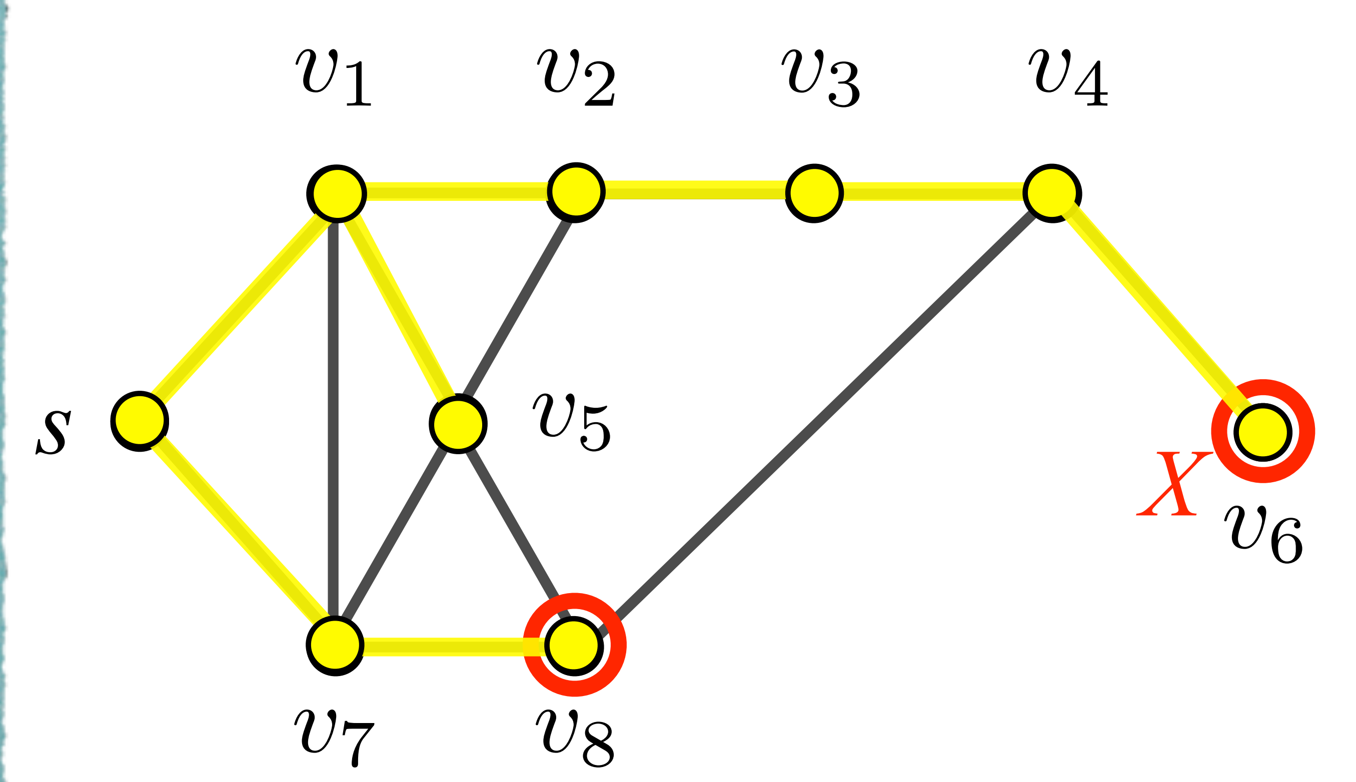
INPUT: Graph $G = (V,E)$, Knoten s

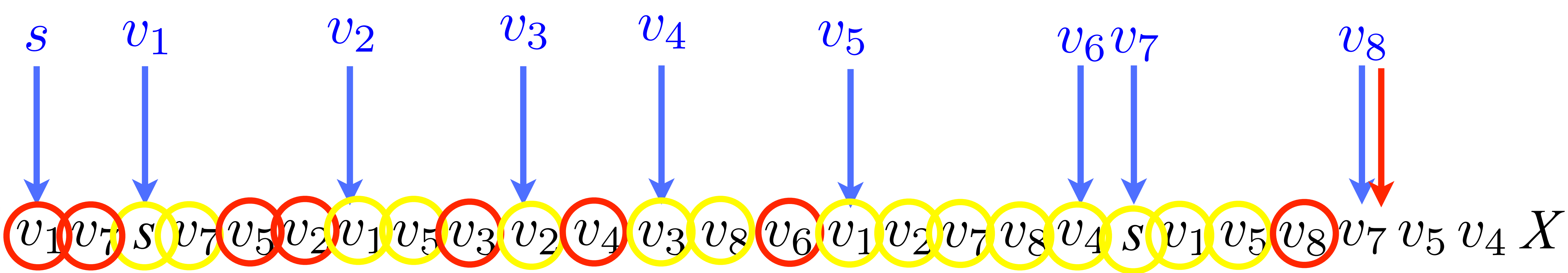
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R\neq\emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

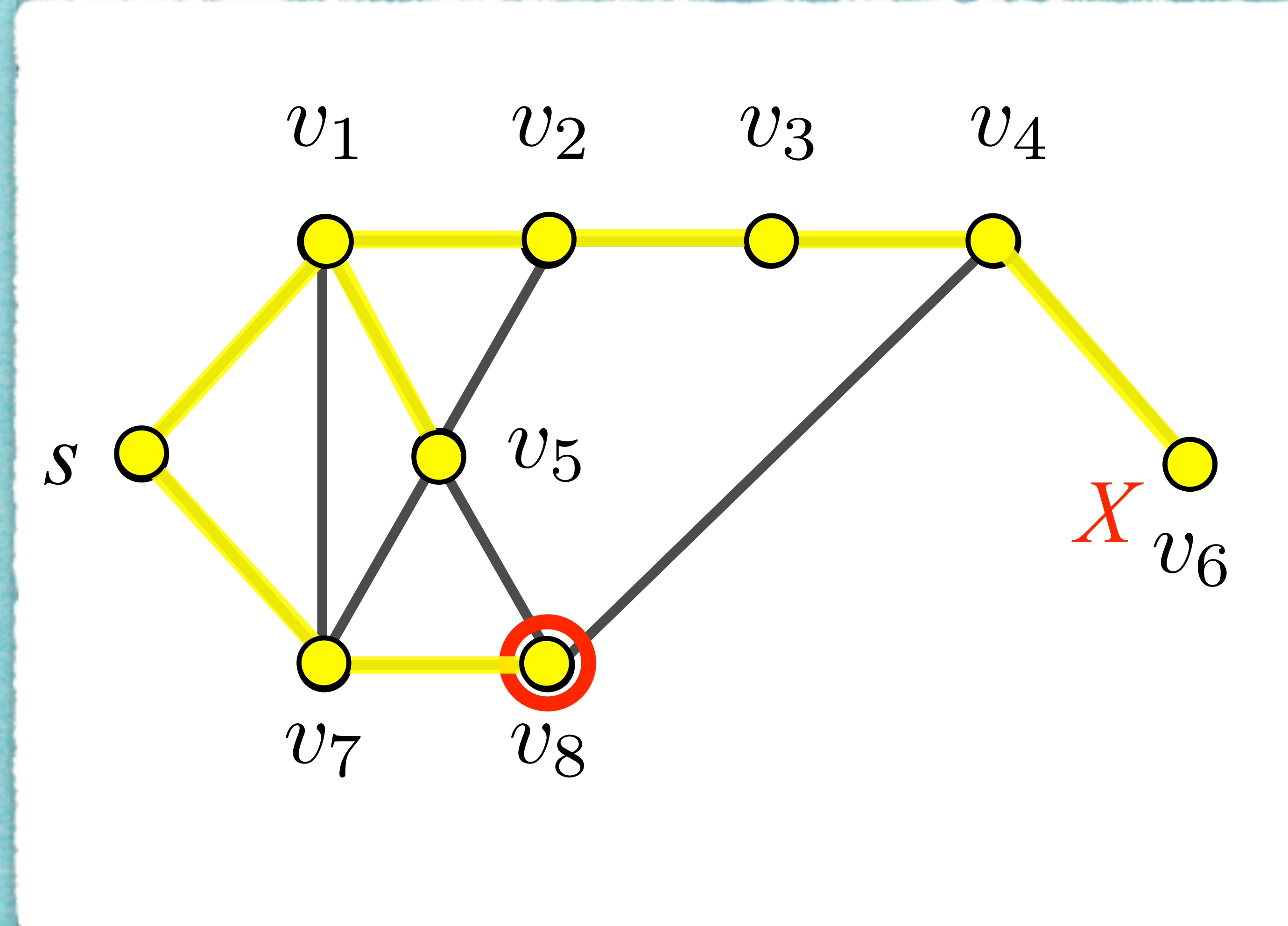
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

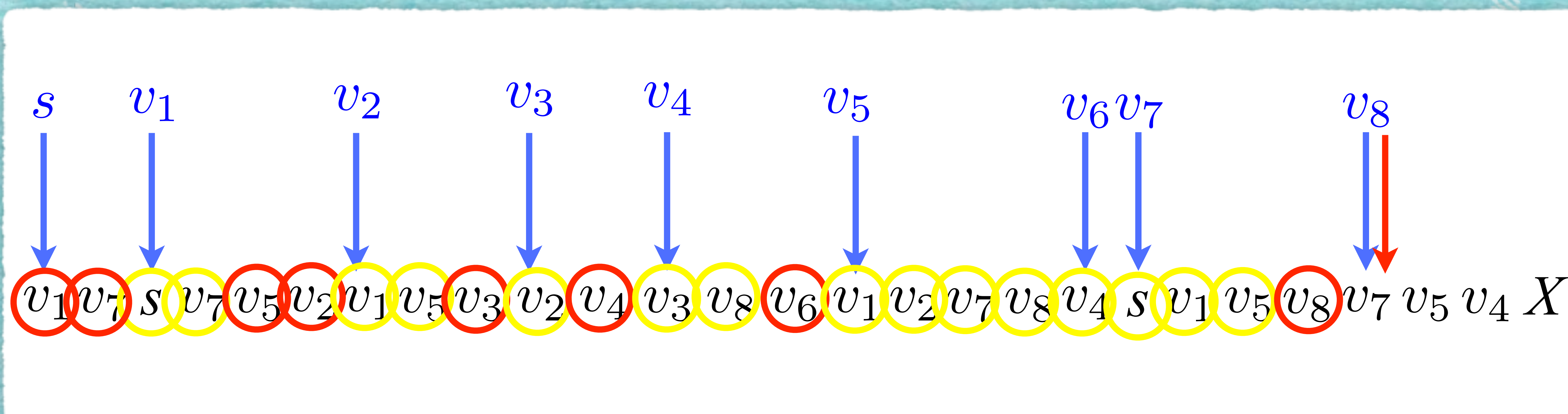
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

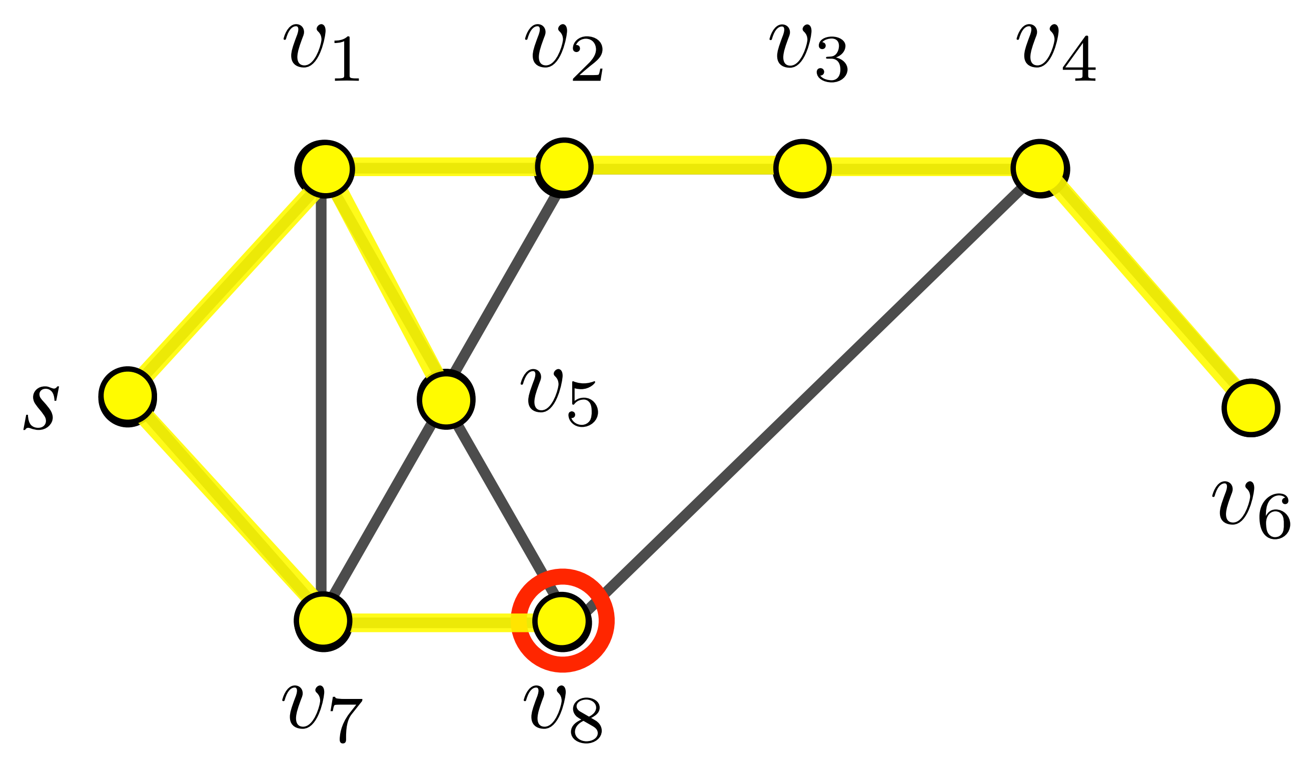
INPUT: Graph $G = (V,E)$, Knoten s

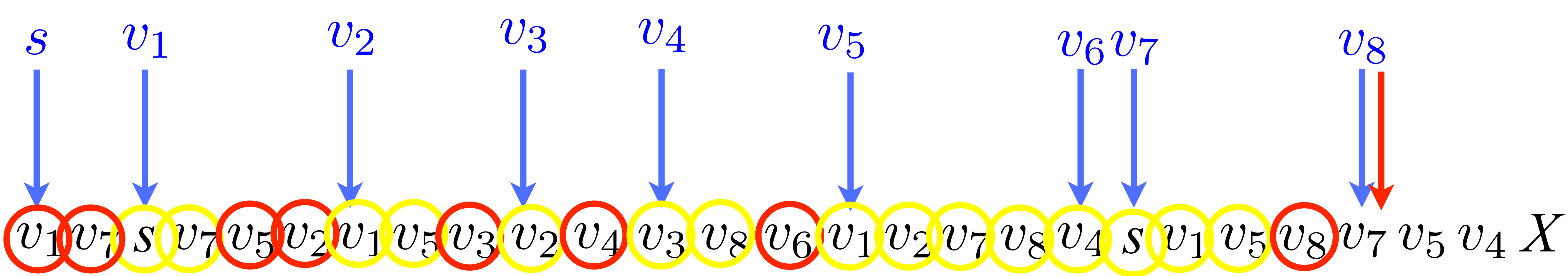
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

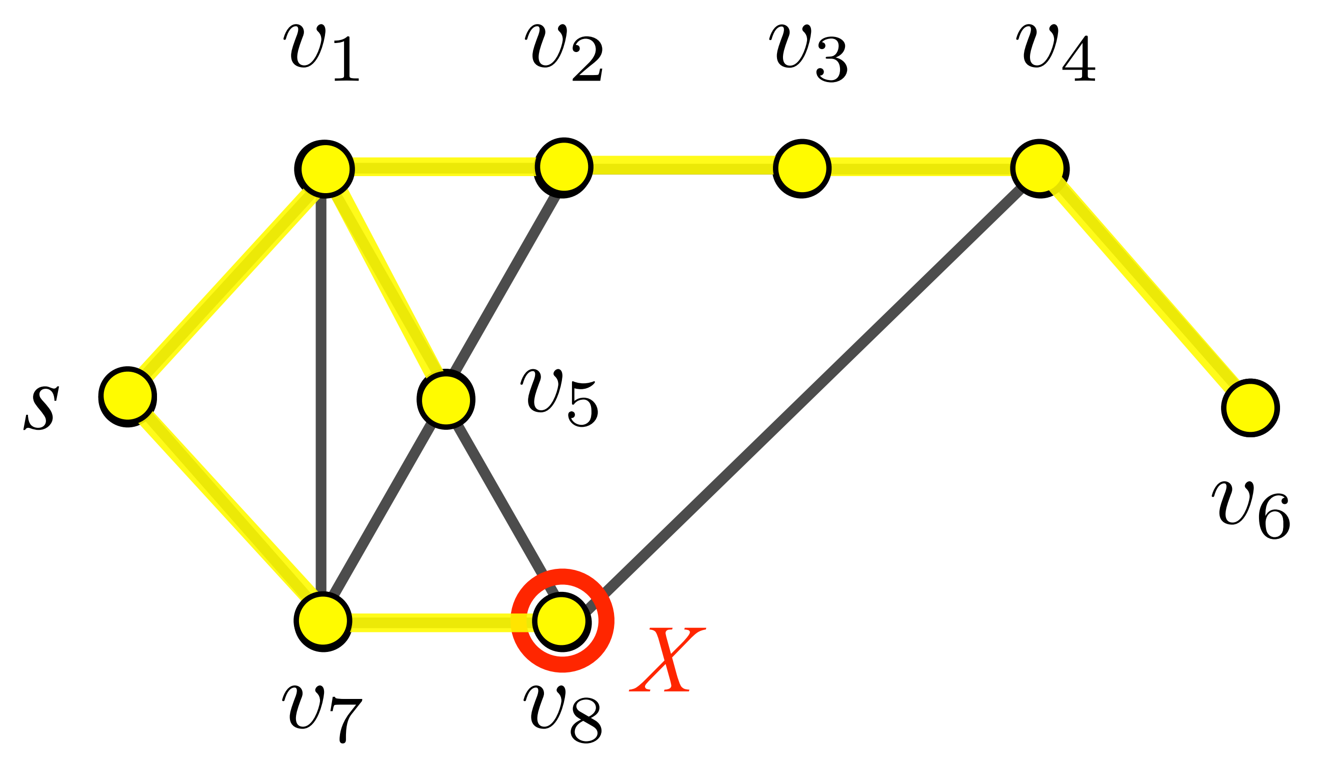
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

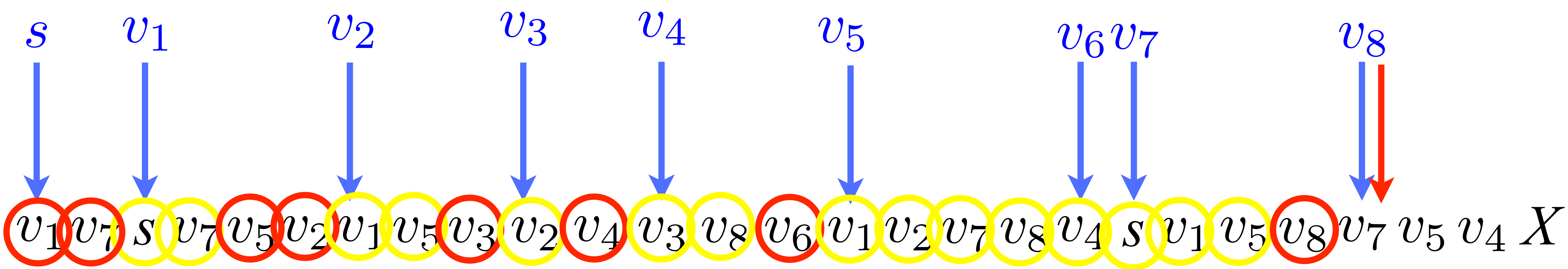
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP



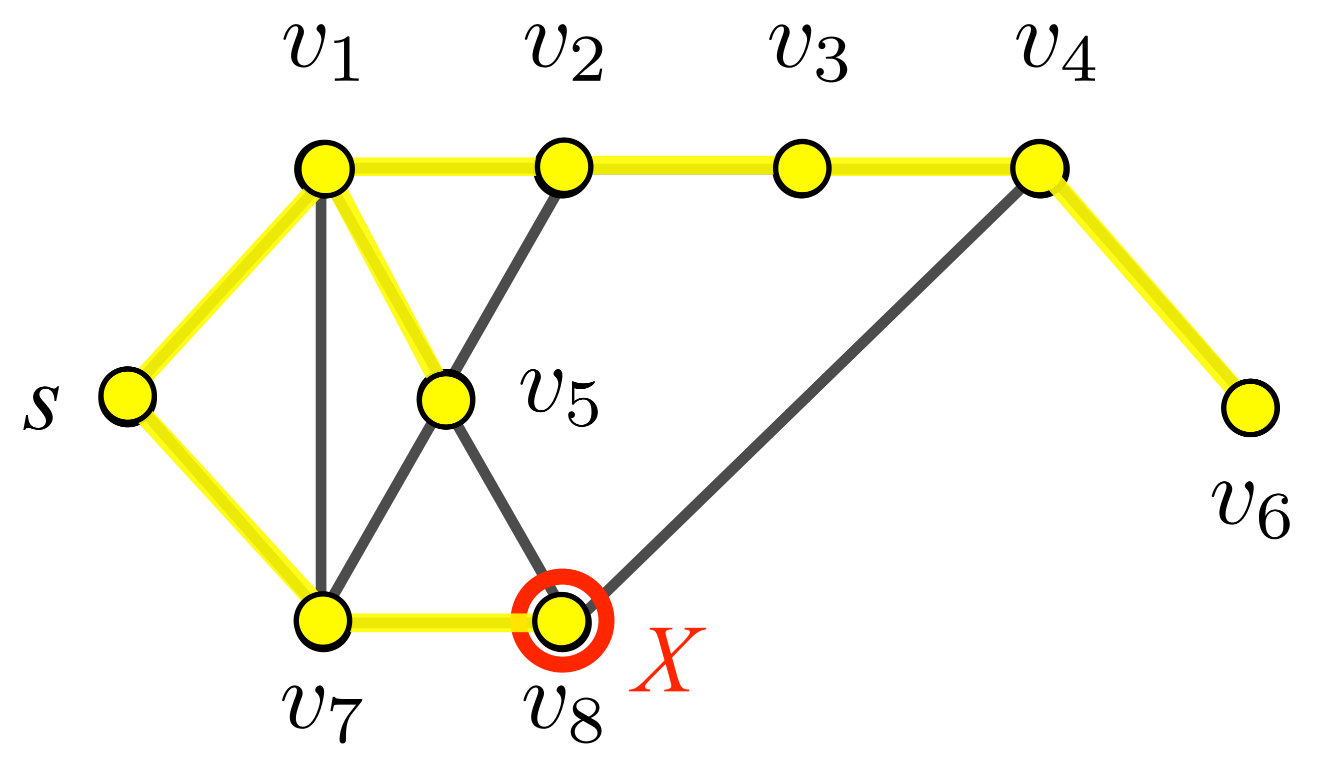


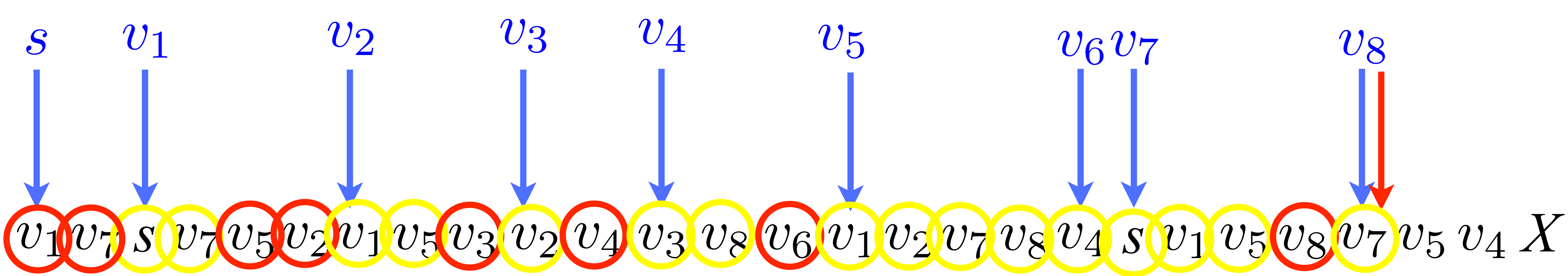
Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    2.1. Wähle $v \in R$
    2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
        2.2.1. $R:=R\setminus\{v\}$
    2.3. ELSE {
        2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
        2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
        }
    }
3. STOP

2





Algorithmus 3.7

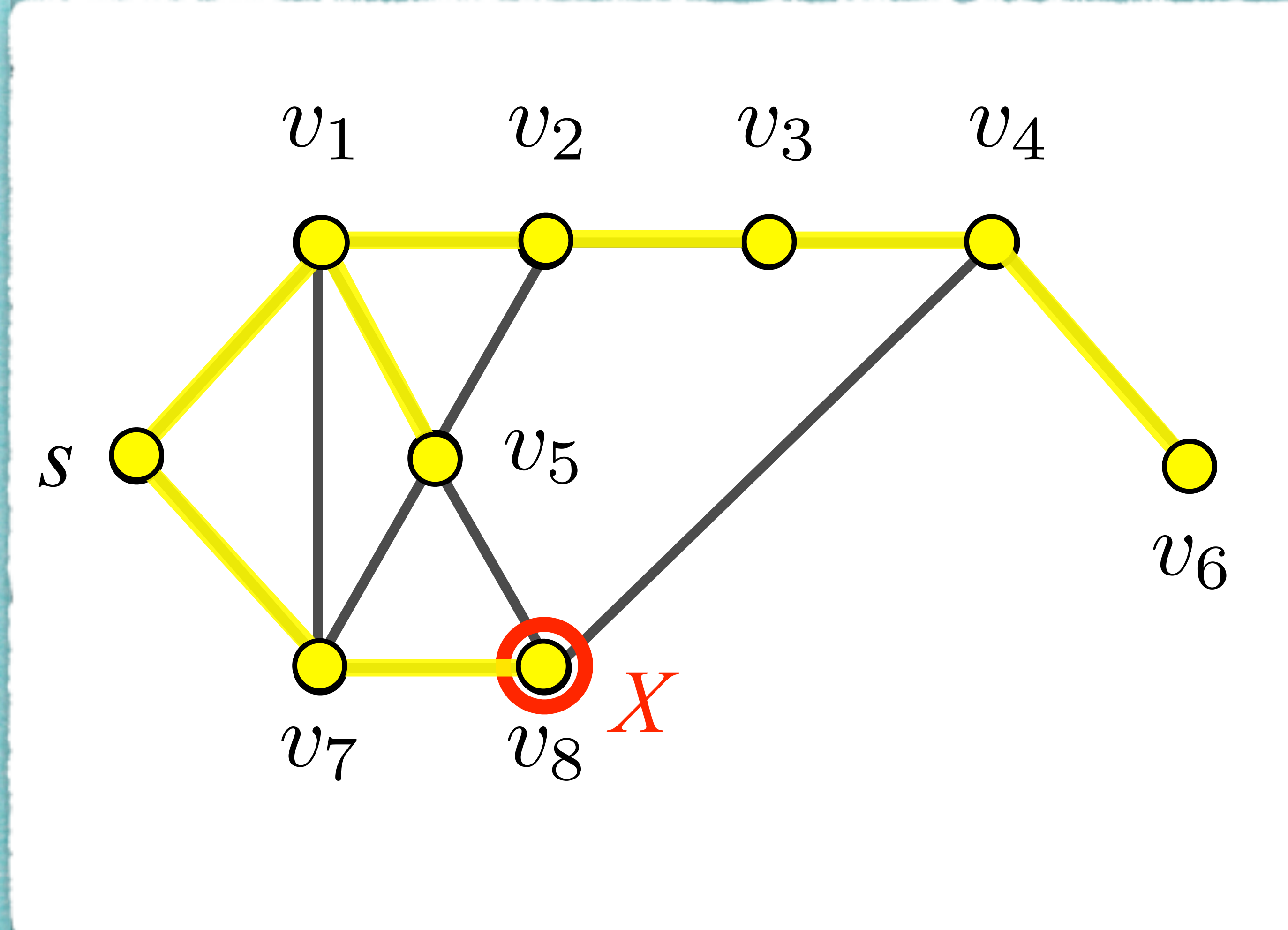
INPUT: Graph G = (V,E), Knoten s

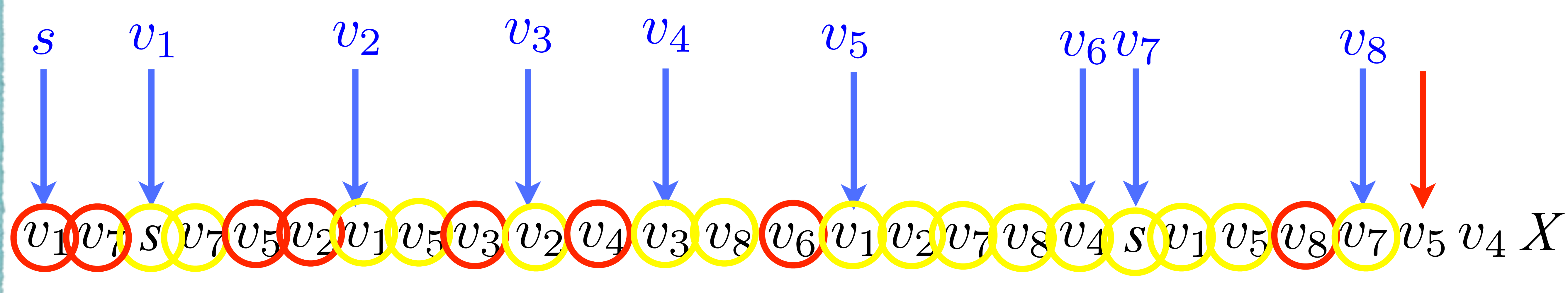
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

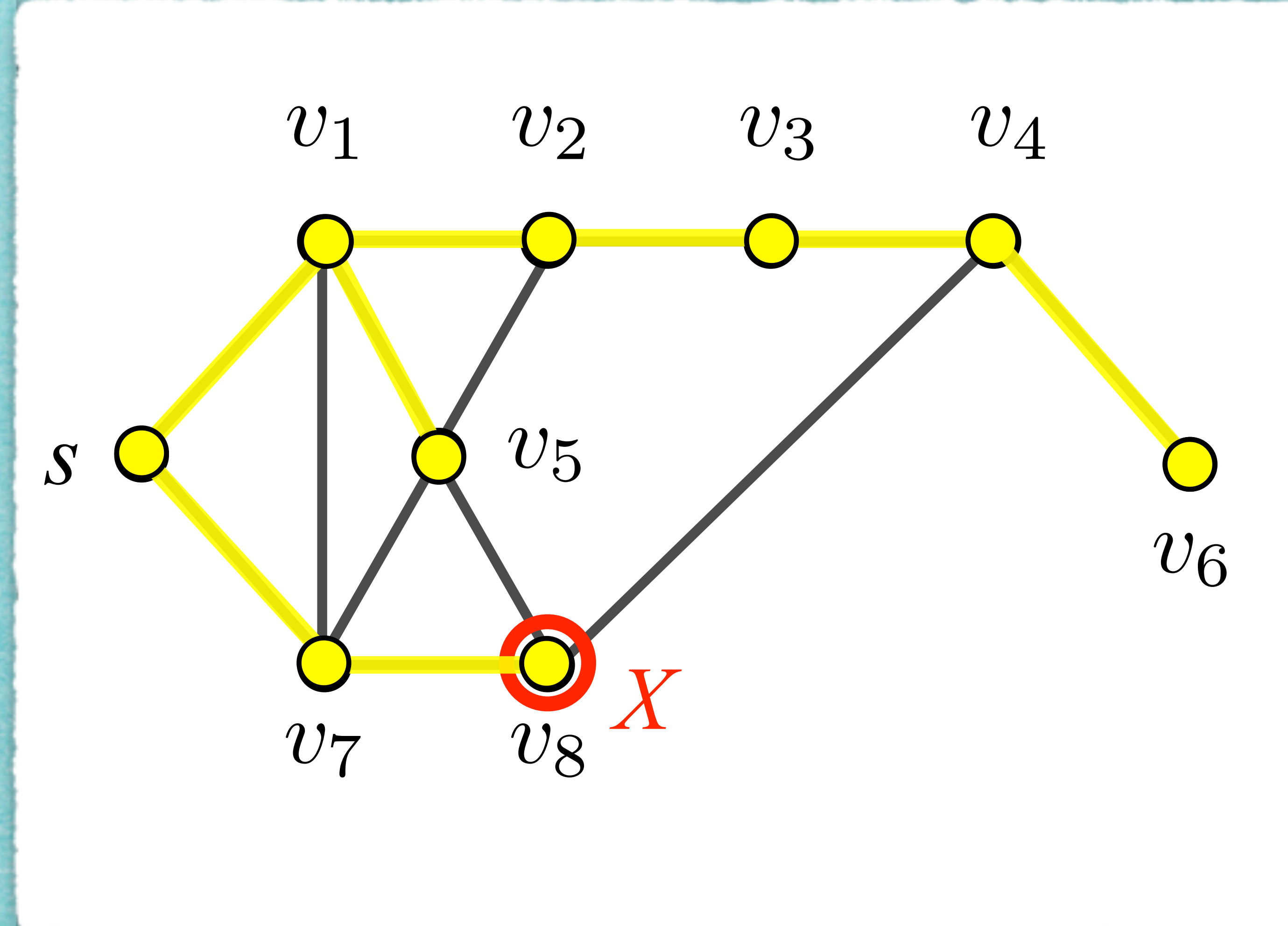
INPUT: Graph G = (V,E), Knoten s

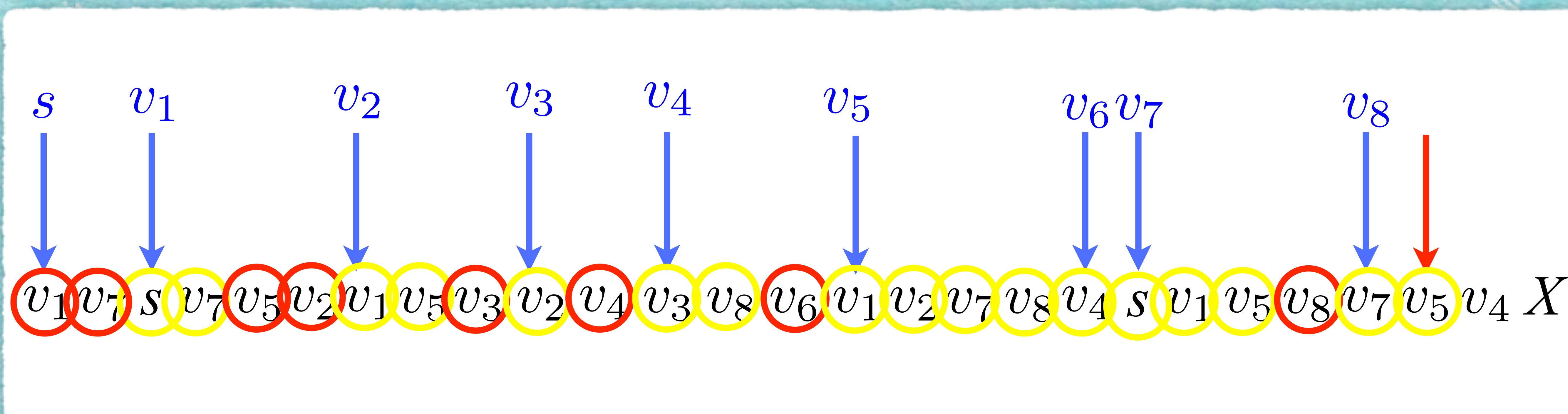
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

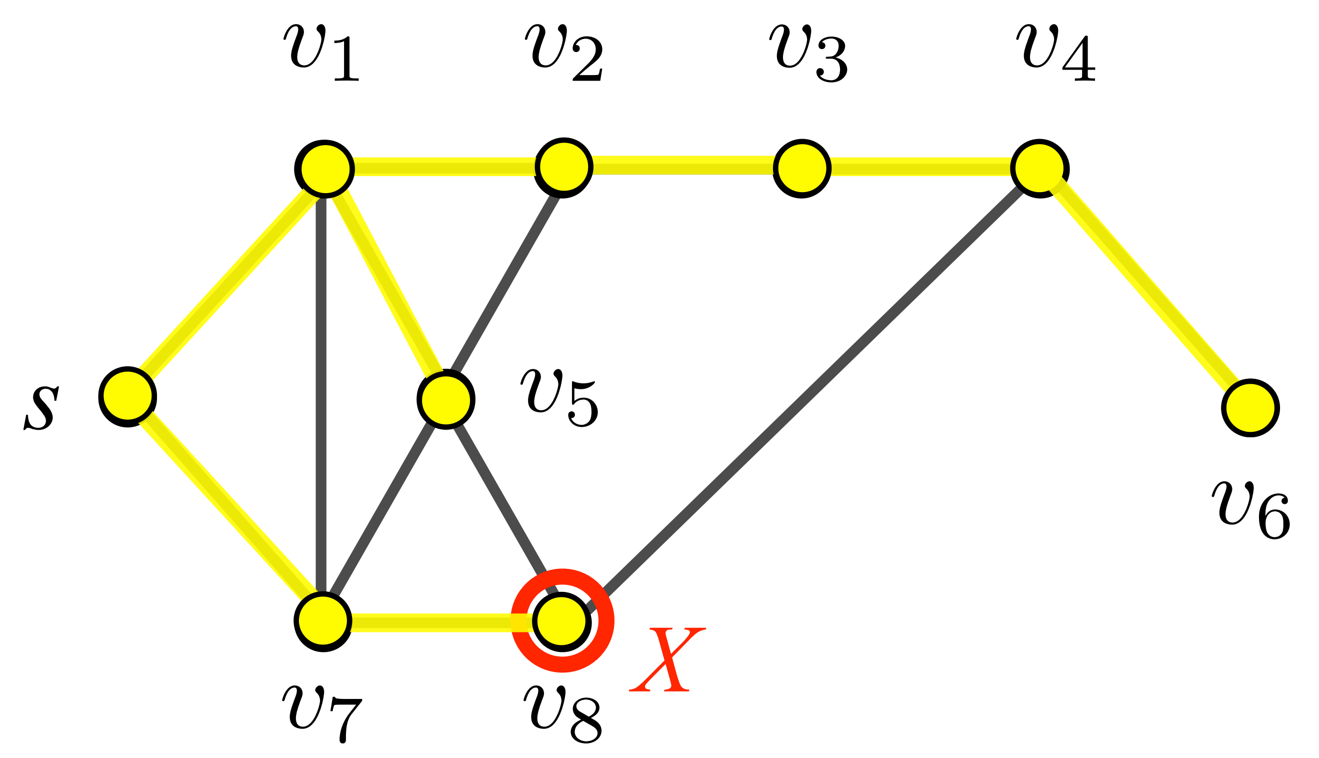
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

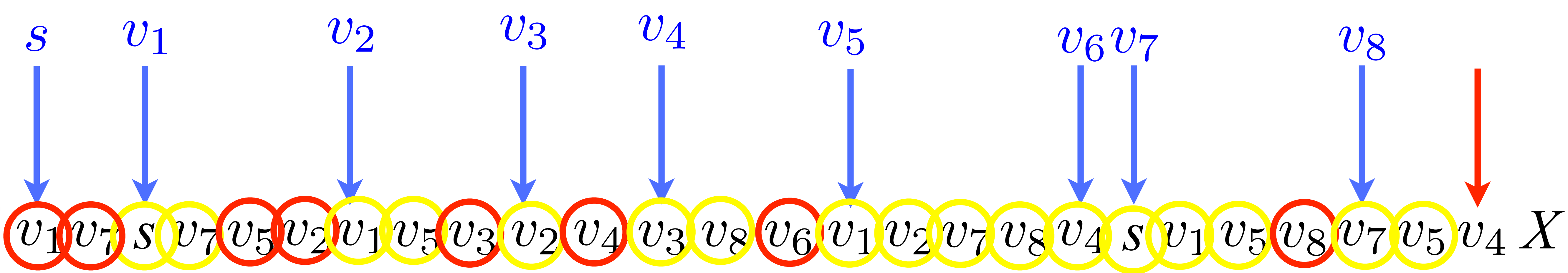
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

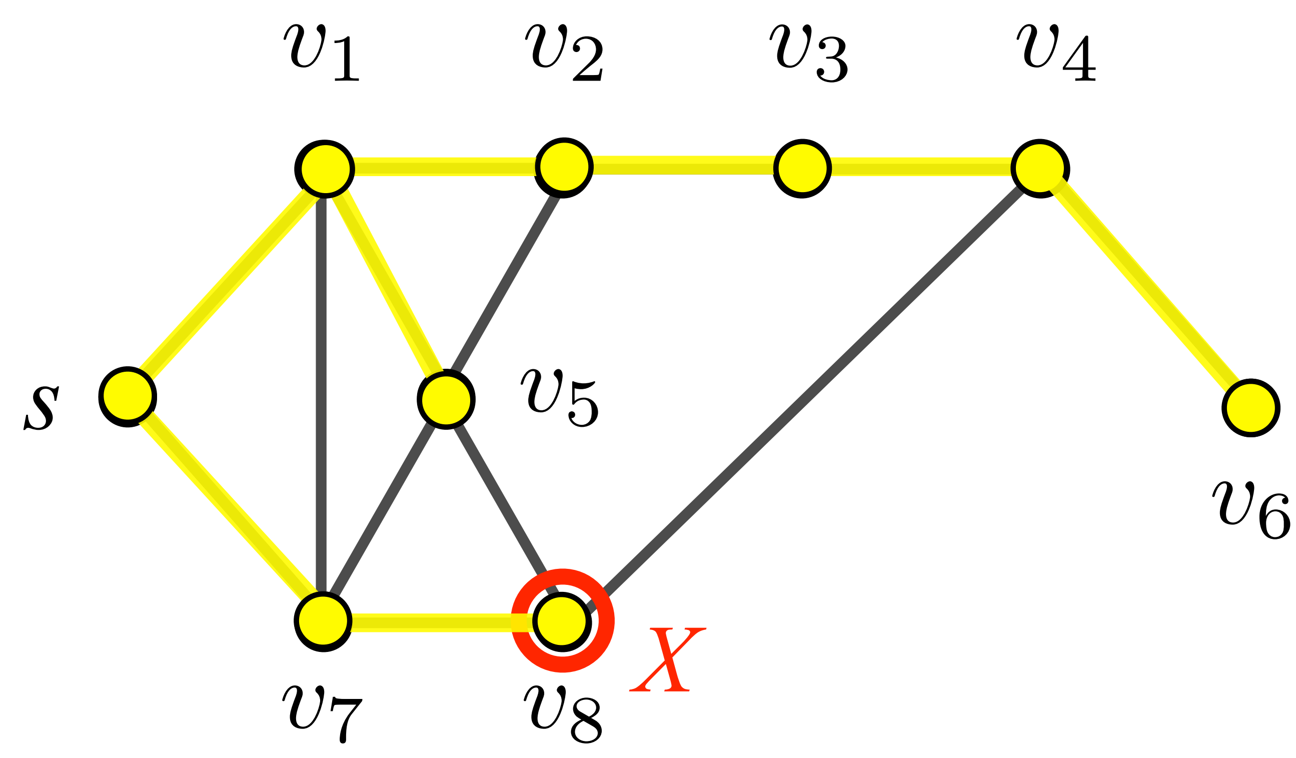
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

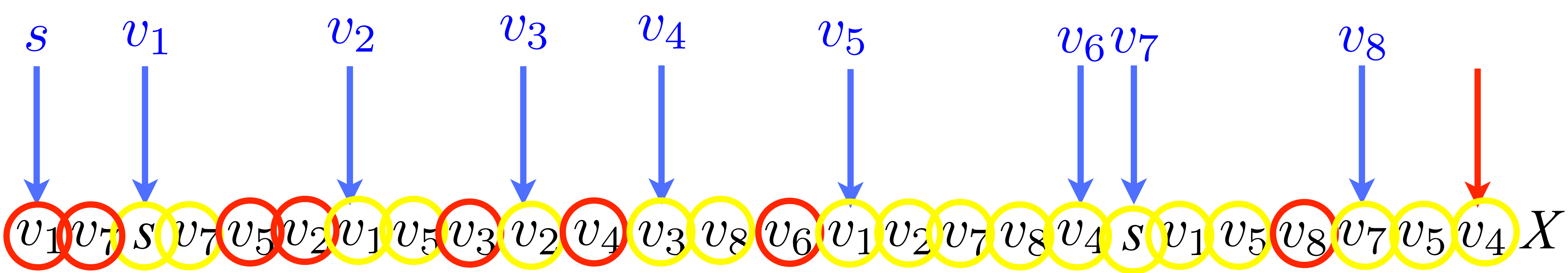
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

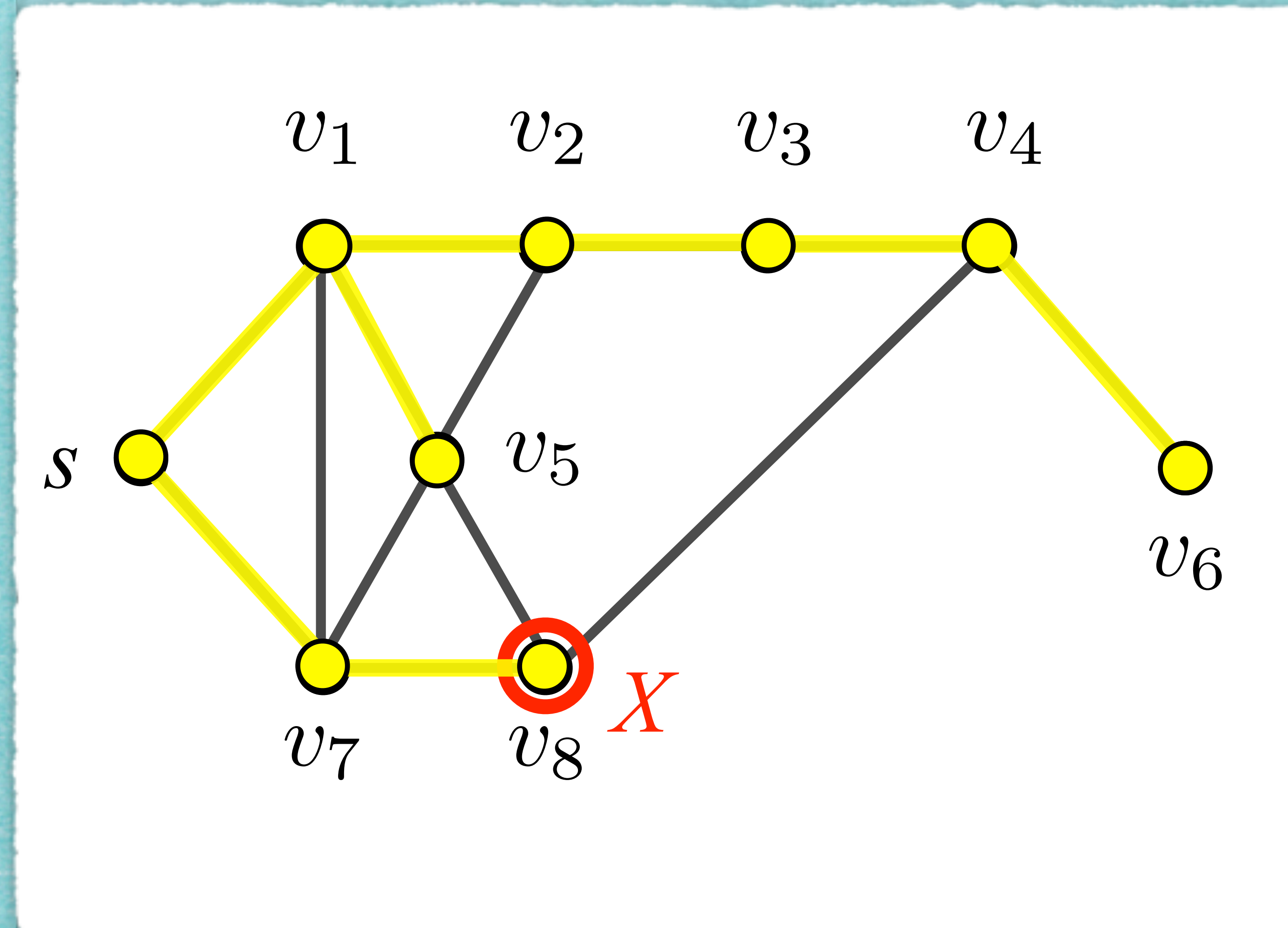
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

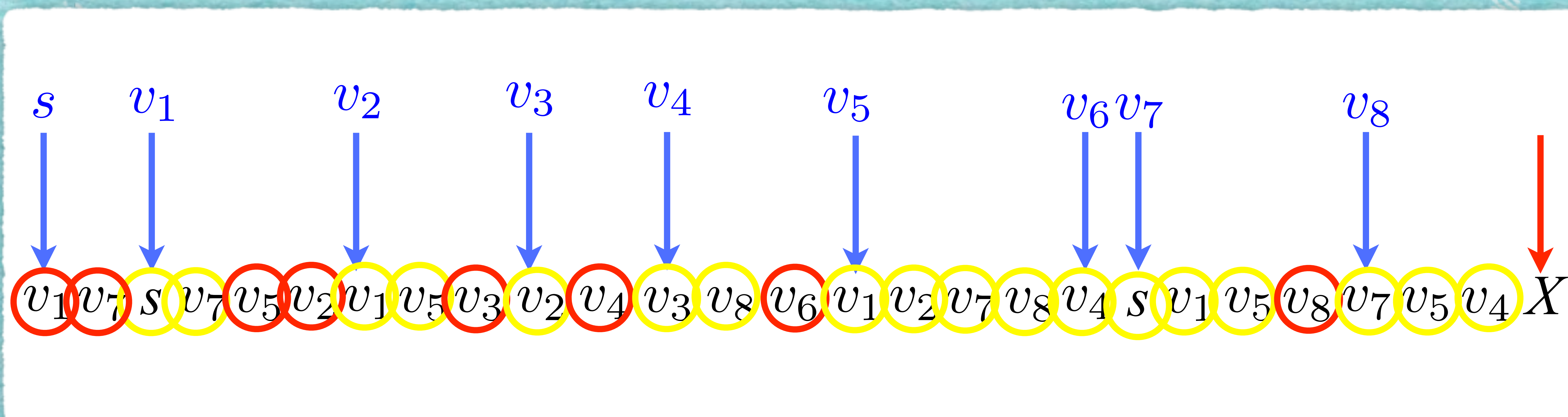
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

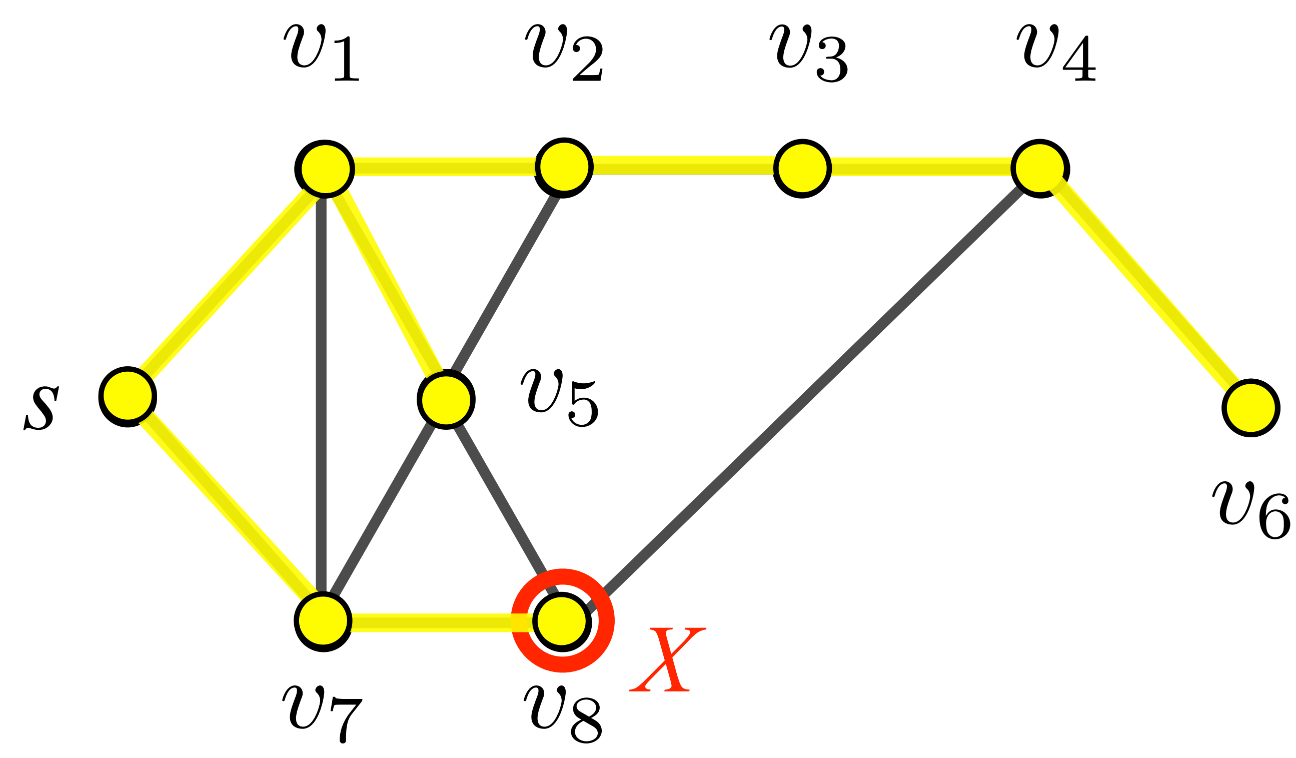
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

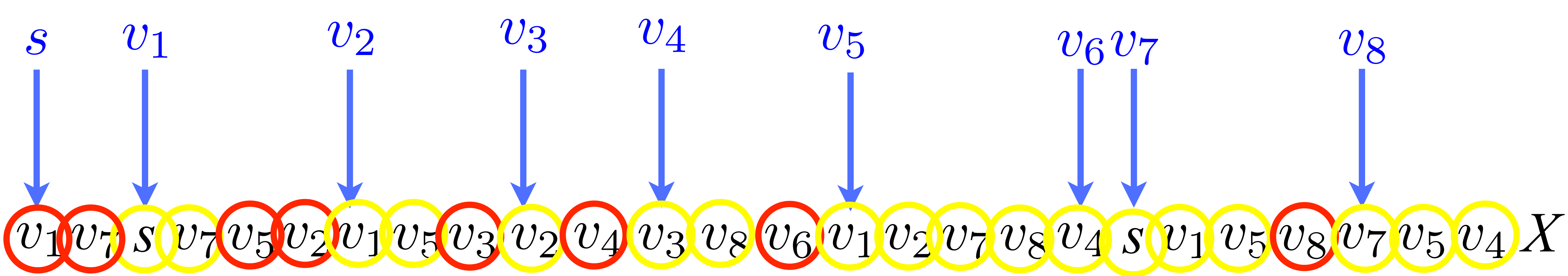
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2





Algorithmus 3.7

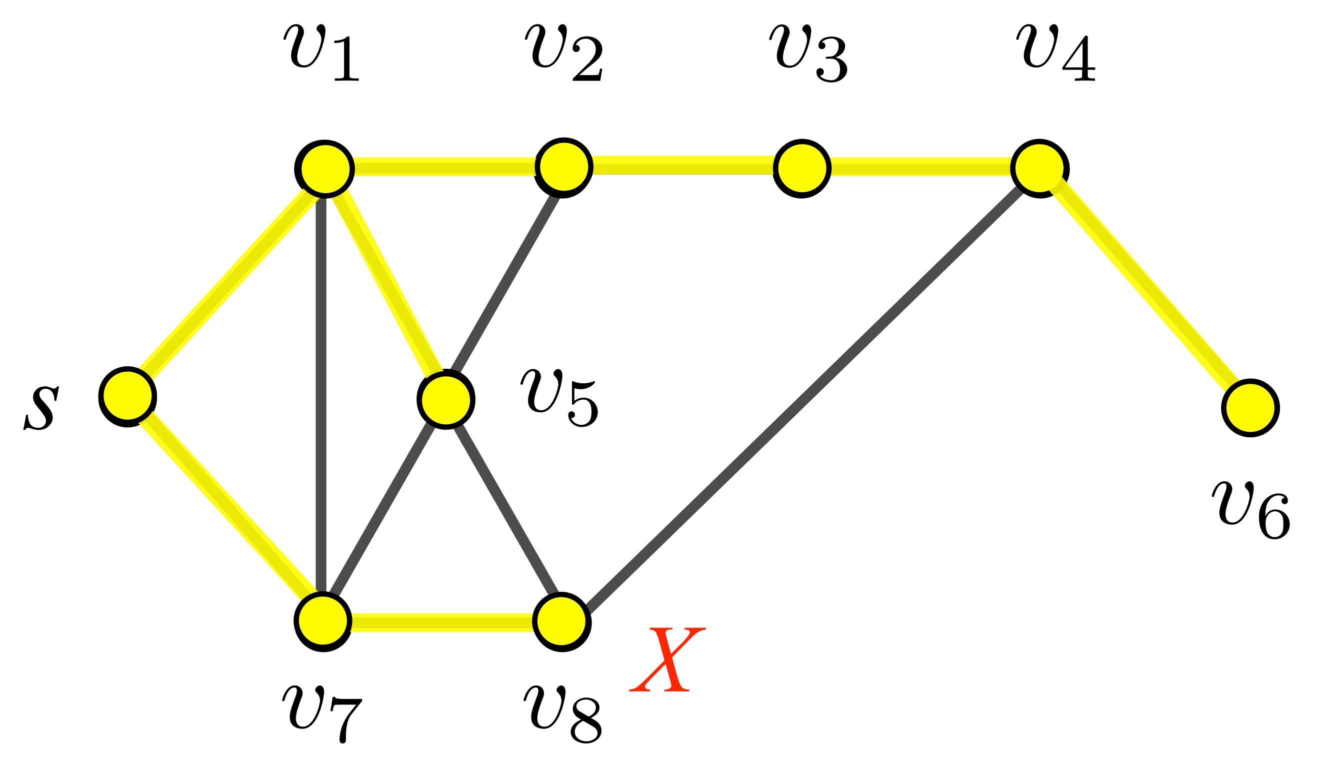
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

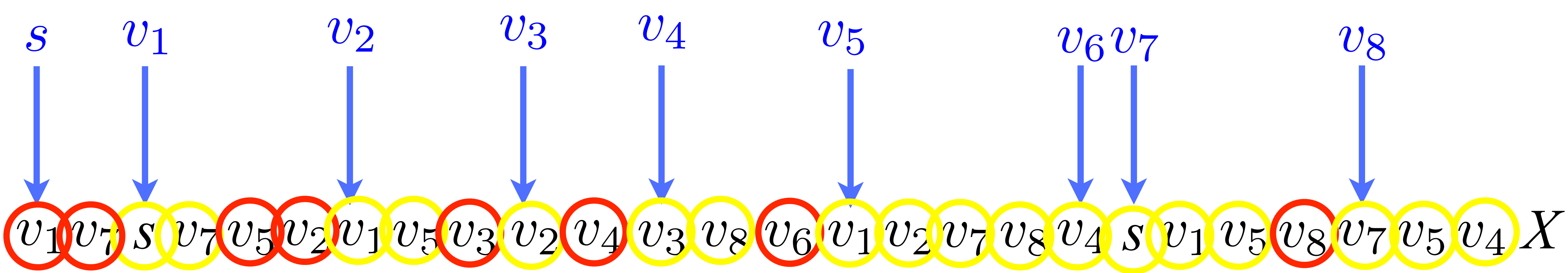
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP

2





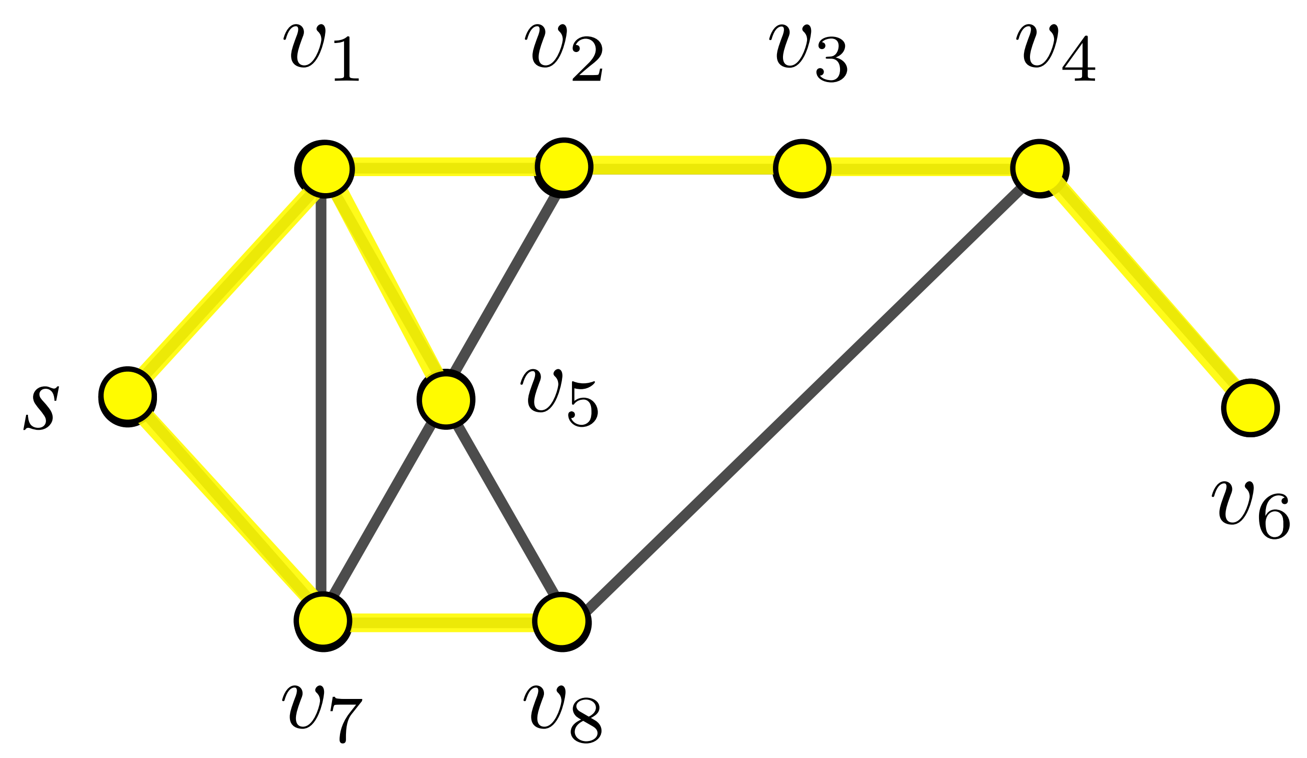
Algorithmus 3.7

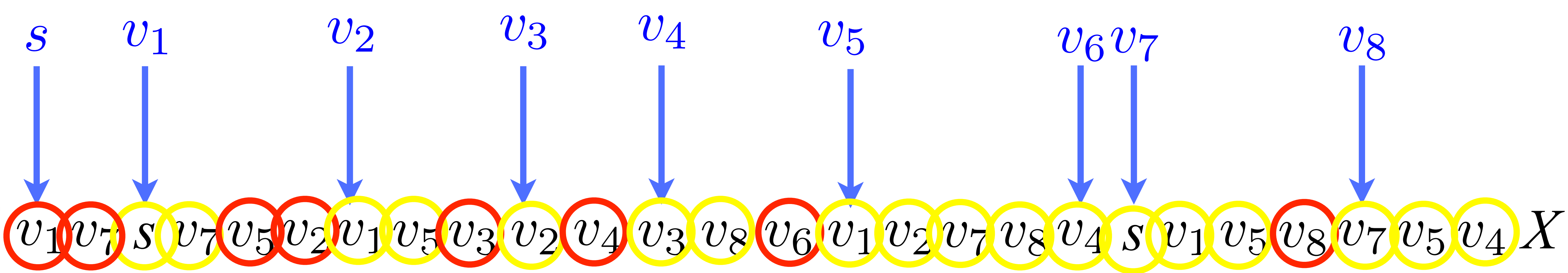
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v ∈ R
   - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w} ∈ E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

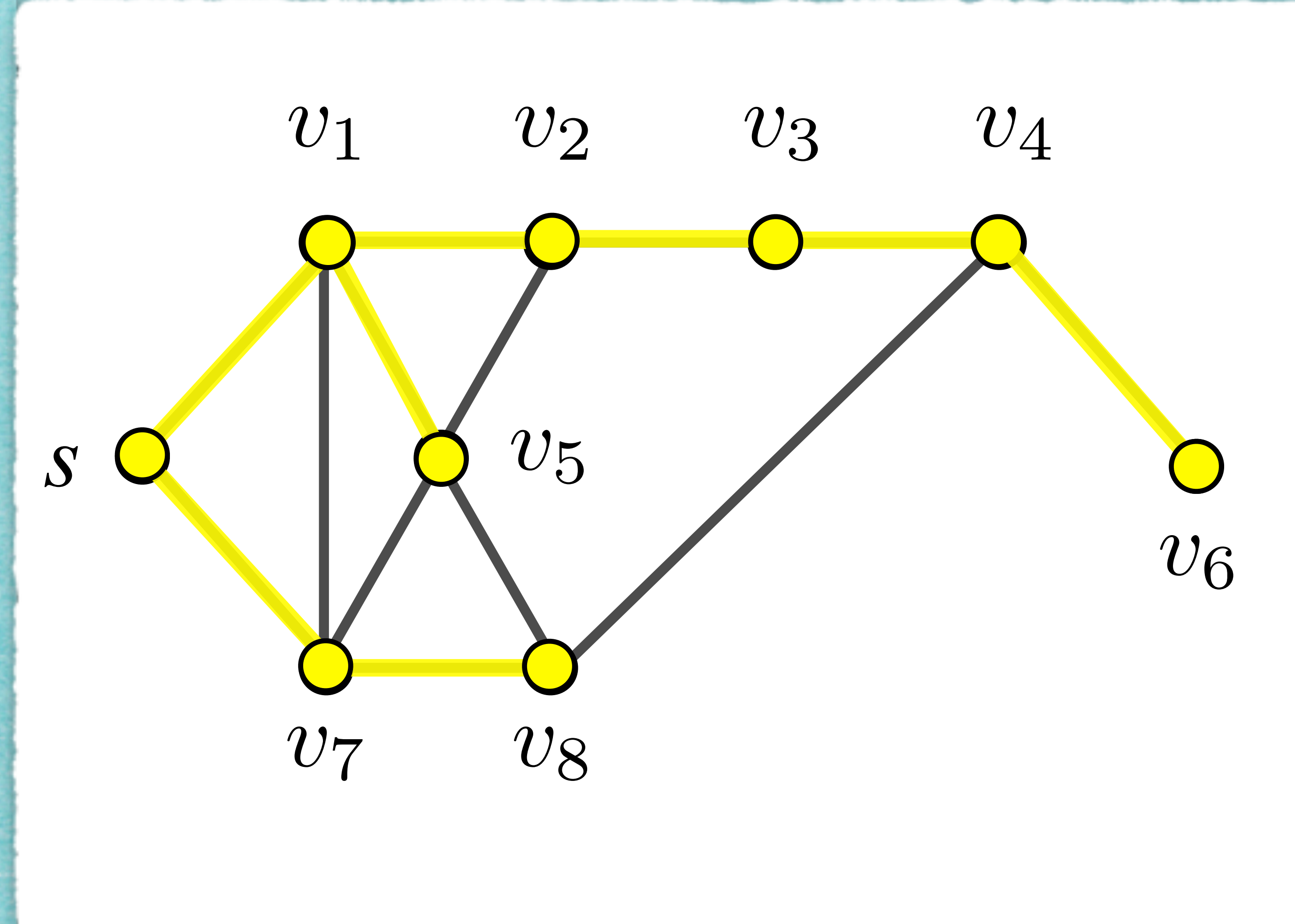
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

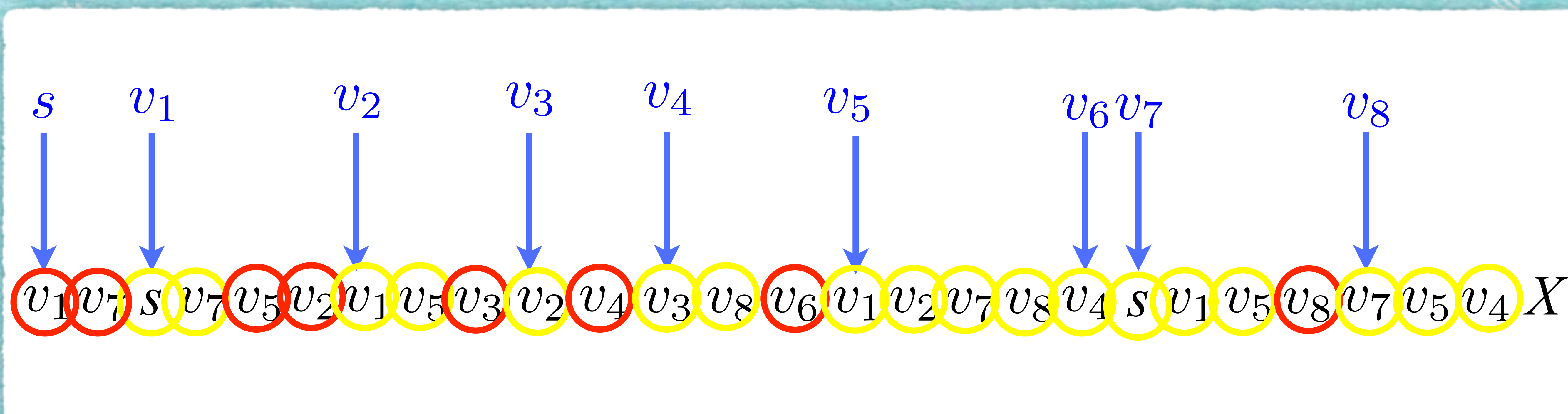
     }

   }

STOP

2





Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

   }
3. STOP

Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   2.1. Wähle $v\in R$
   2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
      2.2.1. $R:=R\setminus\{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP

Adjazenzliste!

Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

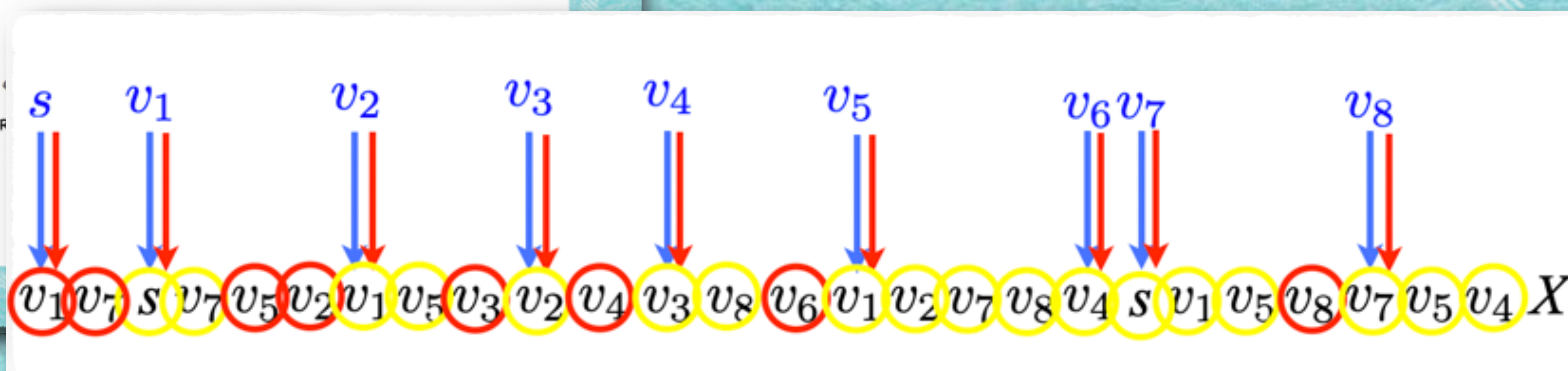
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist, Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v\in R$
   - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w
     - 2.3.2. Setze R := R

     }

   }
3. STOP

Adjazenzliste!



Satz 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

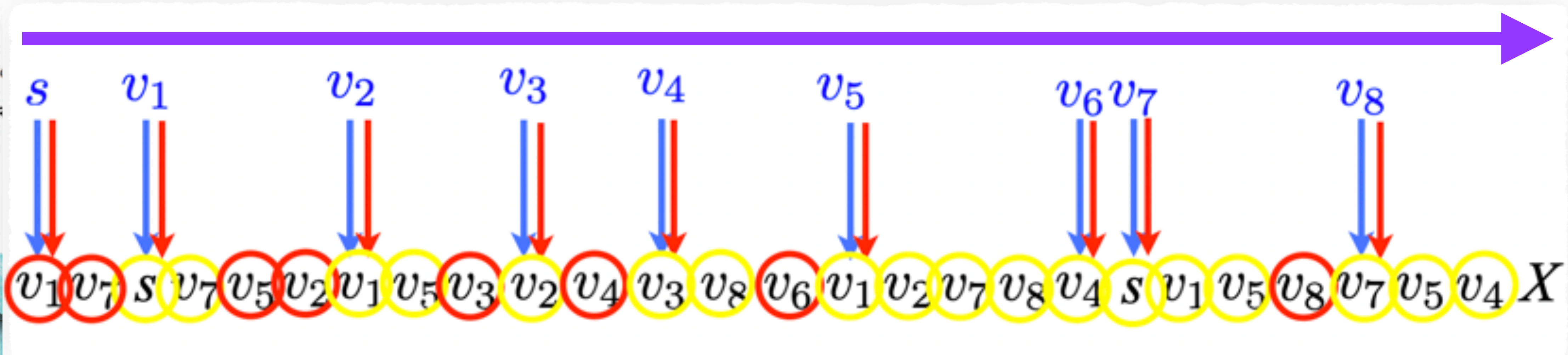
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

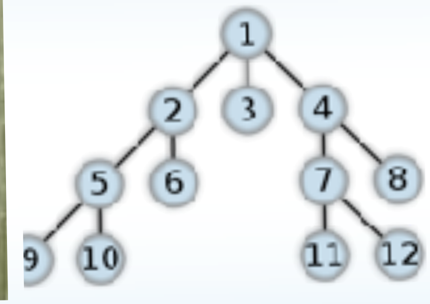
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w
     - 2.3.2. Setze R := R

     }

   }
3. STOP

Adjazenzliste!





Kapitel 3.9: Eigenschaften von DFS und BFS

Algorithmen und Datenstrukturen
WS 2020/21

Prof. Dr. Sándor Fekete

3.9 DFS vs. BFS

3.9 DFS vs. BFS

Einfach gesagt:

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

3.9 DFS vs. BFS

<u>Einfach gesagt:</u>

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

Konkret:

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

Konkret:

- ***DFS ist gut geeignet für die Suche nach einem Ausweg aus einem Labyrinth.***

3.9 DFS vs. BFS

Einfach gesagt:

- ***DFS ist eine bestmögliche individuelle Suchstrategie mit lokaler Information.***
- ***BFS ist eine bestmögliche kooperative Suchstrategie mit globaler Information.***

Konkret:

- ***DFS ist gut geeignet für die Suche nach einem Ausweg aus einem Labyrinth.***
- ***BFS ist gut geeignet für die Suche nach kürzesten Wegen in einem Graphen.***

3.9 DFS

Satz 3.16 (Lokale Suche mit DFS)

Satz 3.16 (Lokale Suche mit DFS)

(1) DFS findet in jedem zusammenhängenden Graphen mit n Knoten einen Weg der Länge höchstens 2n-1, der alle Knoten besucht.

Satz 3.16 (Lokale Suche mit DFS)

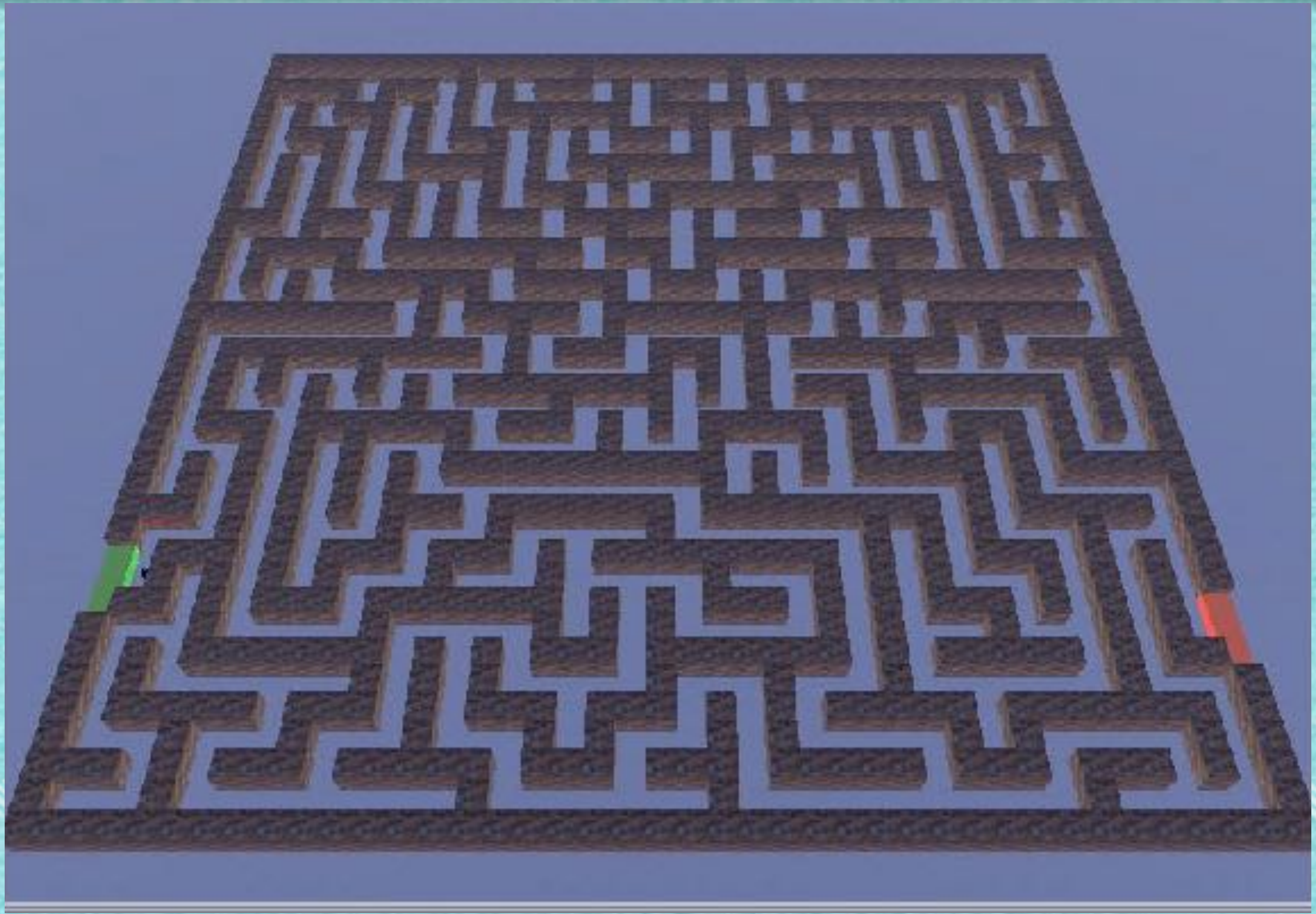
(1) DFS findet in jedem zusammenhängenden Graphen mit n Knoten einen Weg der Länge höchstens 2n-1, der alle Knoten besucht.

(2) Für jede lokale Suchstrategie gibt es einen Graphen mit n Knoten, so dass der letzte Knoten erst nach einer Weglänge von 2n-1 besucht wird.

Satz 3.16 (Lokale Suche mit DFS)

(1) DFS findet in jedem zusammenhängenden Graphen mit n Knoten einen Weg der Länge höchstens 2n-1, der alle Knoten besucht.

(2) Für jede lokale Suchstrategie gibt es einen Graphen mit n Knoten, so dass der letzte Knoten erst nach einer Weglänge von 2n-1 besucht wird.



Satz 3.16 (Lokale Suche mit DFS)

(1) DFS findet in jedem zusammenhängenden Graphen mit n Knoten einen Weg der Länge höchstens 2n-1, der alle Knoten besucht.

(2) Für jede lokale Suchstrategie gibt es einen Graphen mit n Knoten, so dass der letzte Knoten erst nach einer Weglänge von 2n-1 besucht wird.

Satz 3.16 (Lokale Suche mit DFS)

(1) ***DFS findet in jedem zusammenhängenden Graphen mit n Knoten einen Weg der Länge höchstens 2n-1, der alle Knoten besucht.***

(2) ***Für jede lokale Suchstrategie gibt es einen Graphen mit n Knoten, so dass der letzte Knoten erst nach einer Weglänge von 2n-1 besucht wird.***

Beweis: *Übung!*

Algorithmus 3.7

3.9 BFS?

<u>INPUT:</u>	Graph $G = (V,E)$, Knoten s
<u>OUTPUT:</u>	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. wähle Element $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. wähle ein $w \in V \setminus R$ mit $e=\{v,w\} \in E$;
        - 2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

      }

}

Auf die Schnelle mit der Welle

A. LOS bei „NULL"

B. Bis „ANGEKOMMEN!":

- Solange du noch nicht aufgestanden warst:
  - Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:

       2.1. aufstehen

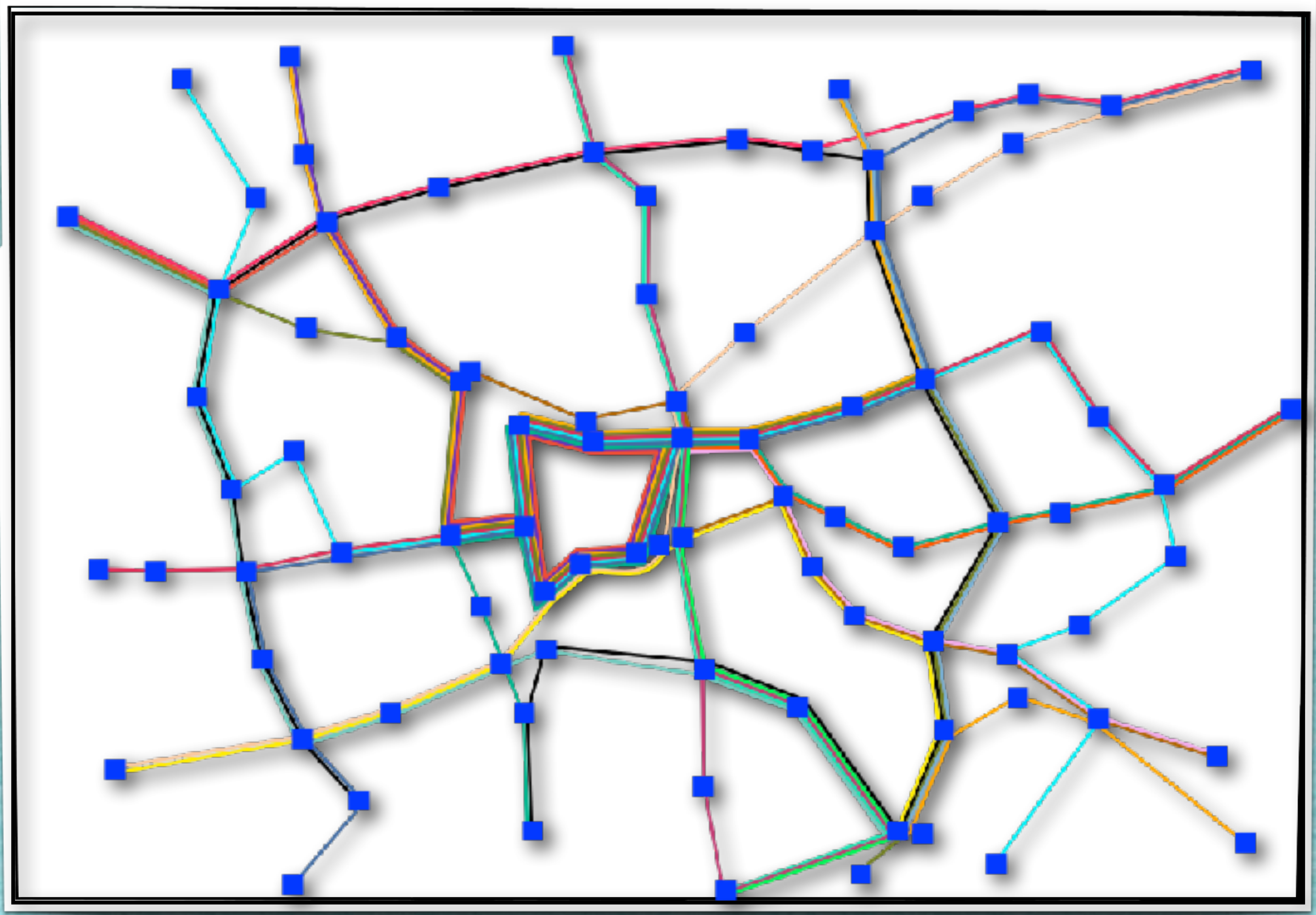
       2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!":

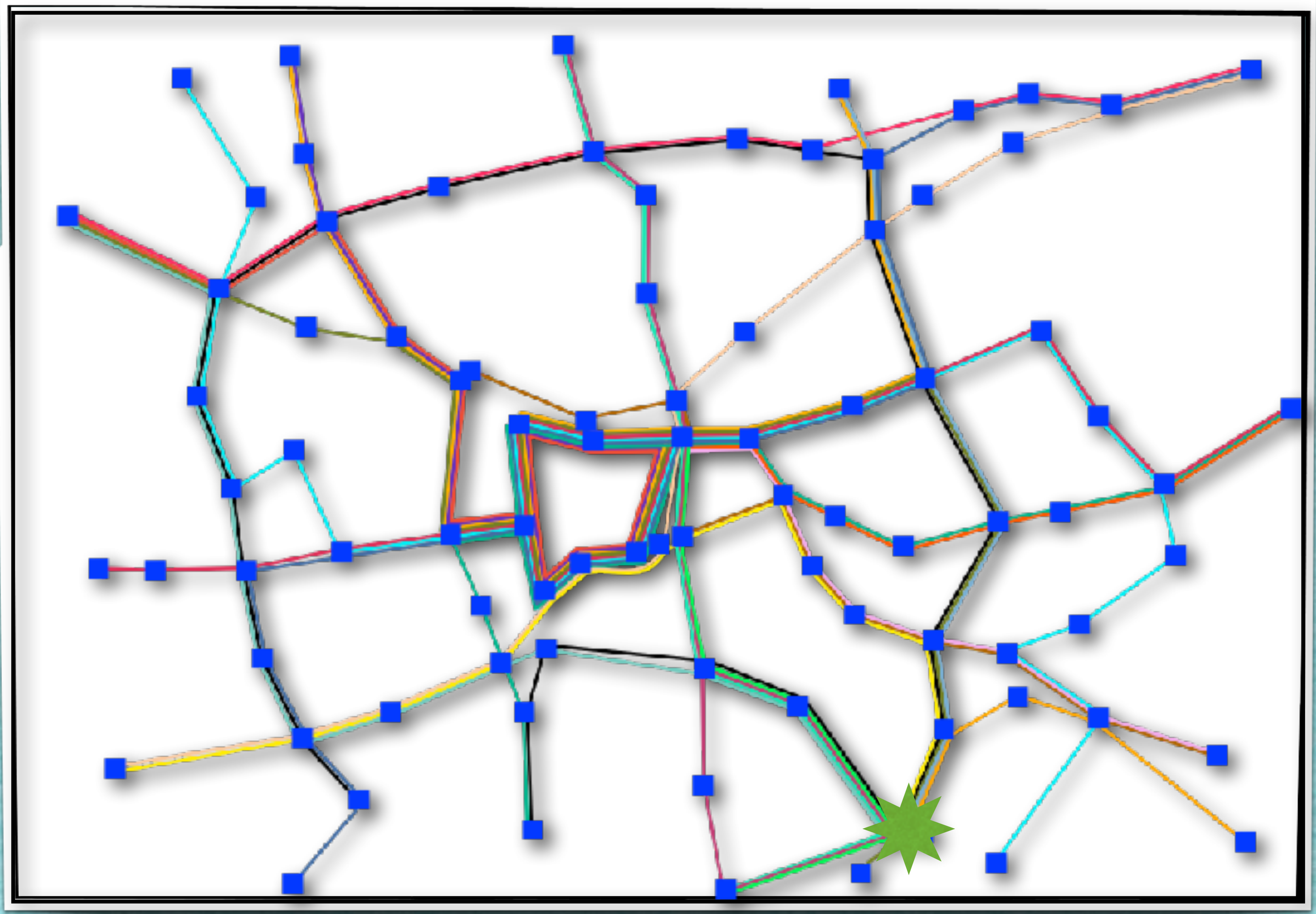
- Auf gemerkten Nachbarn zeigen

Wellenreiten in Graphen

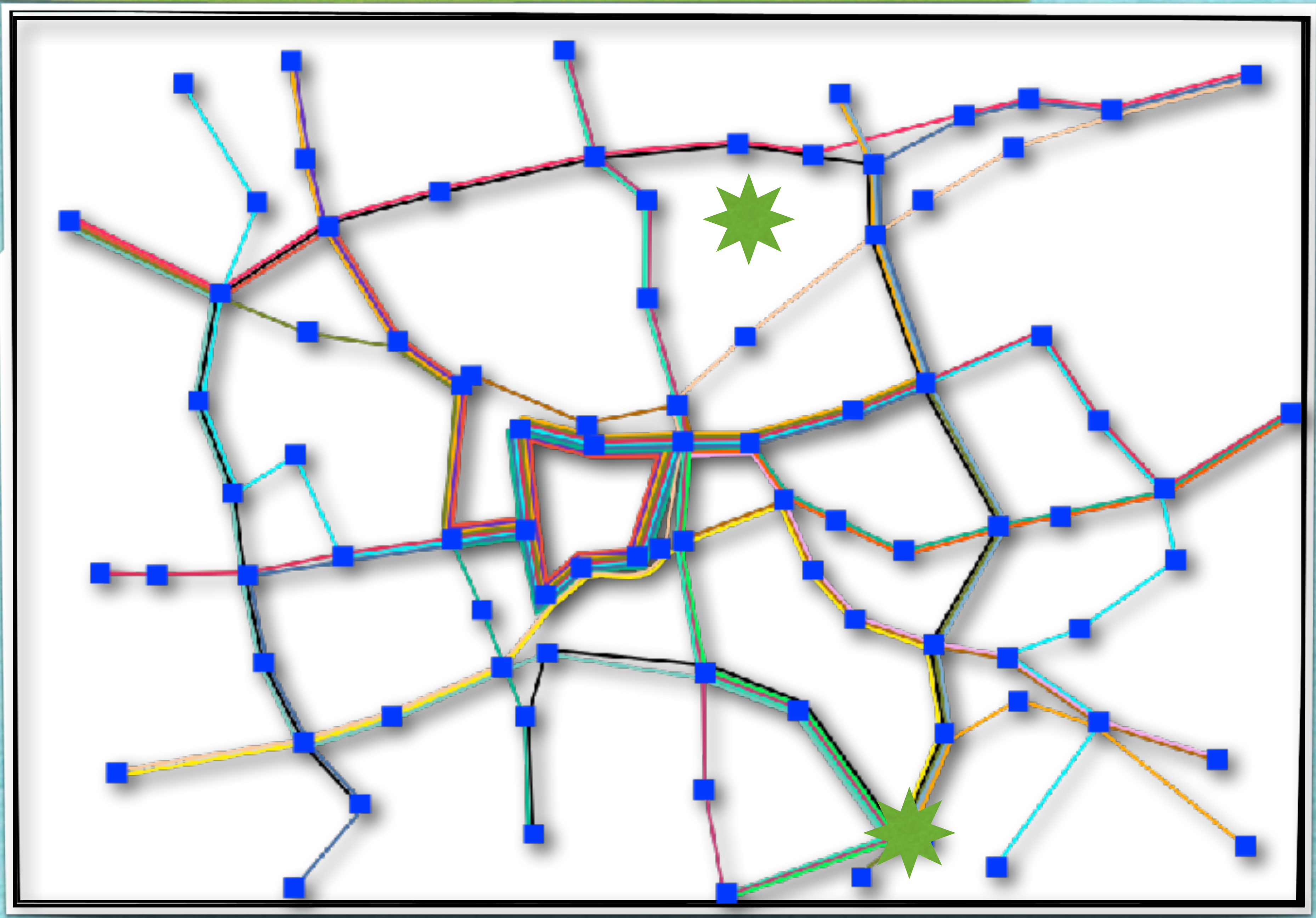
Wellenreiten in Graphen



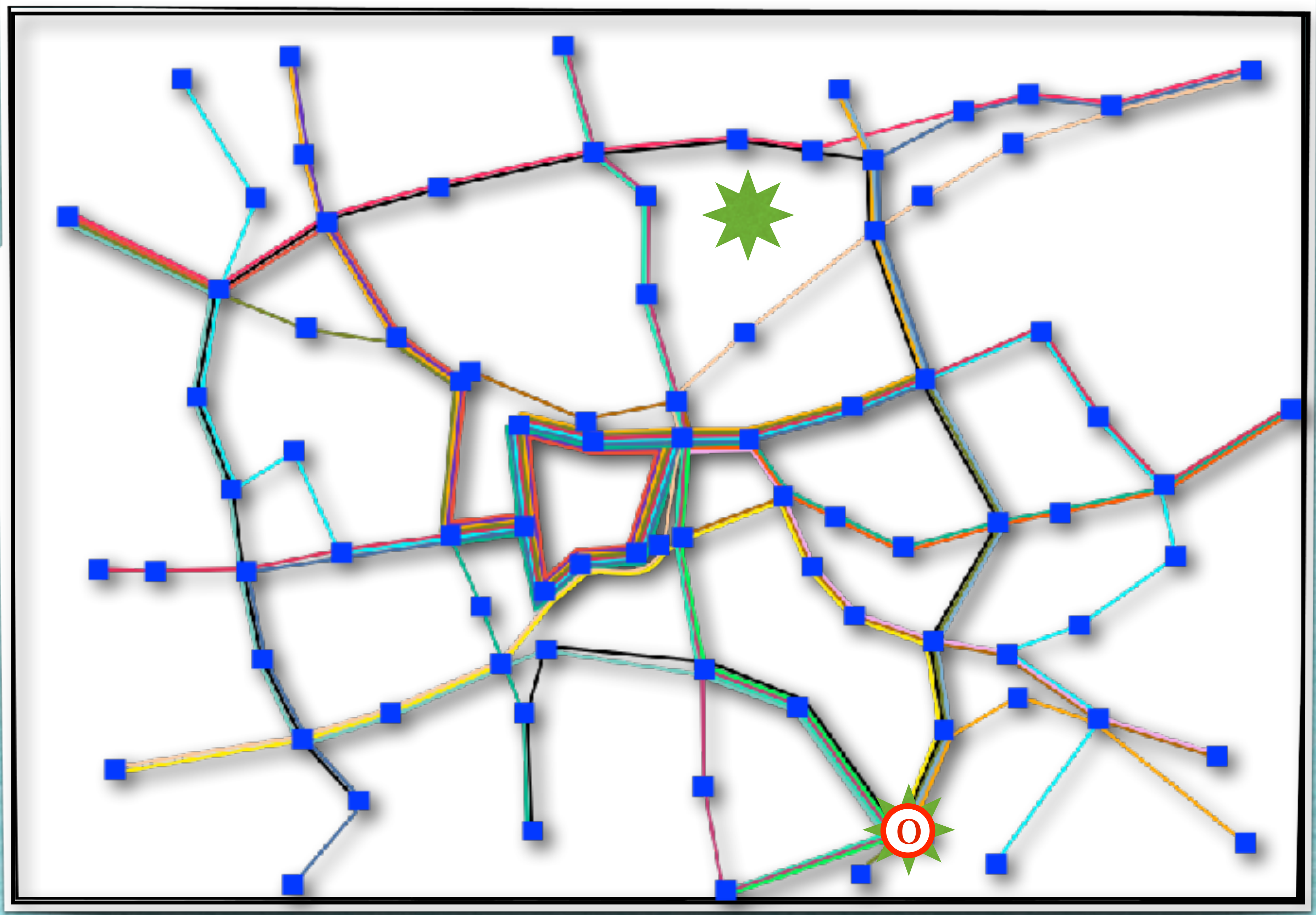
Wellenreiten in Graphen



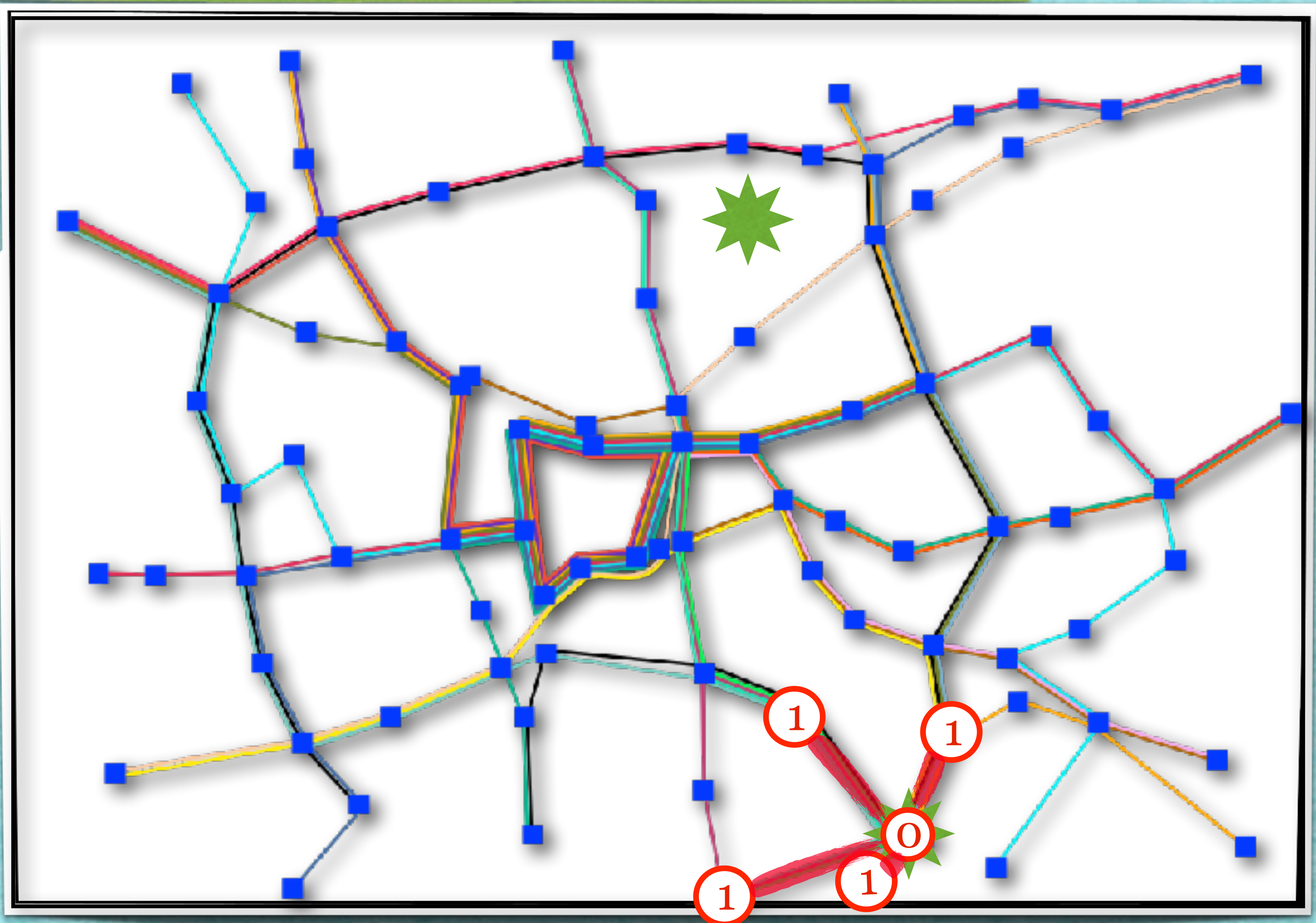
Wellenreiten in Graphen



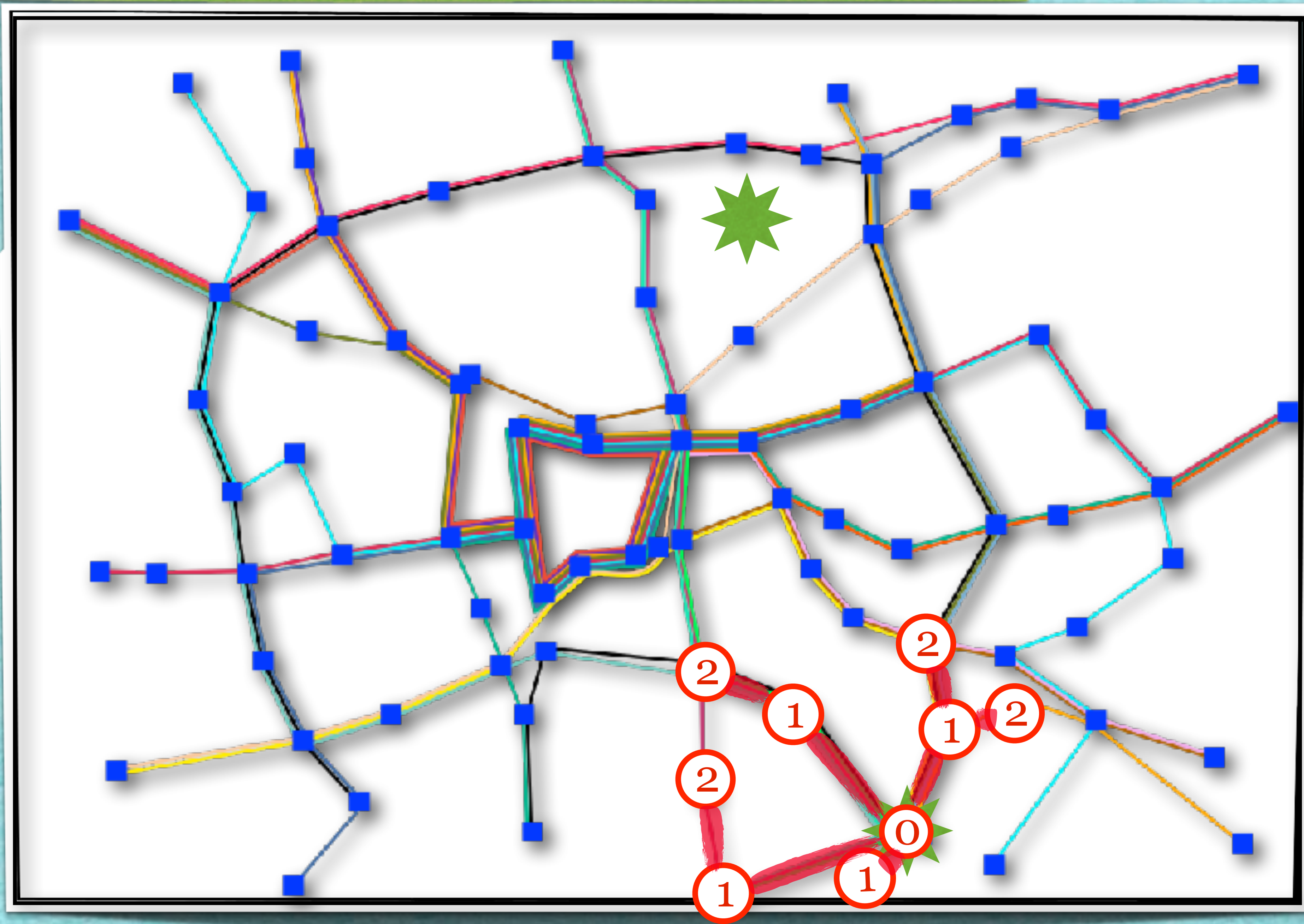
Wellenreiten in Graphen



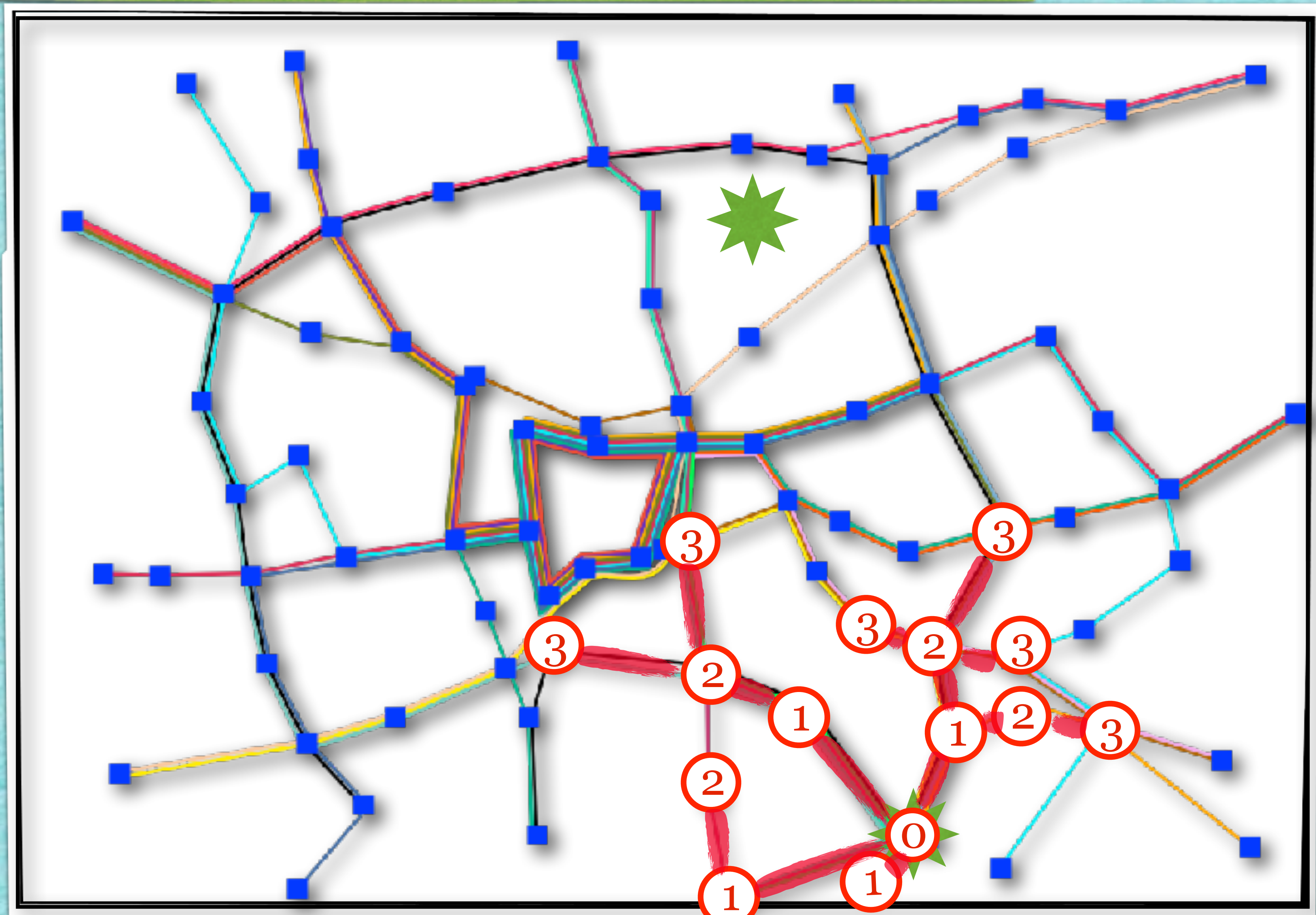
Wellenreiten in Graphen



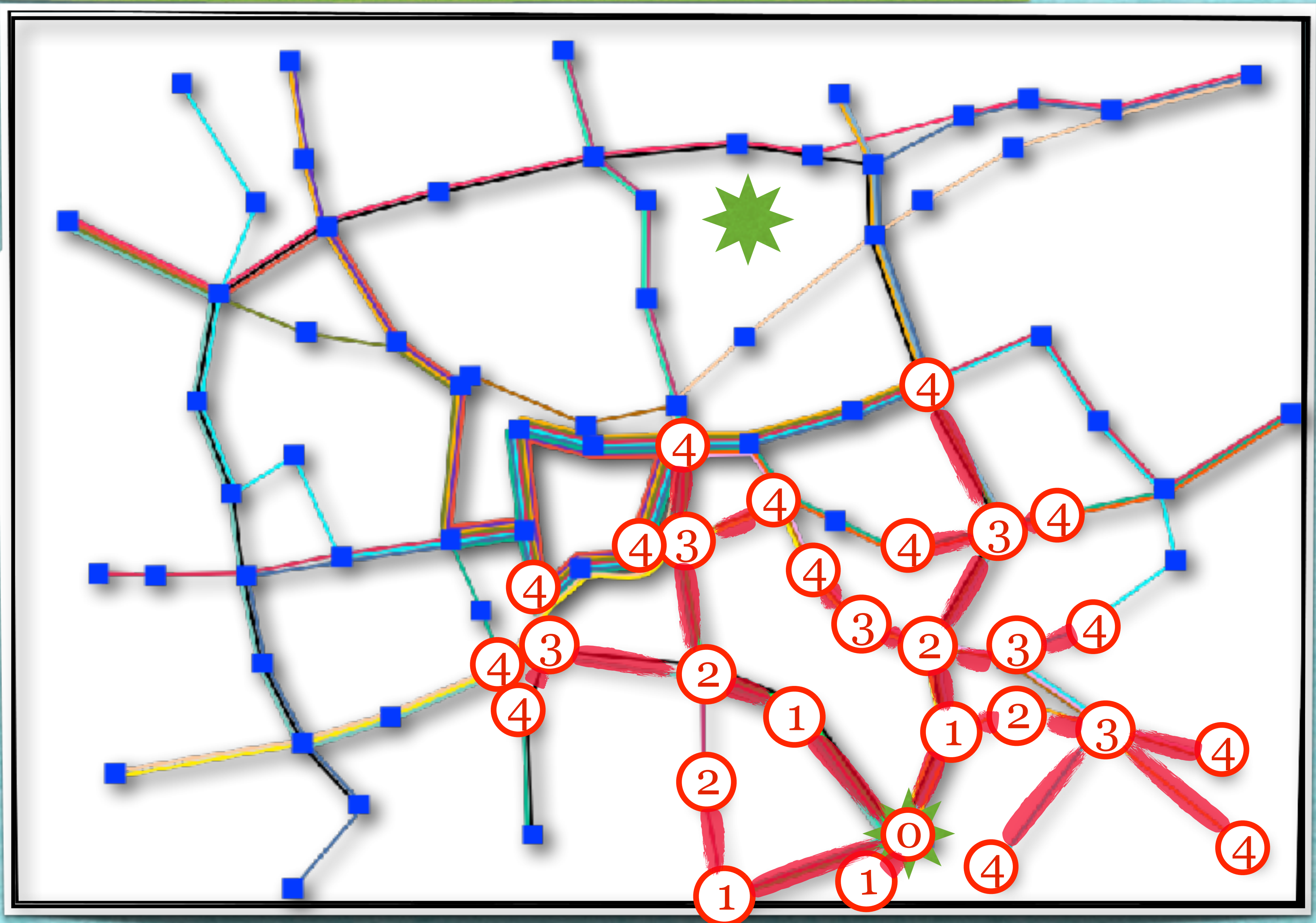
Wellenreiten in Graphen



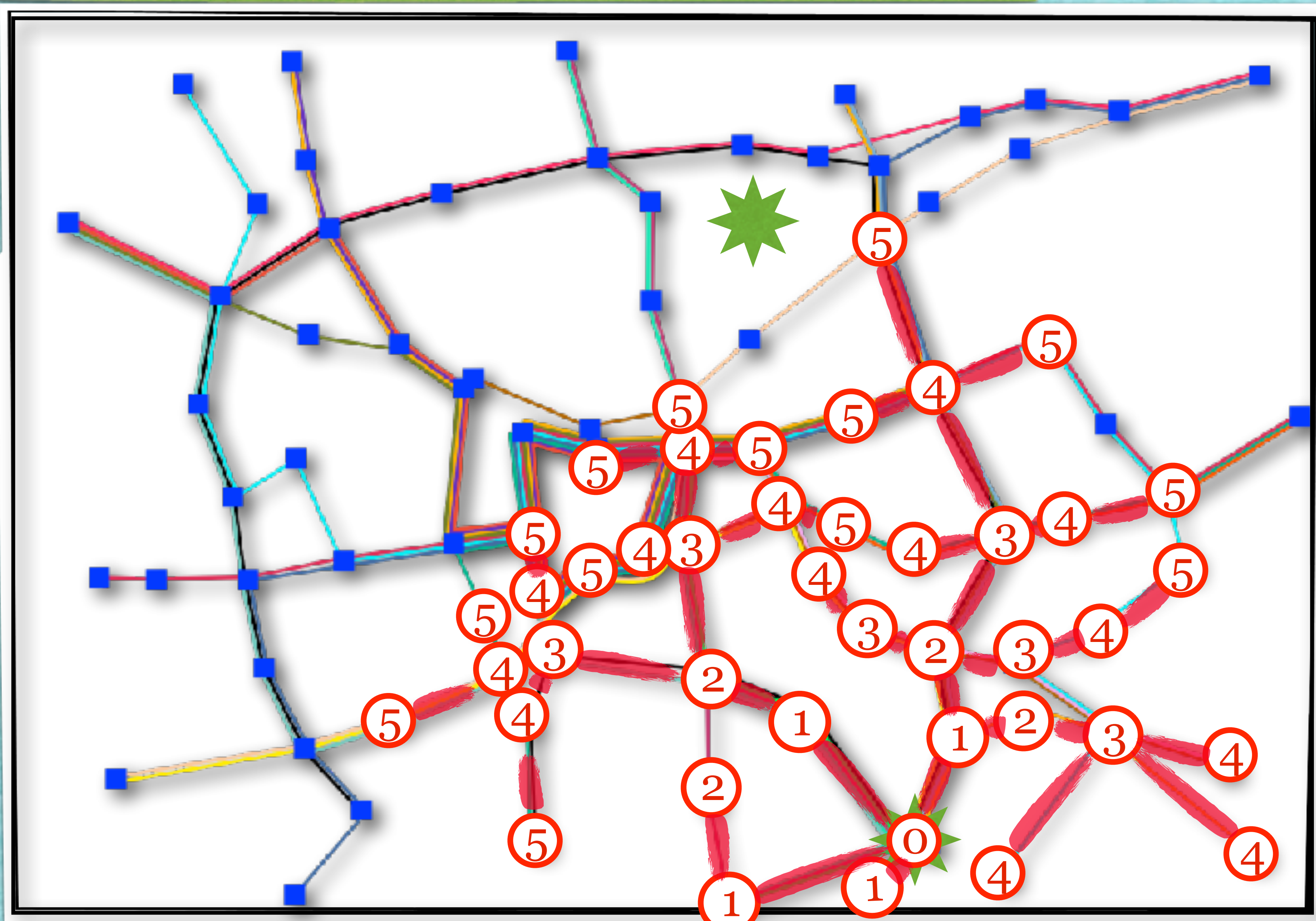
Wellenreiten in Graphen



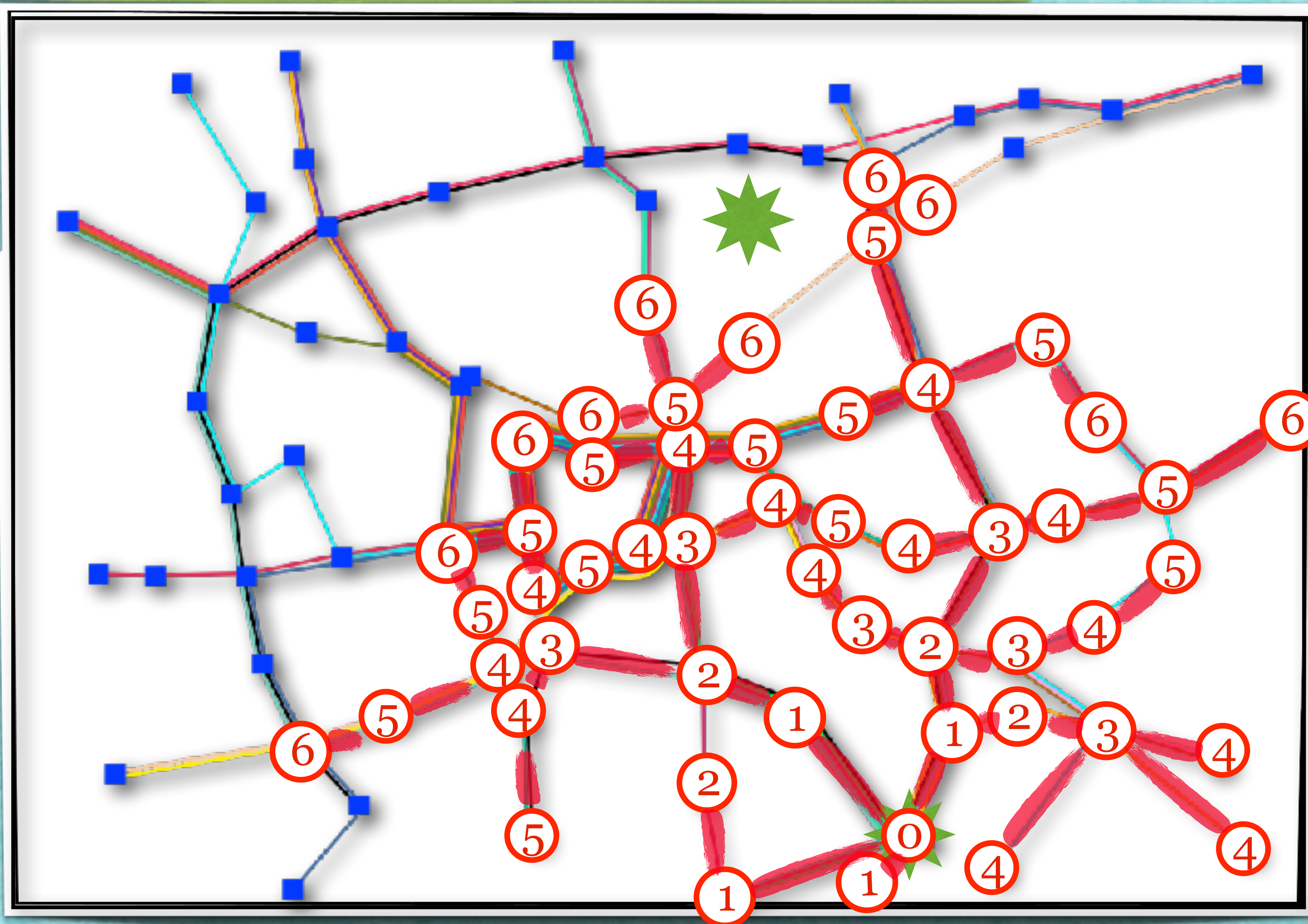
Wellenreiten in Graphen



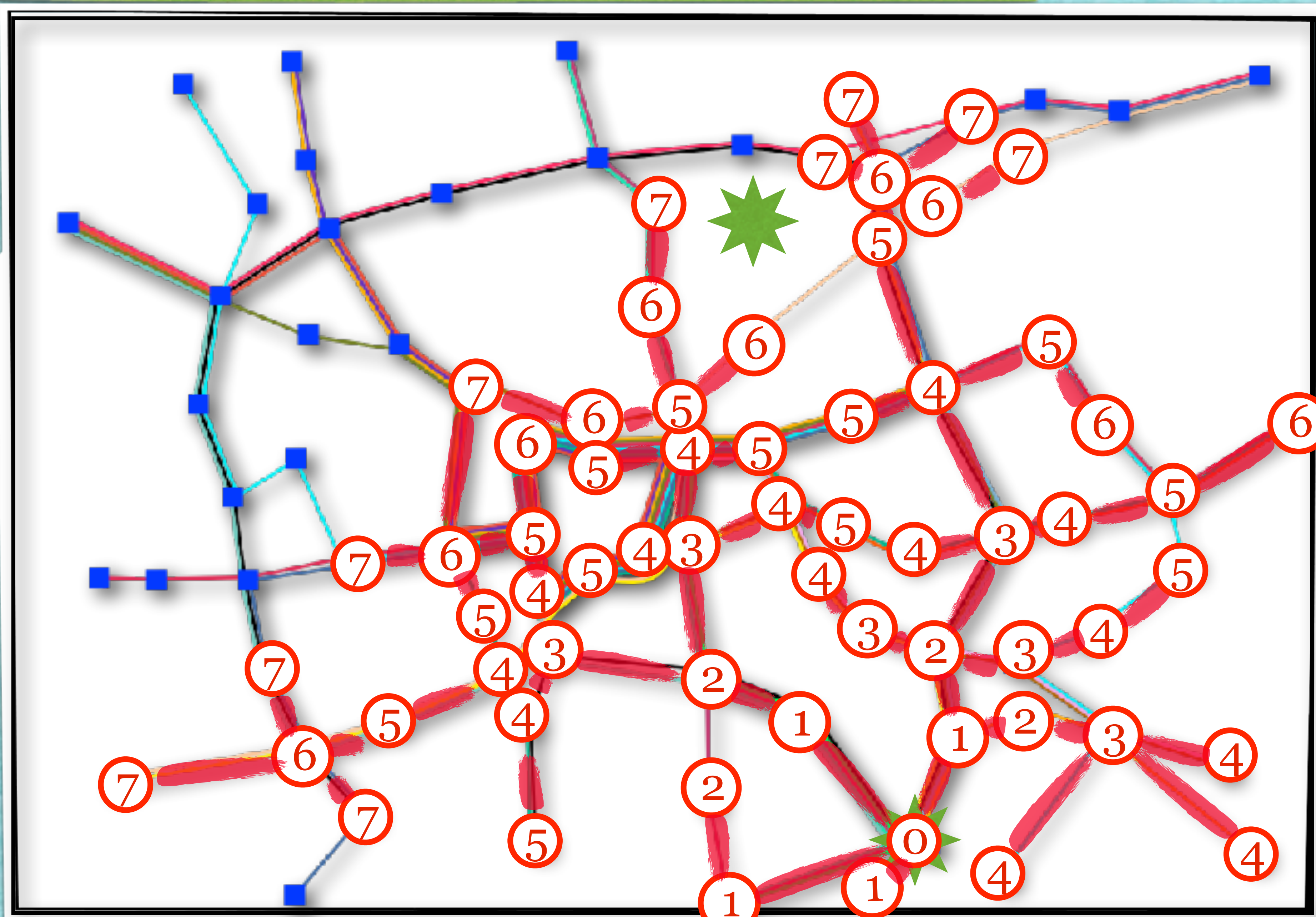
Wellenreiten in Graphen



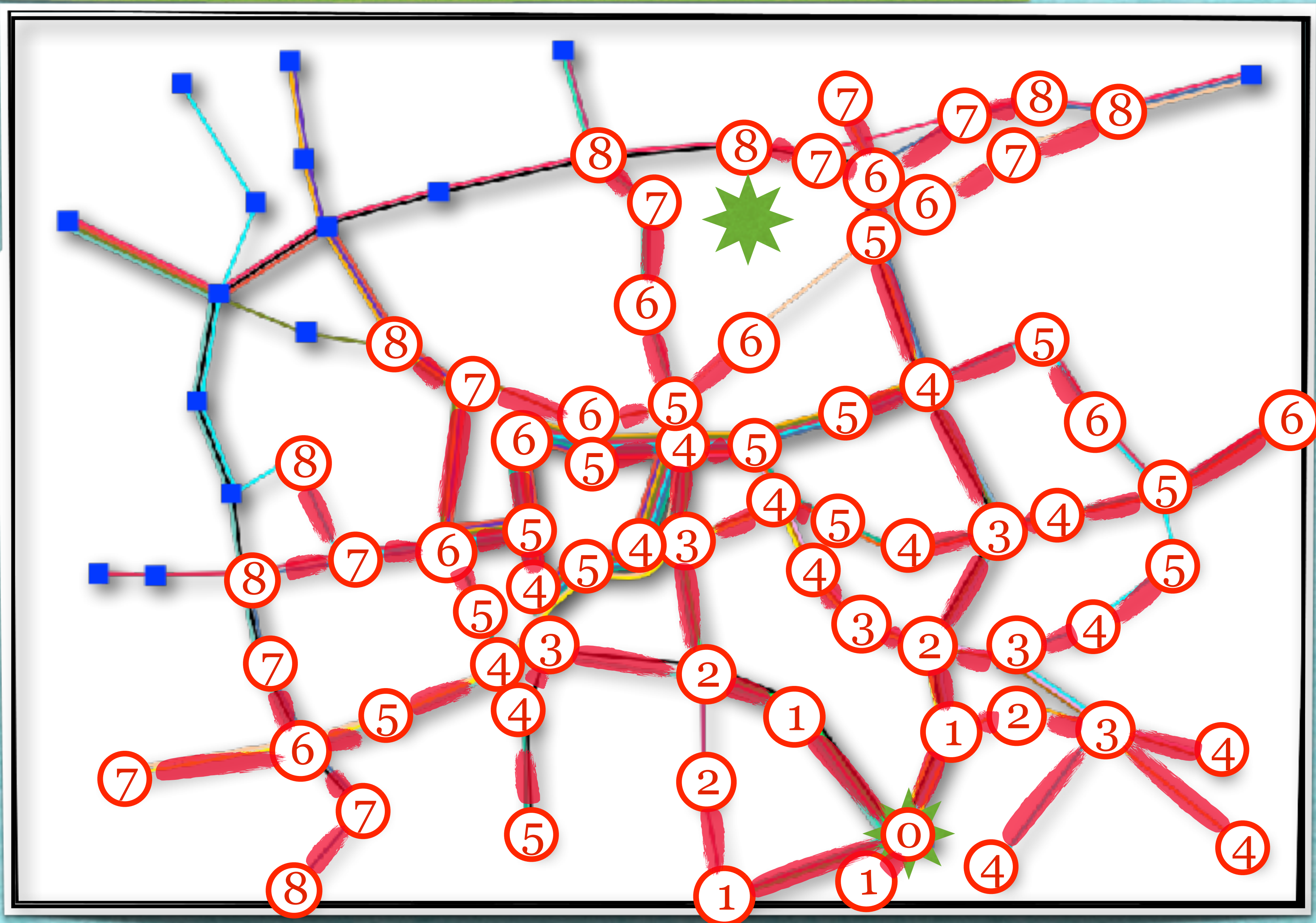
Wellenreiten in Graphen



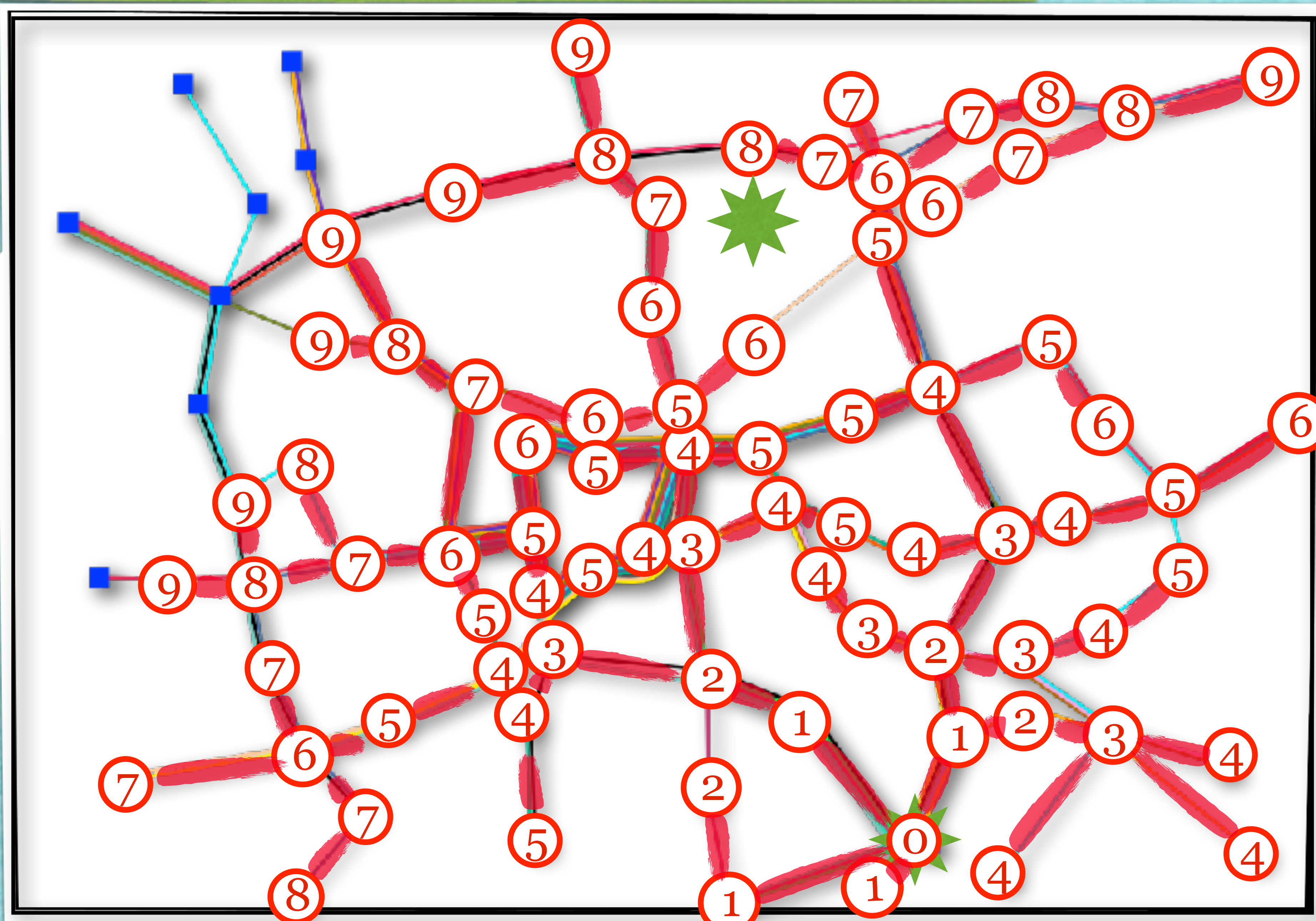
Wellenreiten in Graphen



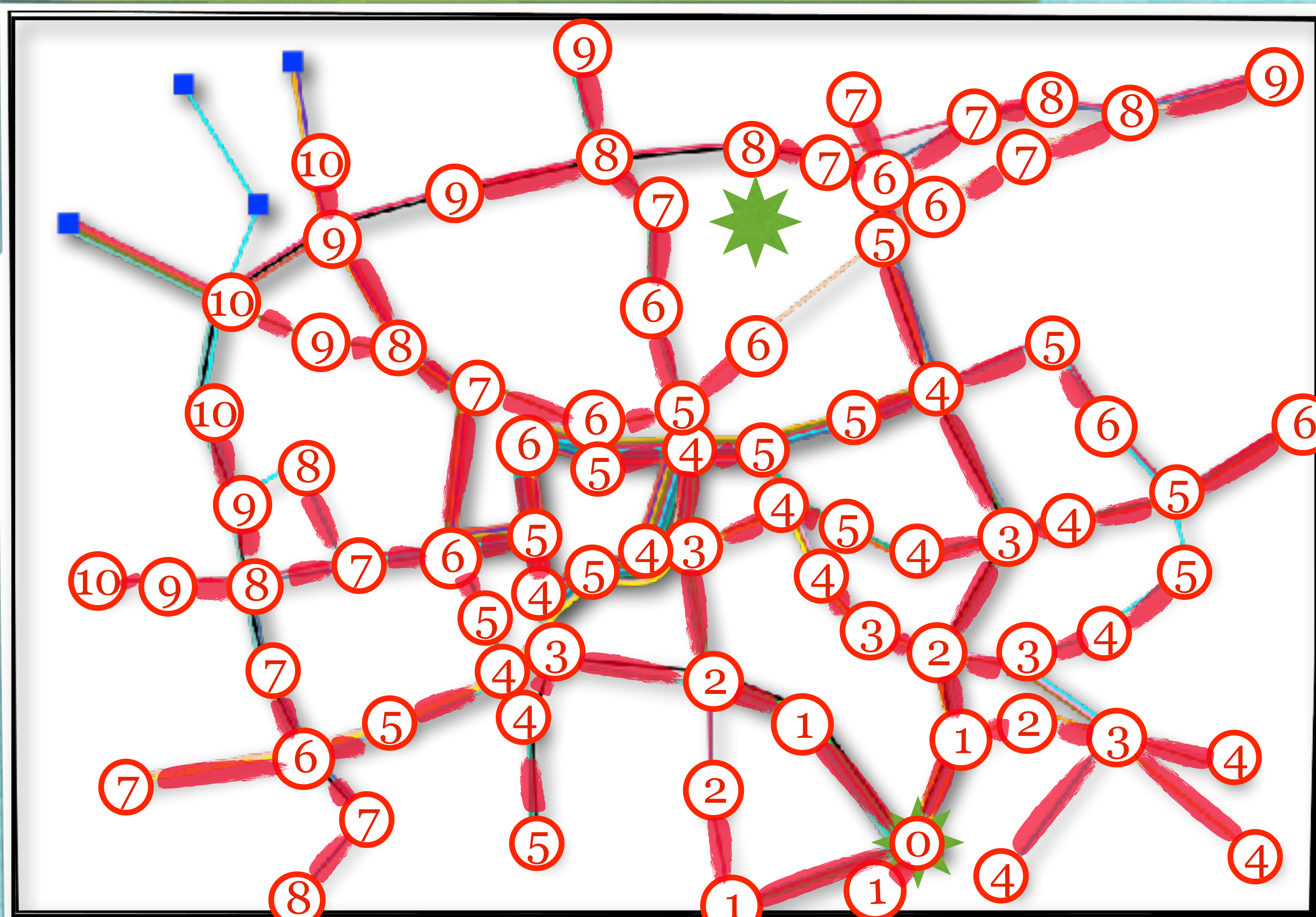
Wellenreiten in Graphen



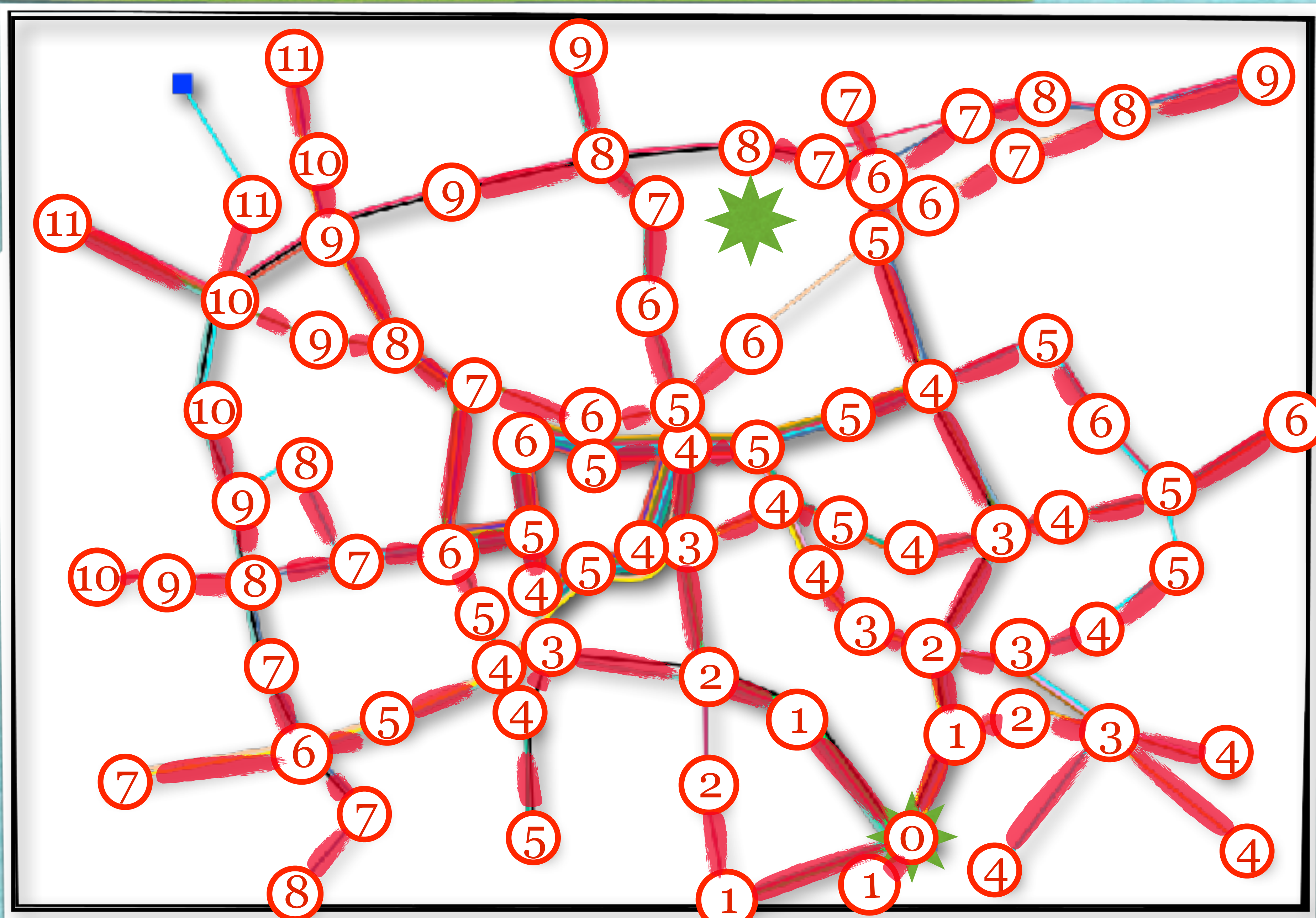
Wellenreiten in Graphen



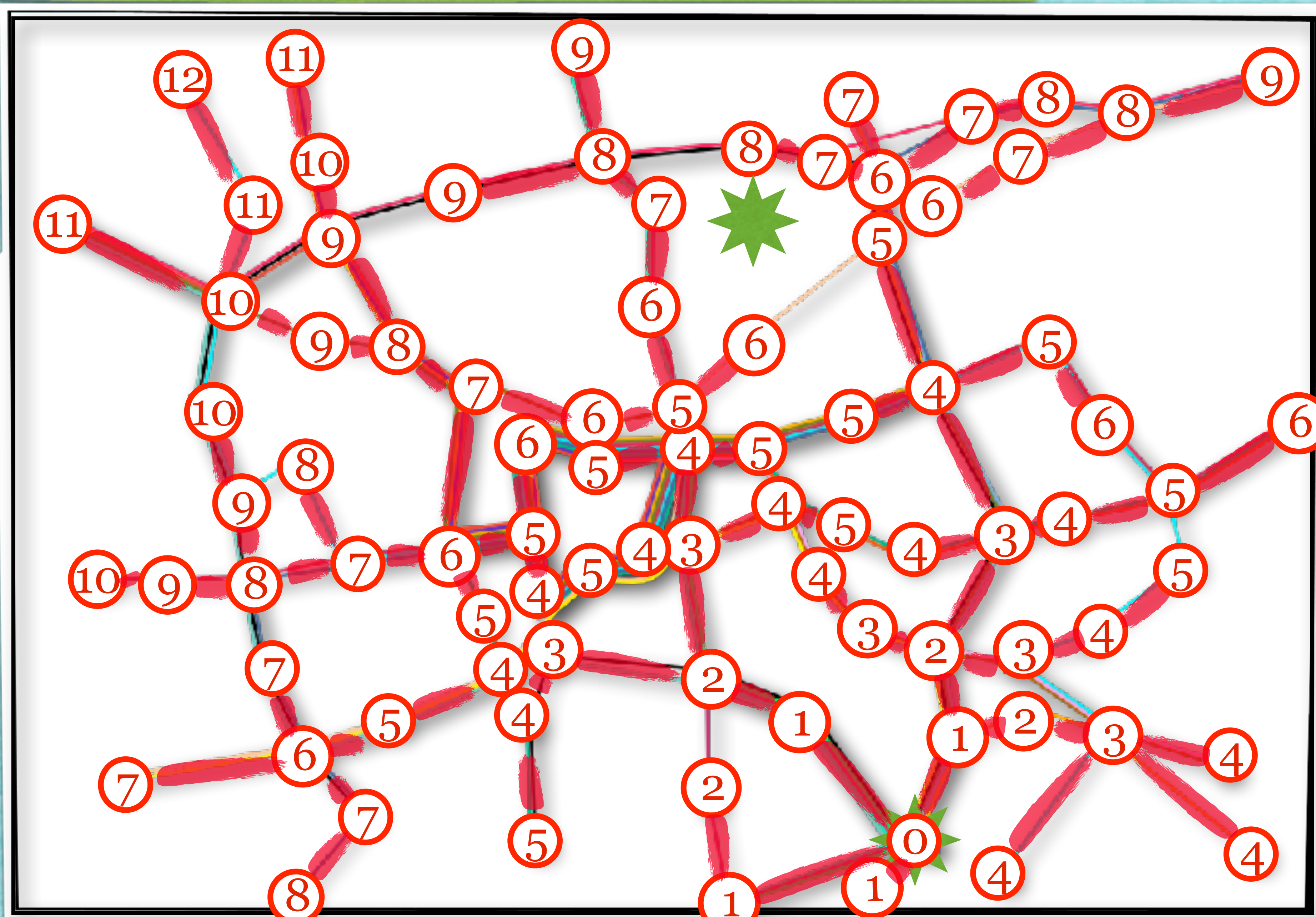
Wellenreiten in Graphen



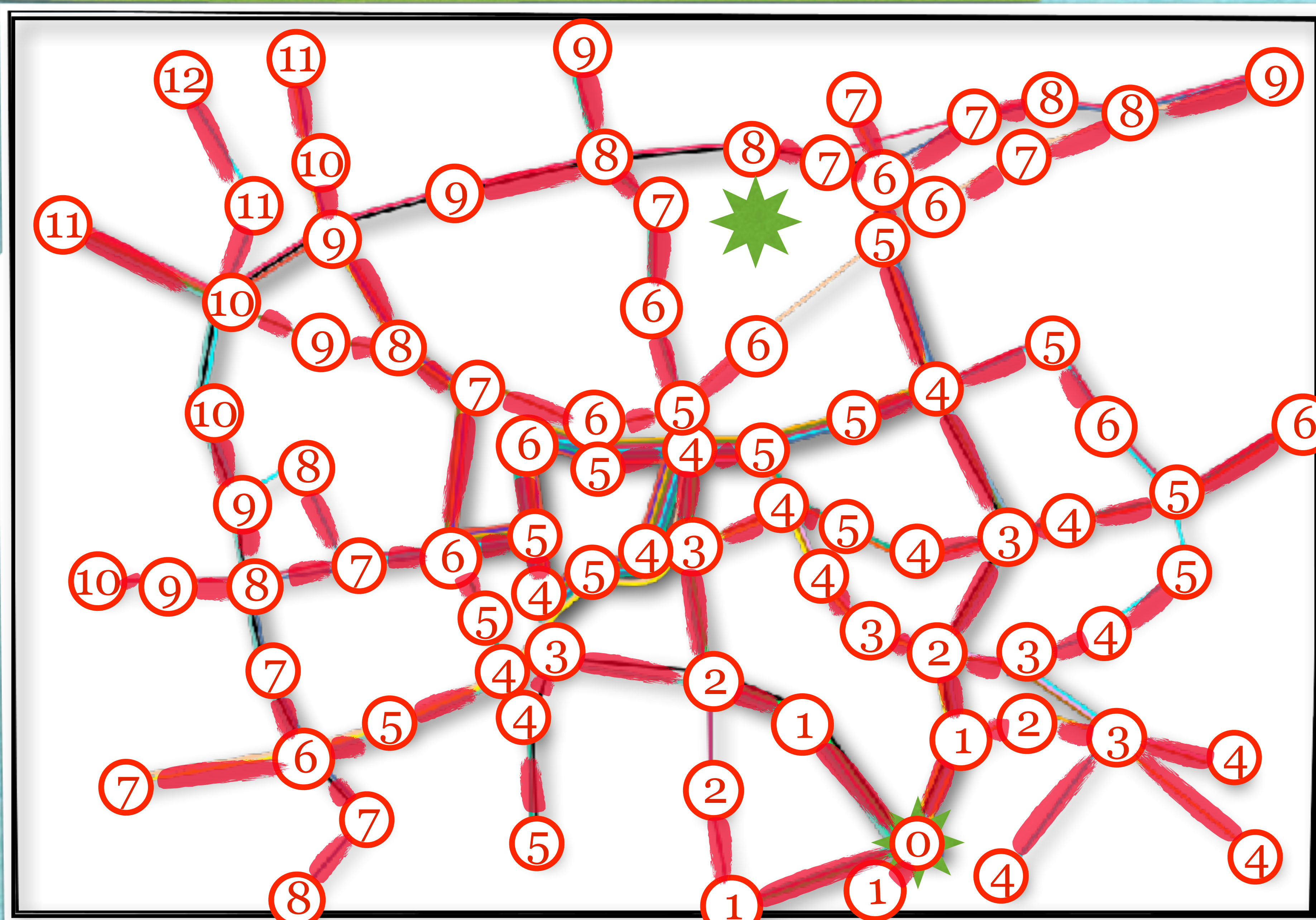
Wellenreiten in Graphen



Wellenreiten in Graphen

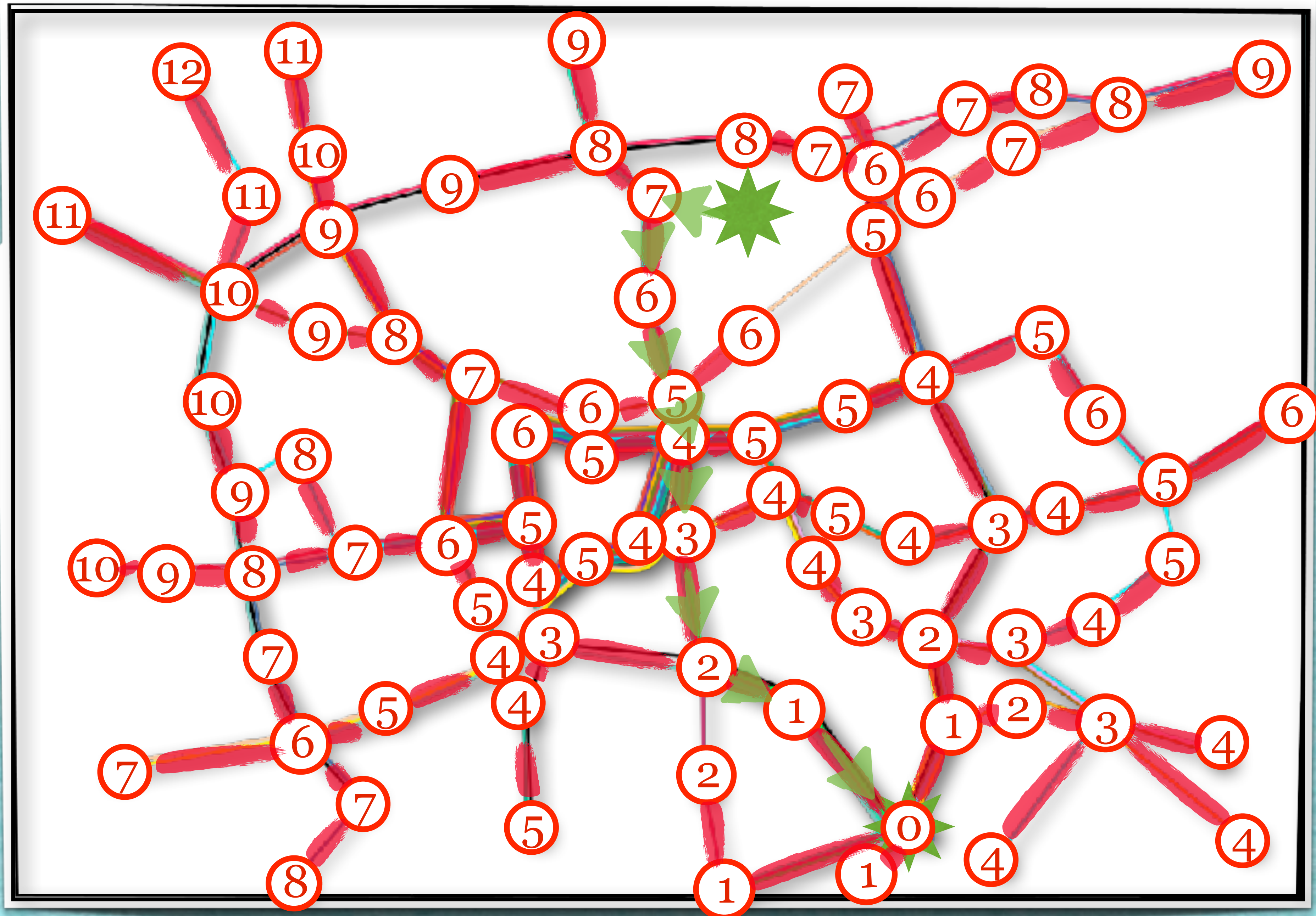


Wellenreiten in Graphen



Breitensuche

Wellenreiten in Graphen



Breitensuche

Algorithmus 3.17

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
    - 2.1. wähle Element $v \in R$
    - 2.2. IF (es gibt kein w $V \backslash Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \backslash \{v\}$
    - 2.3. ELSE {
        - 2.3.1. wähle ein $w \in V \backslash R$ mit $e=\{v,w\} \in E$;
        - 2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

        }

    }

Algorithmus 3.17

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	für jeden Knoten $v \in Y$ die Länge $l(v)$ eines kürzesten s-v-Weges,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
    - 2.1. wähle Element $v \in R$
    - 2.2. IF (es gibt kein w $V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. wähle ein $w \in V \setminus R$ mit $e=\{v,w\} \in E$;
        - 2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

      }

   }

Algorithmus 3.17

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	für jeden Knoten $v \in Y$ die Länge $l(v)$ eines kürzesten s-v-Weges,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

2. WHILE ($R \neq \emptyset$) DO {

   2.1. wähle Element $v \in R$

   2.2. IF (es gibt kein w $V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

      2.2.1. $R:=R \setminus \{v\}$

   2.3. ELSE {

      2.3.1. wähle ein $w \in V \setminus R$ mit $e=\{v,w\} \in E$;

      2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

      }

}

Algorithmus 3.17

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

für jeden Knoten $v \in Y$ die Länge $l(v)$ eines kürzesten s-v-Weges,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

2. WHILE ($R \neq \emptyset$) DO {

    2.1. wähle Element $v \in R$

    2.2. IF (es gibt kein w $V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

        2.2.1. $R:=R \setminus \{v\}$

    2.3. ELSE {

        2.3.1. wähle ein $w \in V \setminus R$ mit $e=\{v,w\} \in E$;

        2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

        2.3.3. setze $l(w):=l(v)+1$

        }

}

Algorithmus 3.17

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	für jeden Knoten $v \in Y$ die Länge $l(v)$ eines kürzesten s-v-Weges,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$

2. WHILE $(R \neq \emptyset)$ DO {

   2.1. wähle Element $v \in R$

   2.2. IF (es gibt kein w $V \setminus Y$ mit $e=\{v,w\} \in E$) THEN

      2.2.1. $R:=R \setminus \{v\}$

   2.3. ELSE {

      2.3.1. wähle ein $w \in V \setminus R$ mit $e=\{v,w\} \in E$;

      2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;

      2.3.3. setze $l(w):=l(v)+1$

      }

}

3.9 BFS

Satz 3.18

Satz 3.18

(1) Das Verfahren 3.17 ist endlich.

Satz 3.18

(1) Das Verfahren 3.17 ist endlich.

(2) Die Laufzeit ist $O(n+m)$.

Satz 3.18

(1) *Das Verfahren 3.17 ist endlich.*

(2) *Die Laufzeit ist O(n+m).*

(3) *Am Ende ist für jeden erreichbaren Knoten v ∈ Y die Länge eines kürzesten Weges von s nach v im Baum (Y,T) durch l(v) gegeben.*

Satz 3.18

(1) *Das Verfahren 3.17 ist endlich.*

(2) *Die Laufzeit ist O(n+m).*

(3) *Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v im Baum (Y,T) durch l(v) gegeben.*

(4) *Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v im Graphen (V,E) durch l(v) gegeben.*

Satz 3.18

(1) *Das Verfahren 3.17 ist endlich.*

(2) *Die Laufzeit ist O(n+m).*

(3) *Am Ende ist für jeden erreichbaren Knoten v $\in$ Y die Länge eines kürzesten Weges von s nach v im Baum (Y,T) durch l(v) gegeben.*

(4) *Am Ende ist für jeden erreichbaren Knoten v $\in$ Y die Länge eines kürzesten Weges von s nach v im Graphen (V,E) durch l(v) gegeben.*

Beweis:

Satz 3.18

(1) Das Verfahren 3.17 ist endlich.

(2) Die Laufzeit ist $O(n+m)$.

(3) Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v im Baum (Y,T) durch $l(v)$ gegeben.

(4) Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v im Graphen (V,E) durch $l(v)$ gegeben.

Beweis:

(1) Wie für Algorithmus 3.7 gelten alle Eigenschaften.
zusätzlich ist für jeden Knoten $v \in Y$ per Induktion,
der Wert $l(v)$ tatsächlich definiert.

Satz 3.18

(1) Das Verfahren 3.17 ist endlich.

(2) Die Laufzeit ist O(n+m).

(3) Am Ende ist für jeden erreichbaren Knoten v ∈ Y die Länge eines kürzesten Weges von s nach v im Baum (Y,T) durch l(v) gegeben.

(4) Am Ende ist für jeden erreichbaren Knoten v ∈ Y die Länge eines kürzesten Weges von s nach v im Graphen (V,E) durch l(v) gegeben.

Beweis:

(1) Wie für Algorithmus 3.7 gelten alle Eigenschaften. zusätzlich ist für jeden Knoten v ∈ Y per Induktion, der Wert l(v) tatsächlich definiert.

(2) Die Laufzeit bleibt von Algorithmus 3.7 erhalten.

Mehr Details!

s.fekete@tu-bs.de

GRÅPH SCÄN

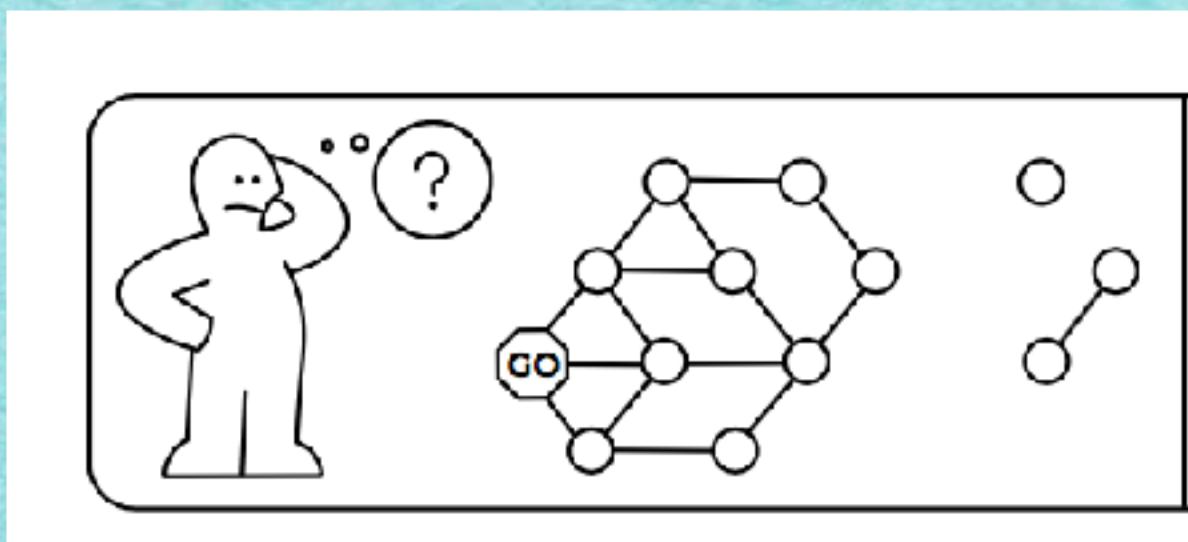
GRÅPH SCÄN

idea-instructions.com/graph-scan/

IDEA

GRÅPH SCÄN

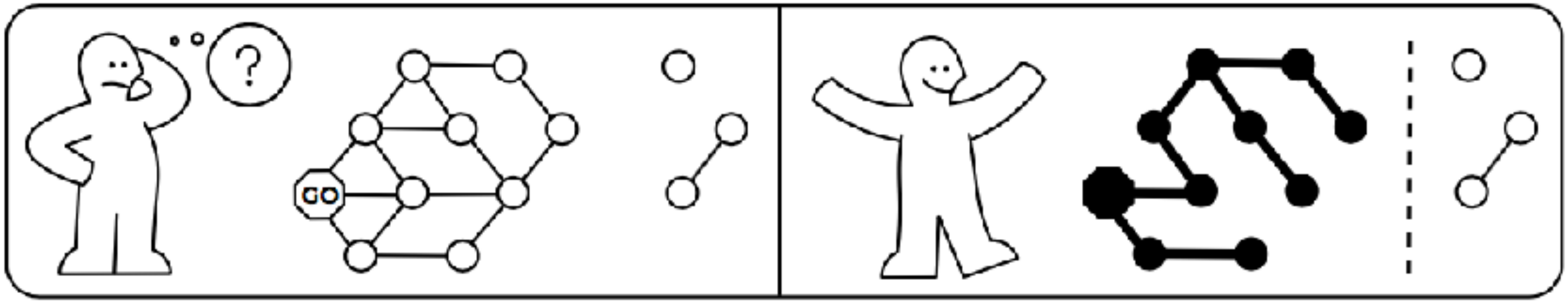
IDEA



GRÅPH SCÄN

idea-instructions.com/graph-scan/

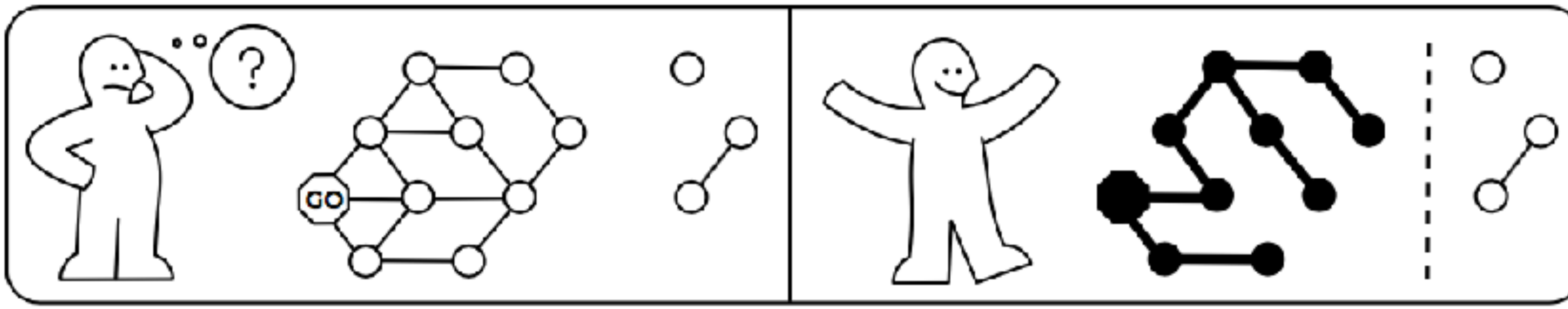
IDEA

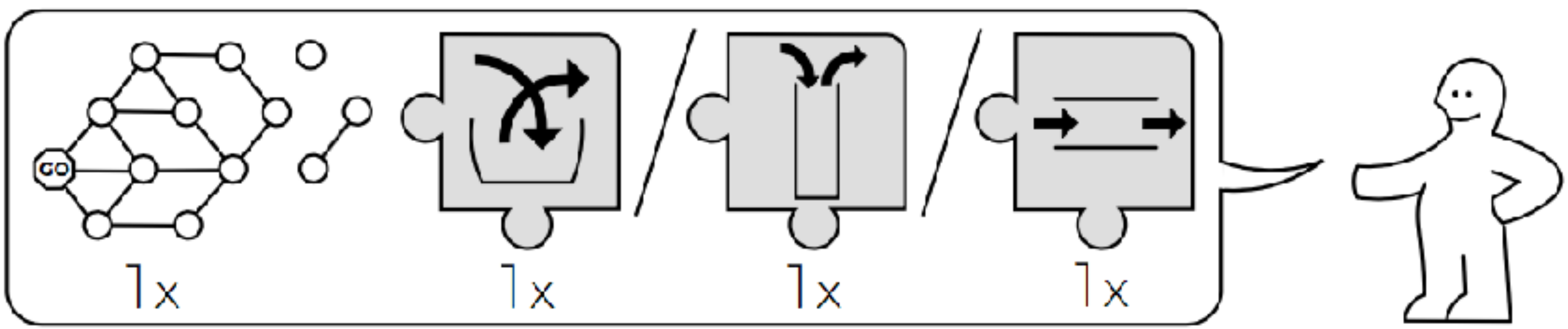


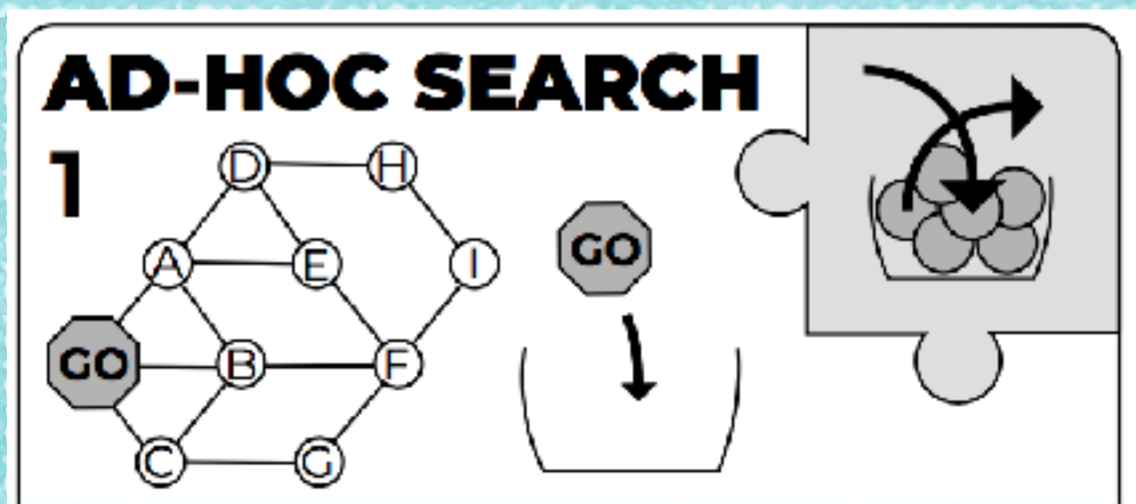
GRÅPH SCÄN

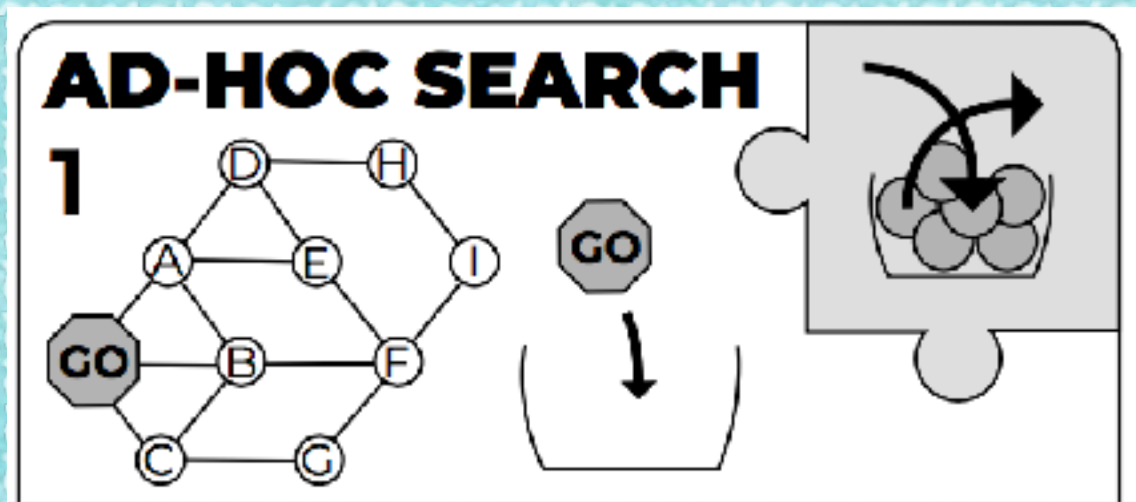
idea-instructions.com/graph-scan/

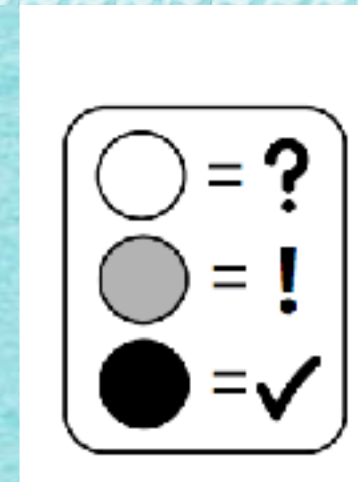
IDEA



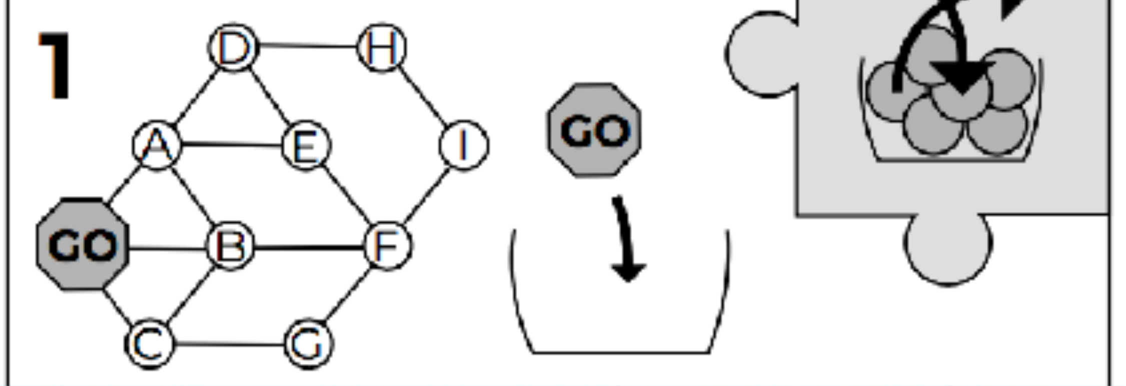


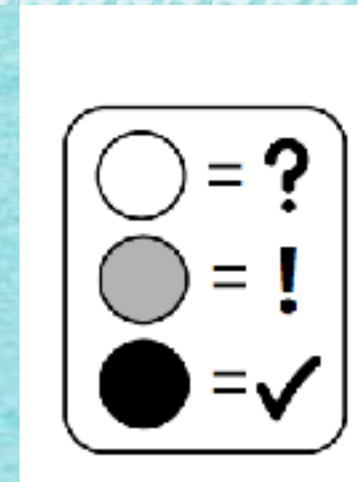

AD-HOC SEARCH
1
D
H
A
E
I
GO
B
F
C
G
GO

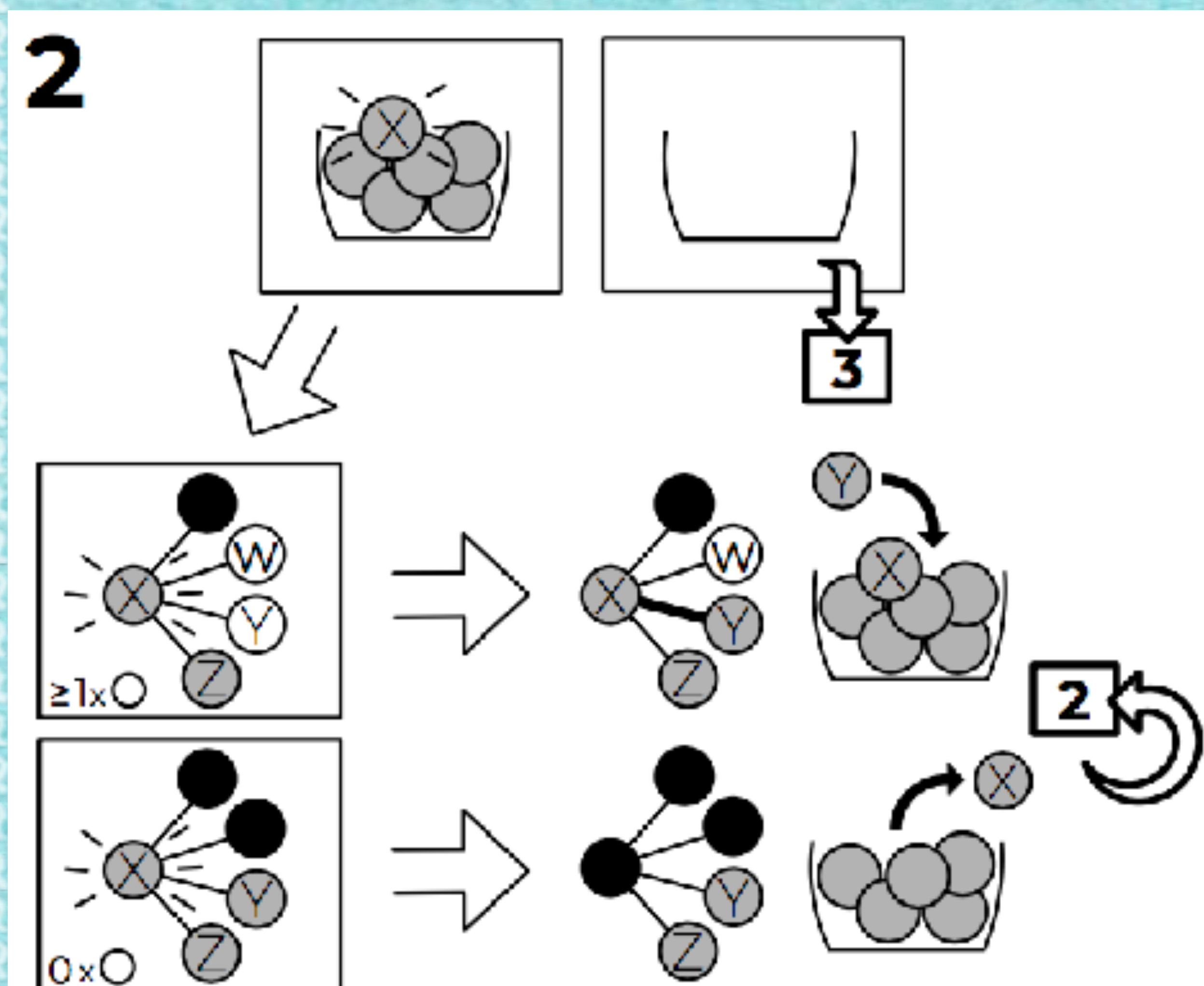

AD-HOC SEARCH
1
D
H
A
E
I
GO
B
F
C
G
GO


= ?
= !
=✓

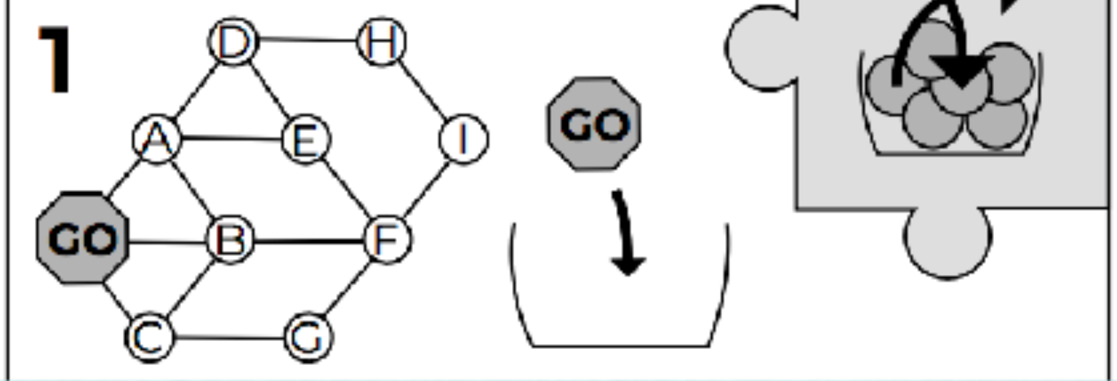
AD-HOC SEARCH

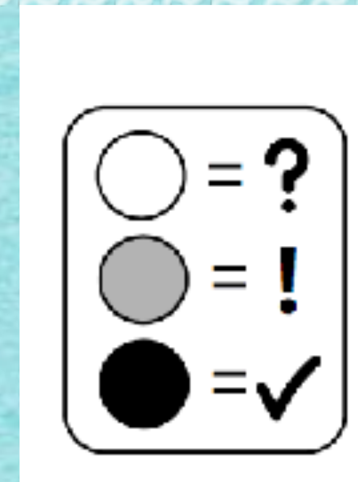


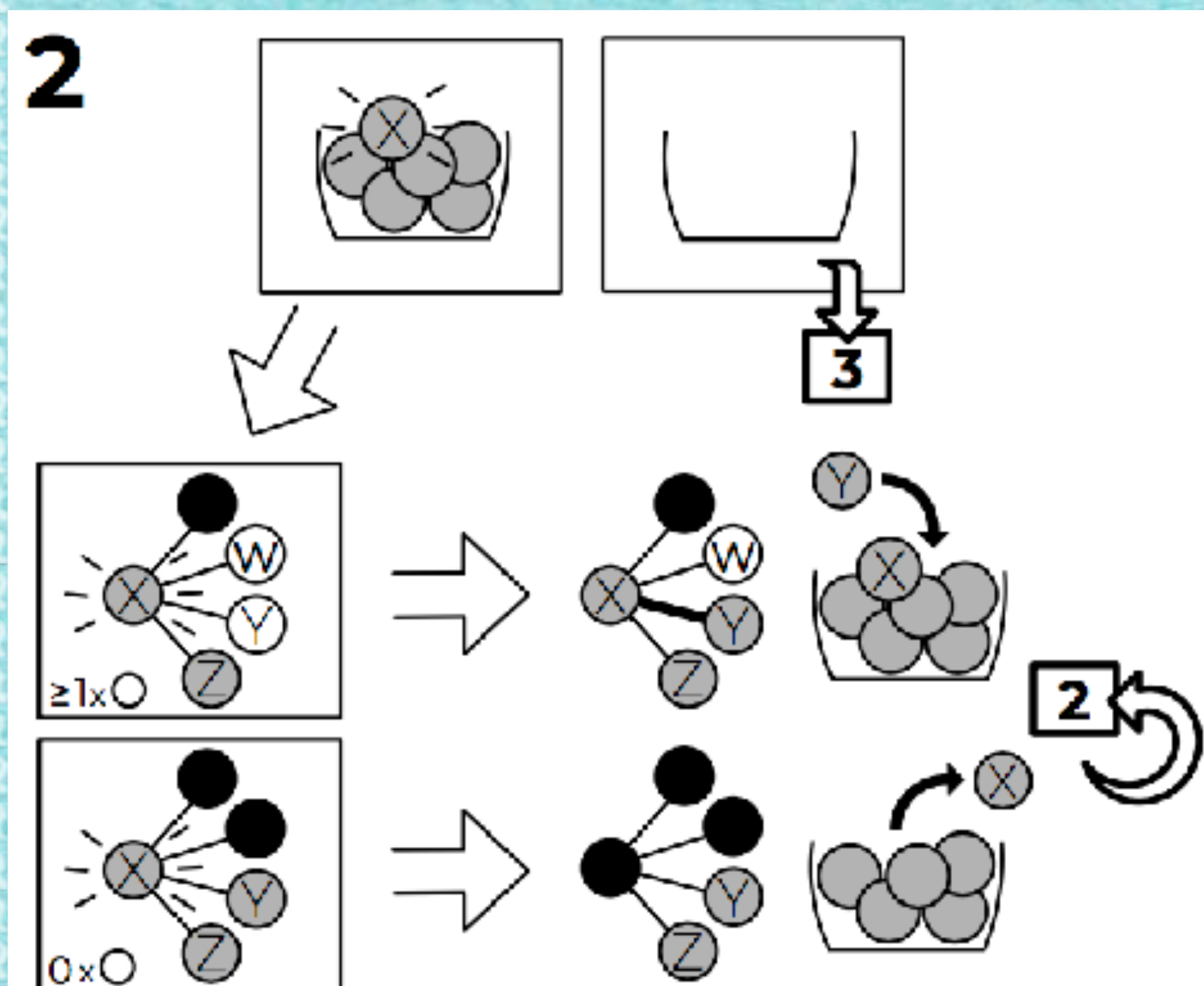


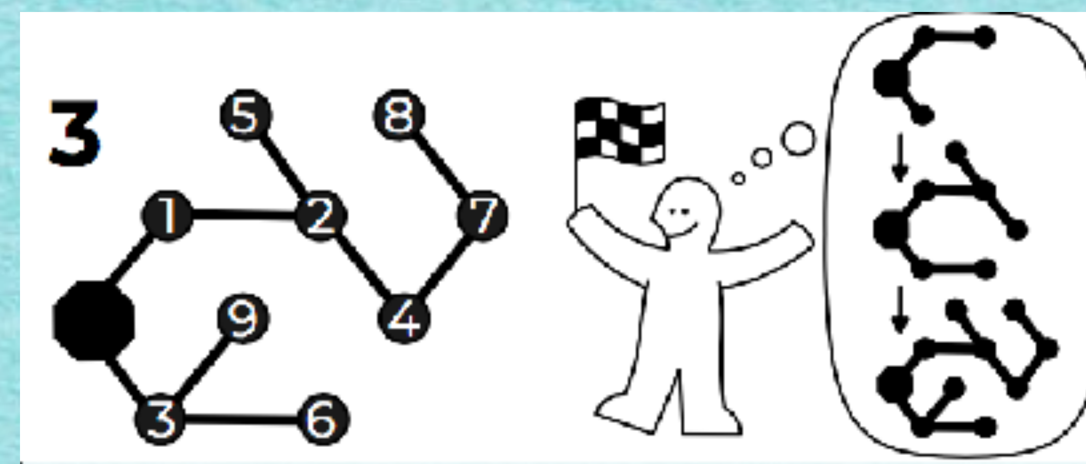


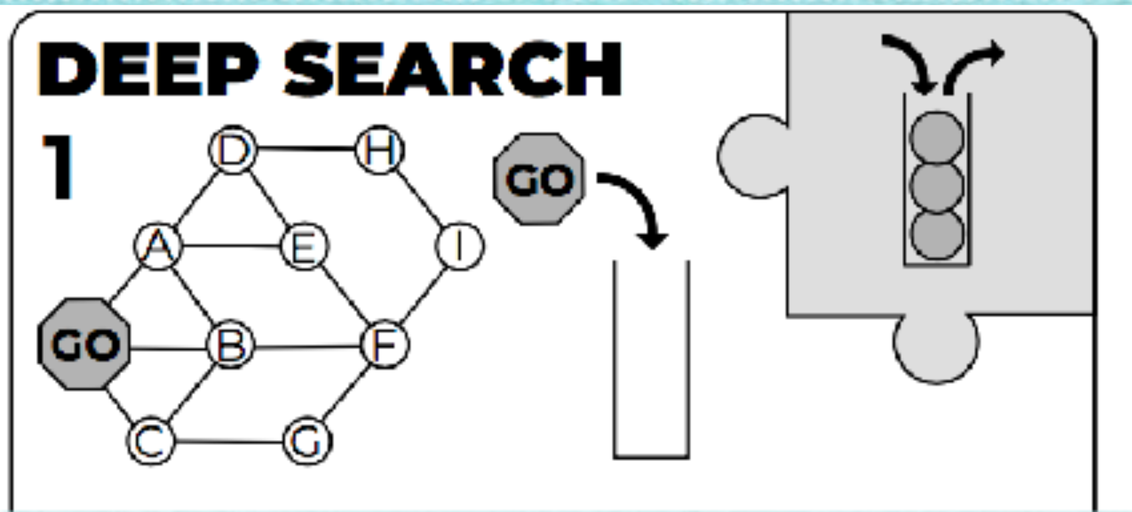
AD-HOC SEARCH





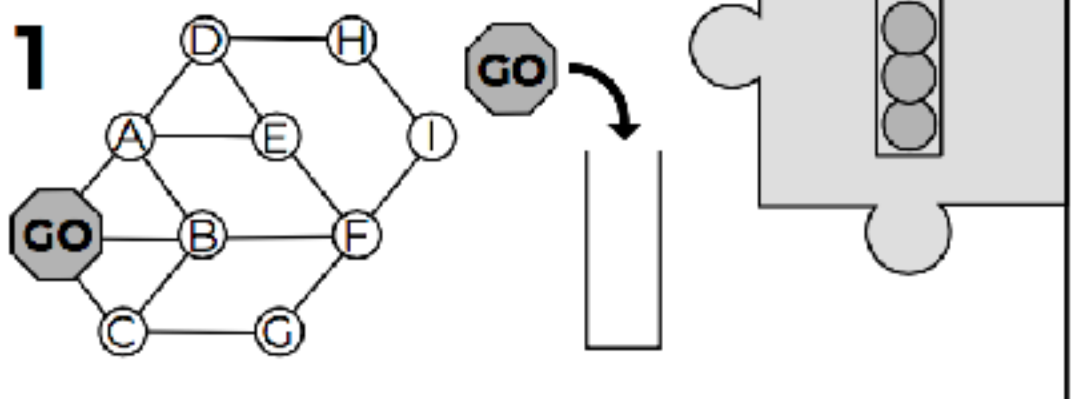


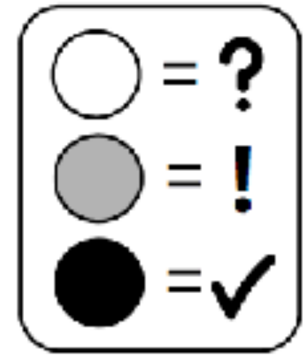



DEEP SEARCH
1
GO
A
B
C
D
E
F
G
H
I
GO

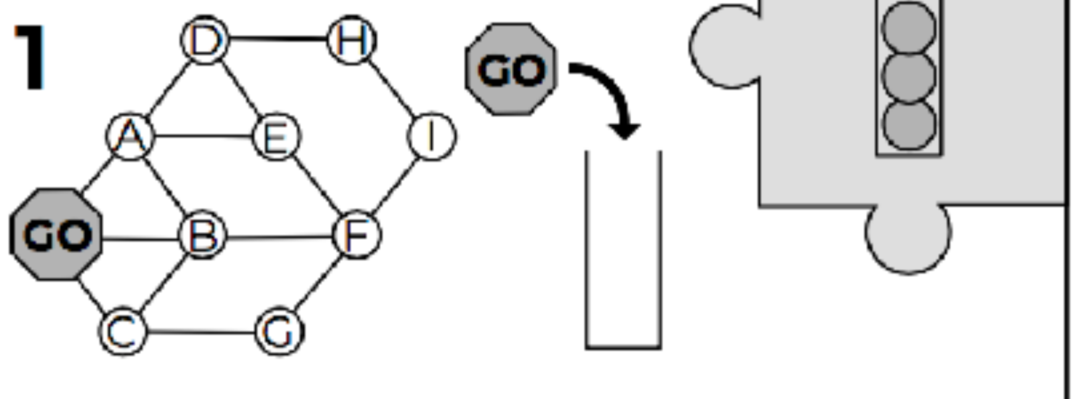
DEEP SEARCH

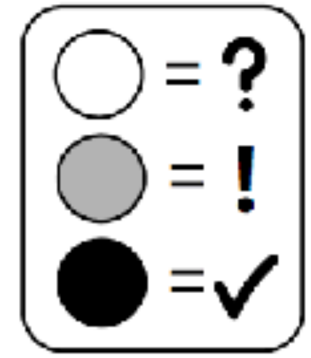
1

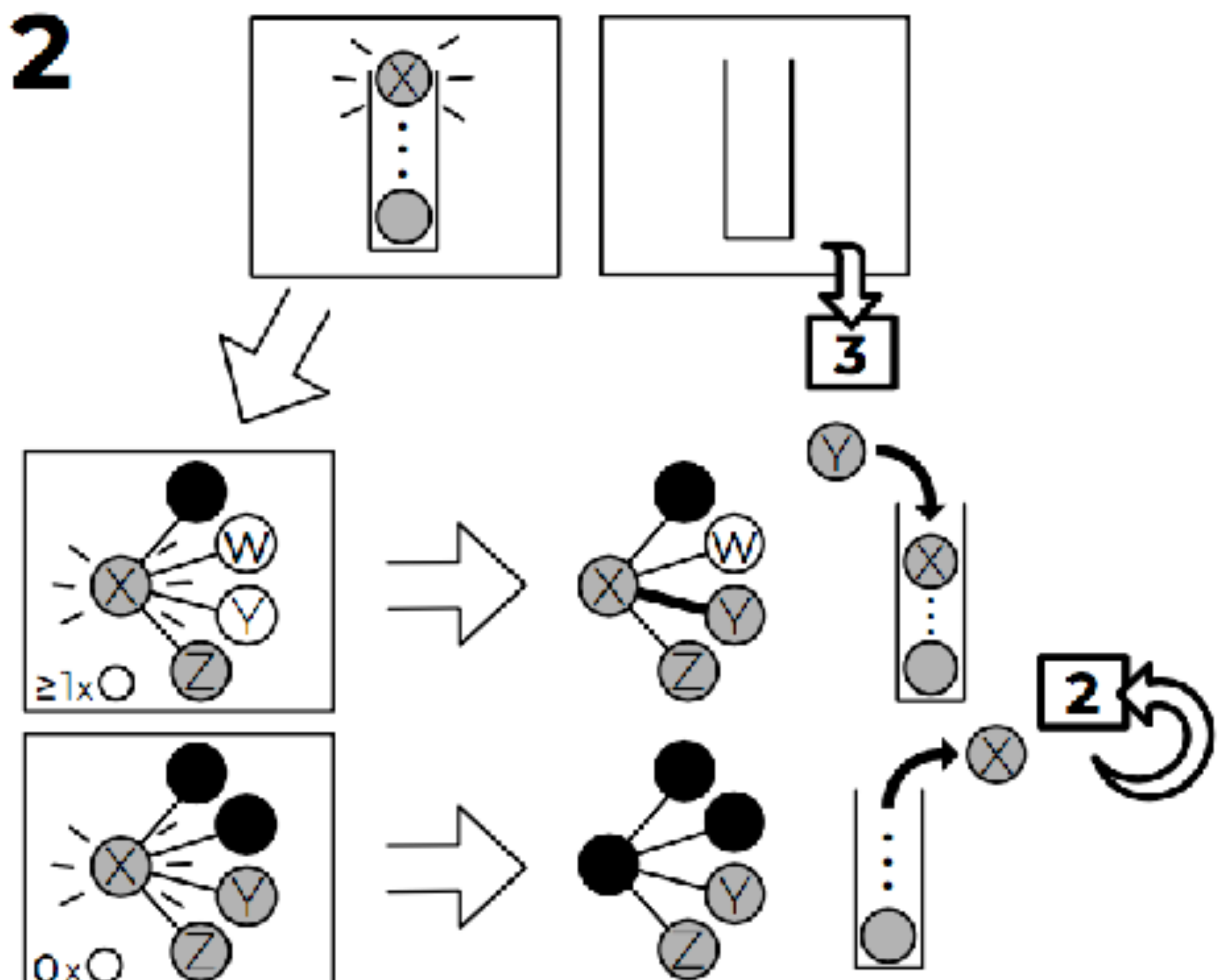




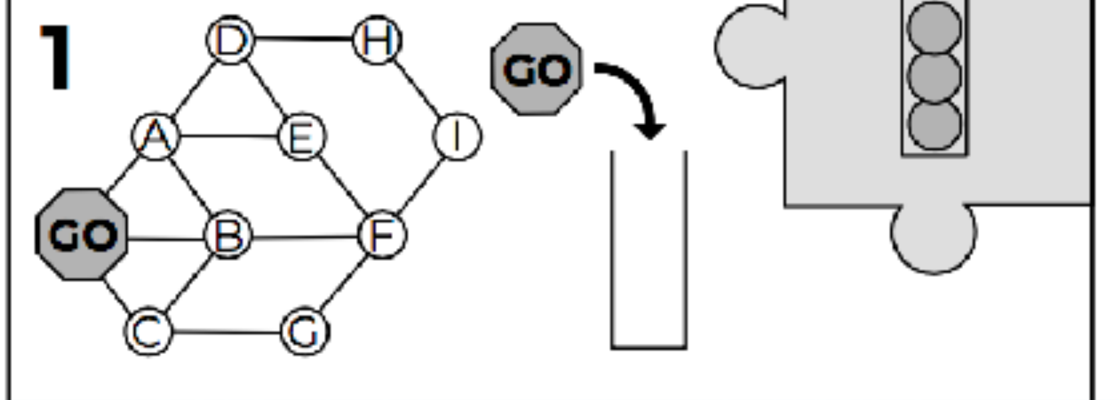
DEEP SEARCH

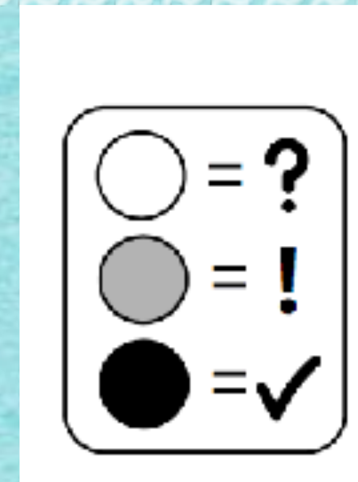


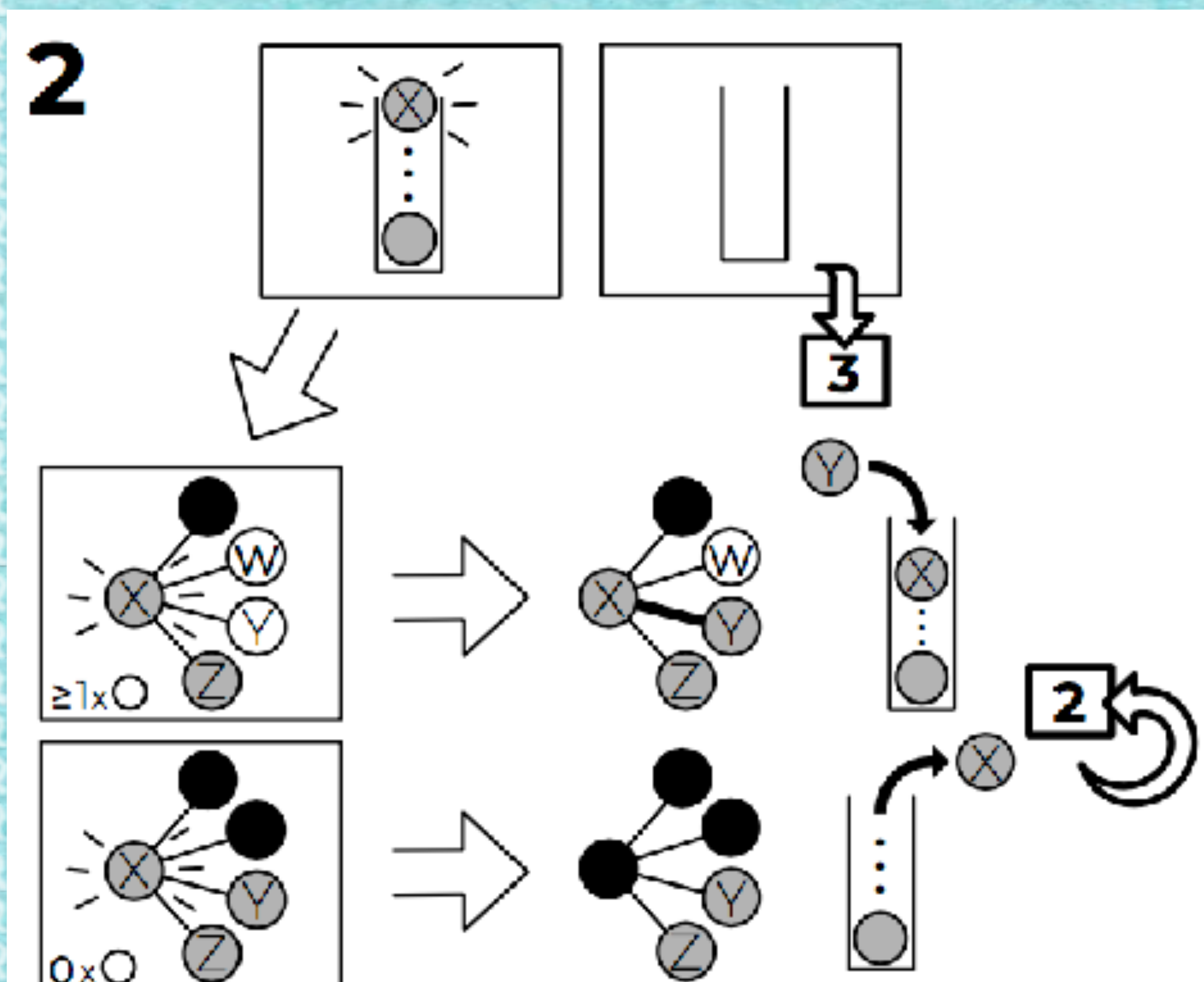


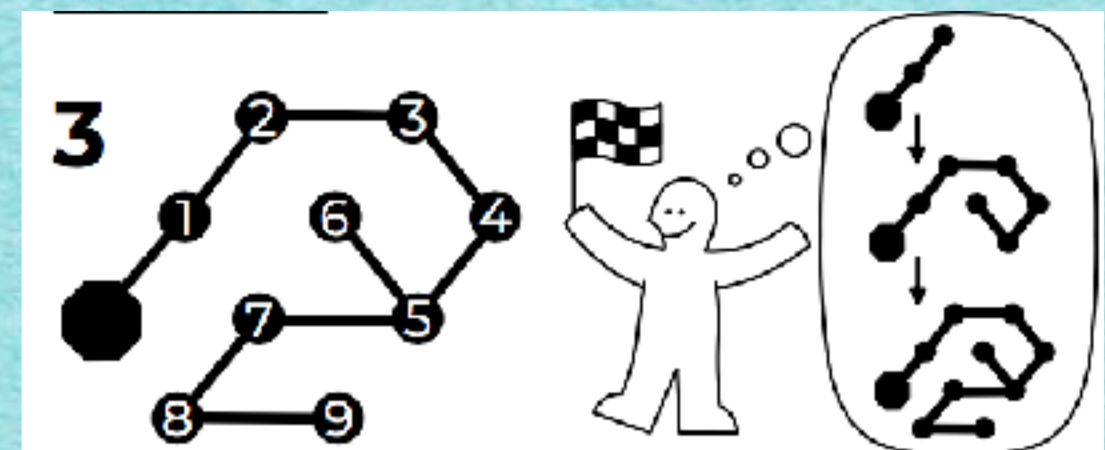


DEEP SEARCH



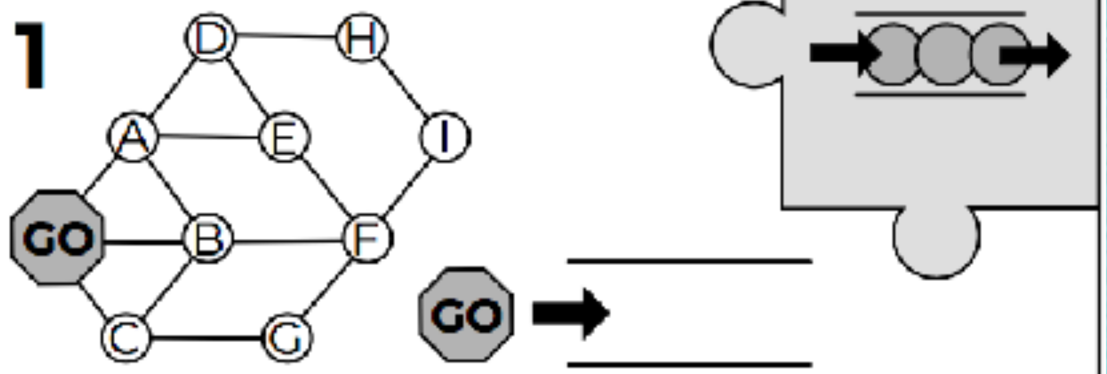


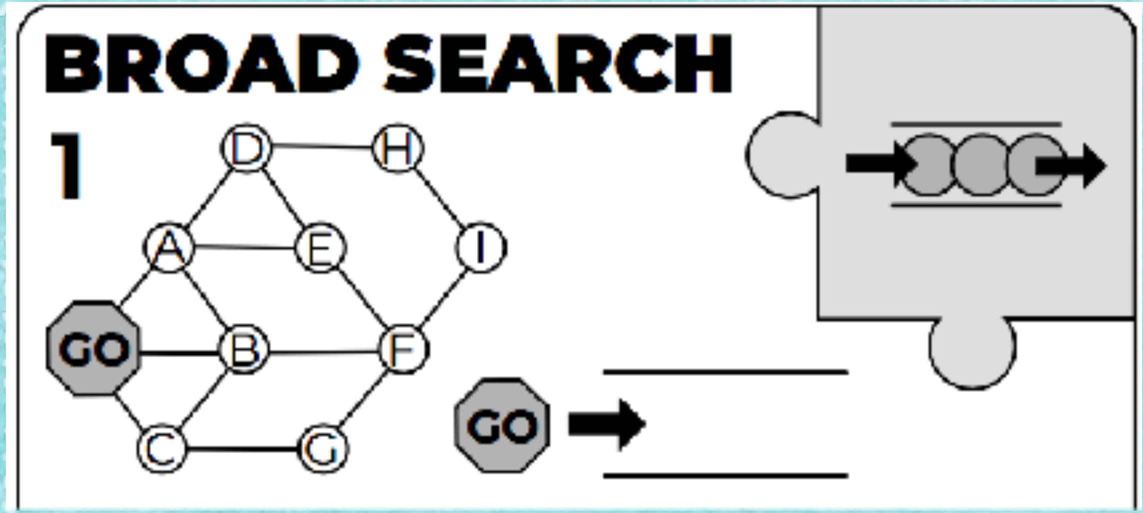


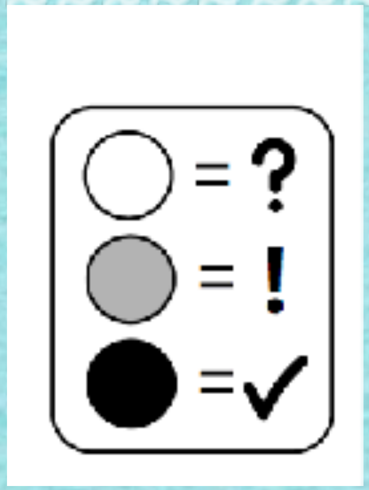


BROAD SEARCH

1

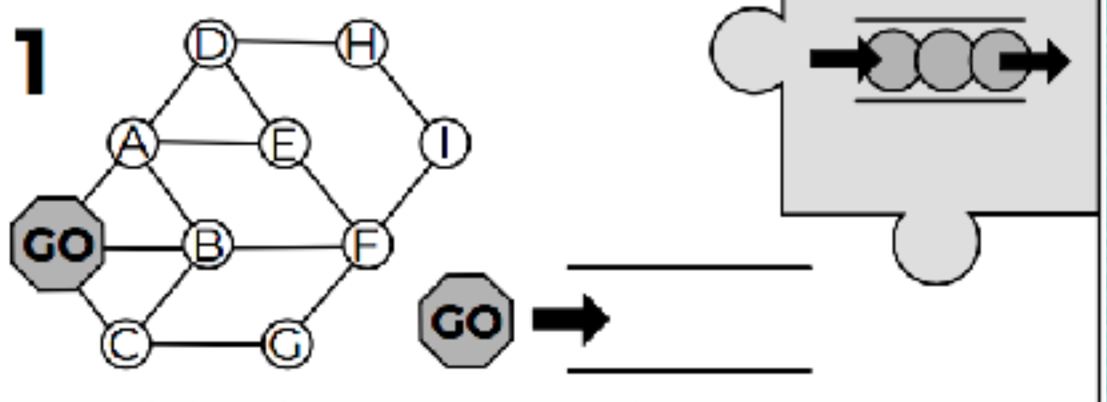


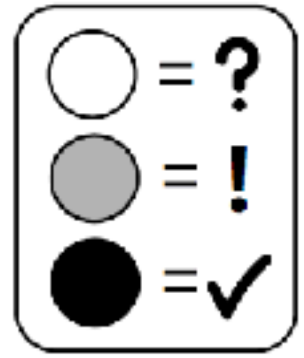

BROAD SEARCH
1
D
H
A
E
I
GO
B
F
C
G
GO


= ?
= !
=✓

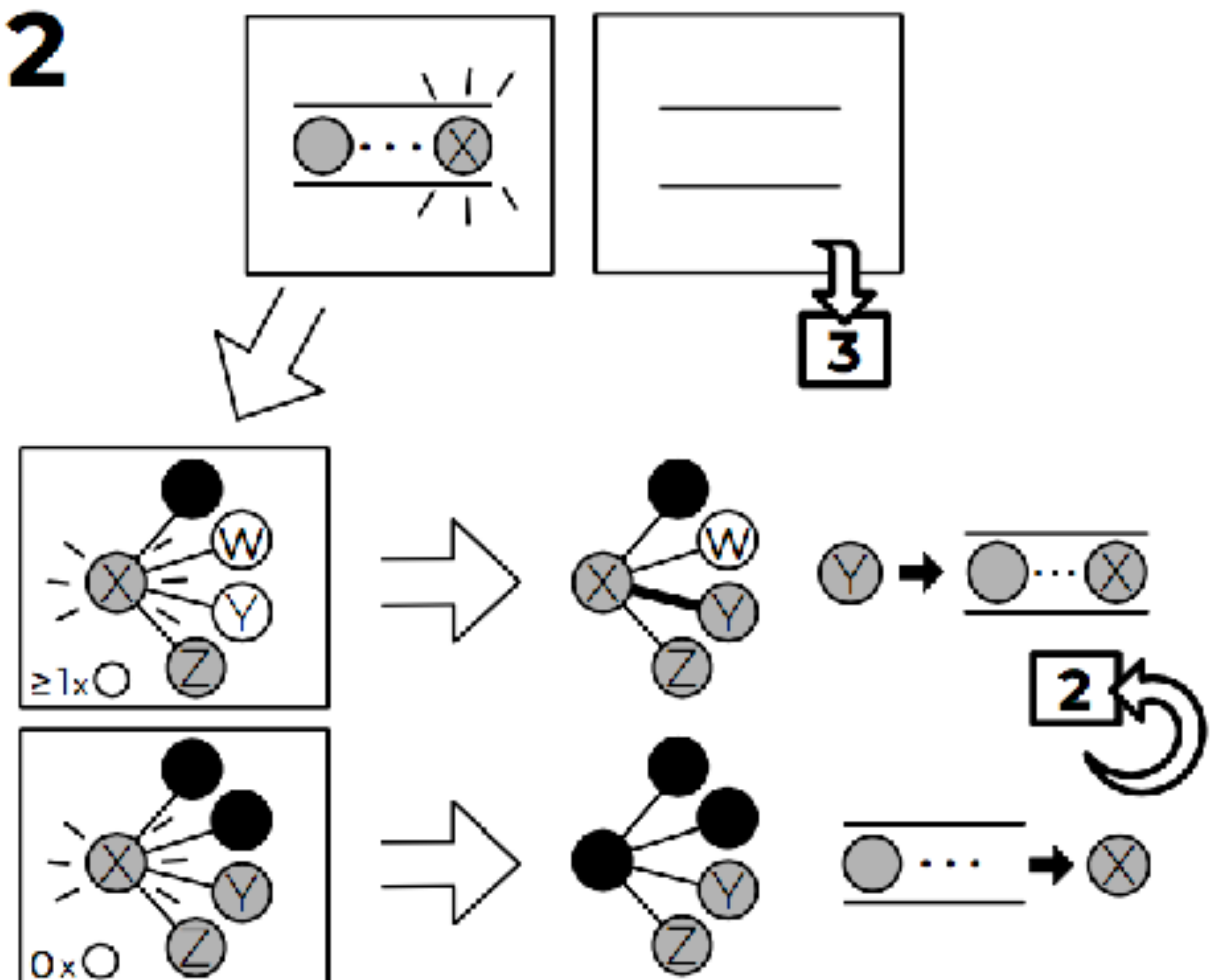
BROAD SEARCH

1

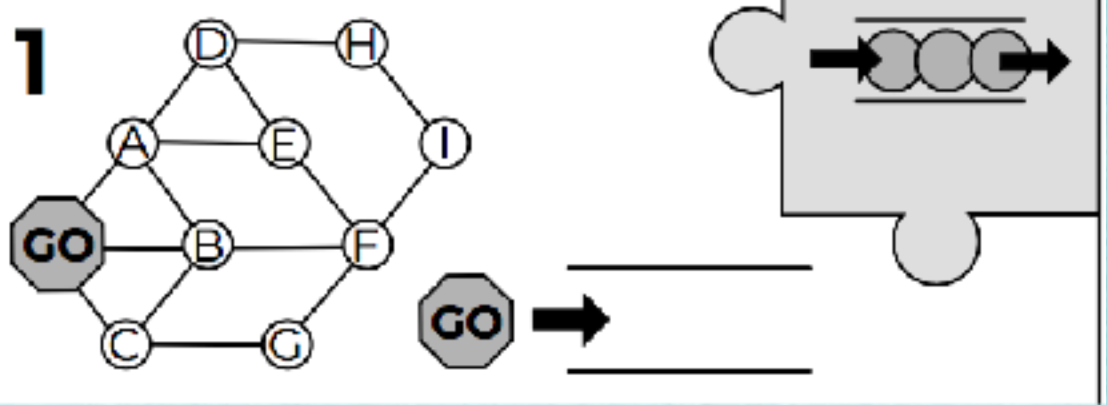


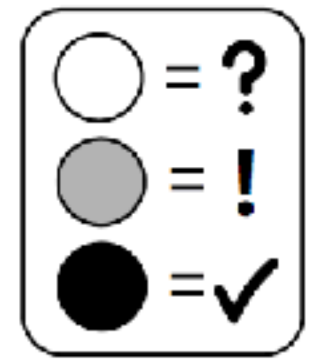


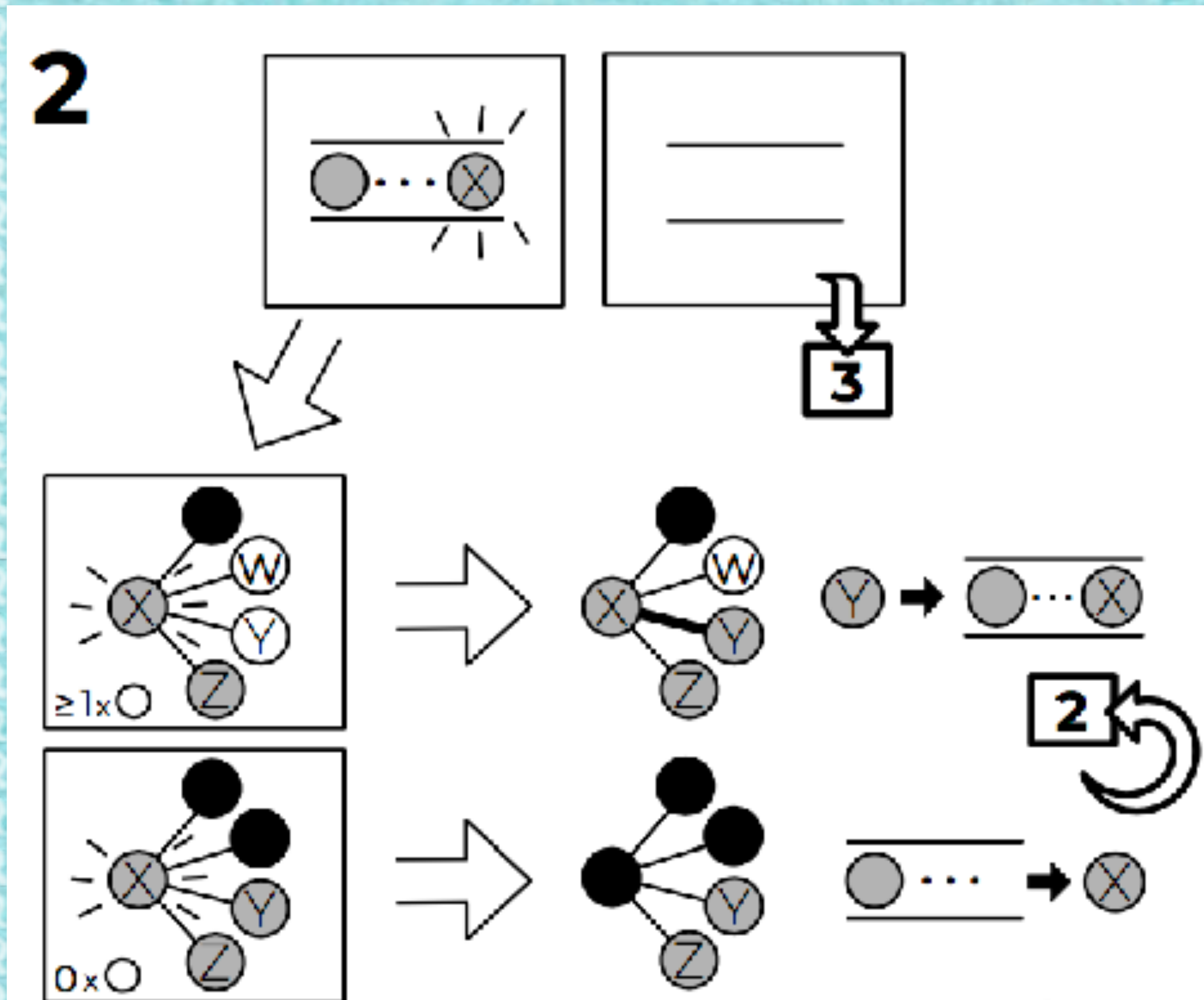
2

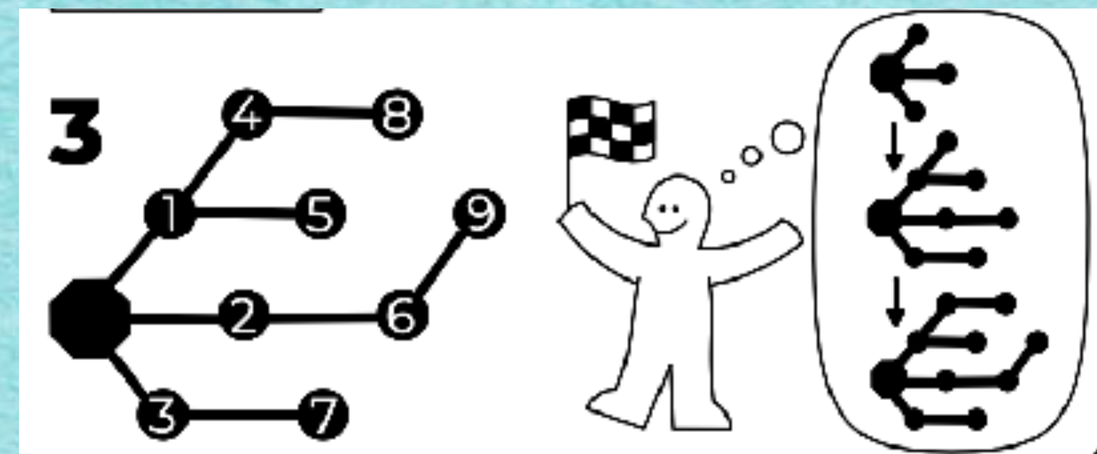


BROAD SEARCH



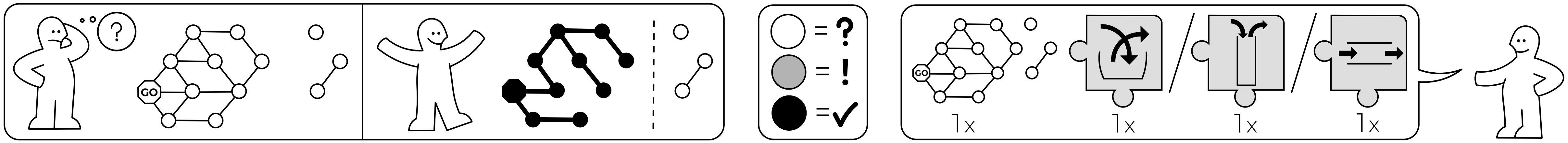




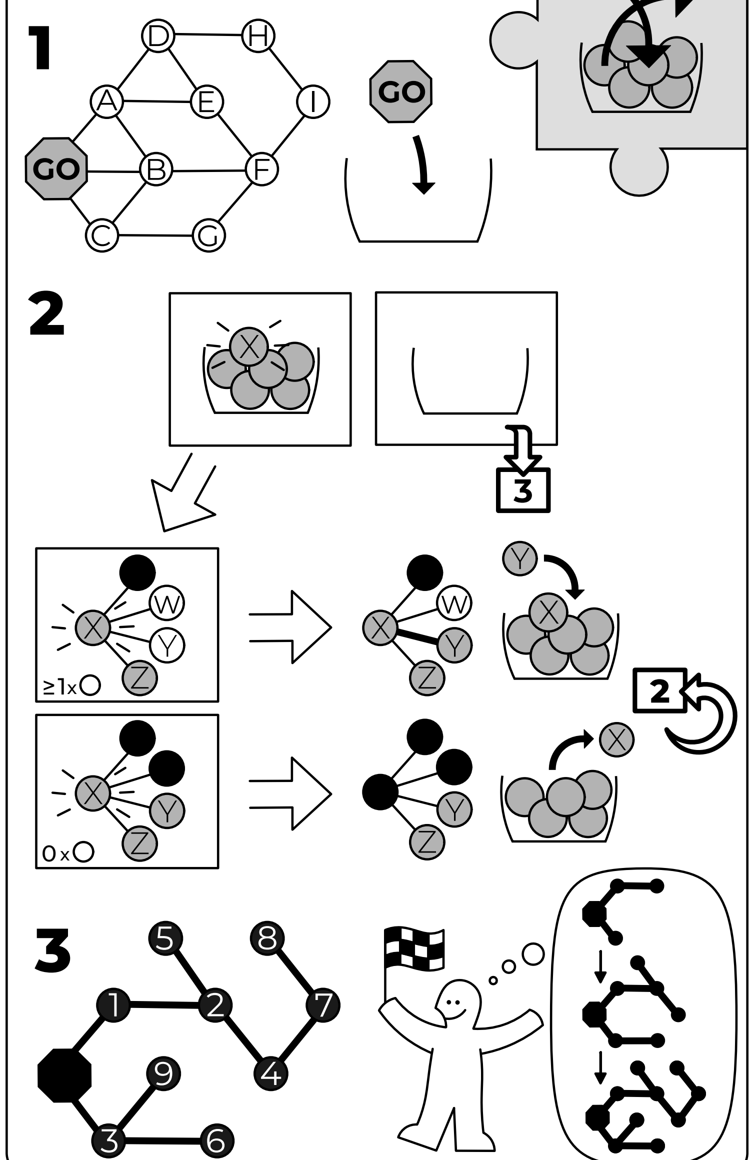


GRÅPH SKÄN

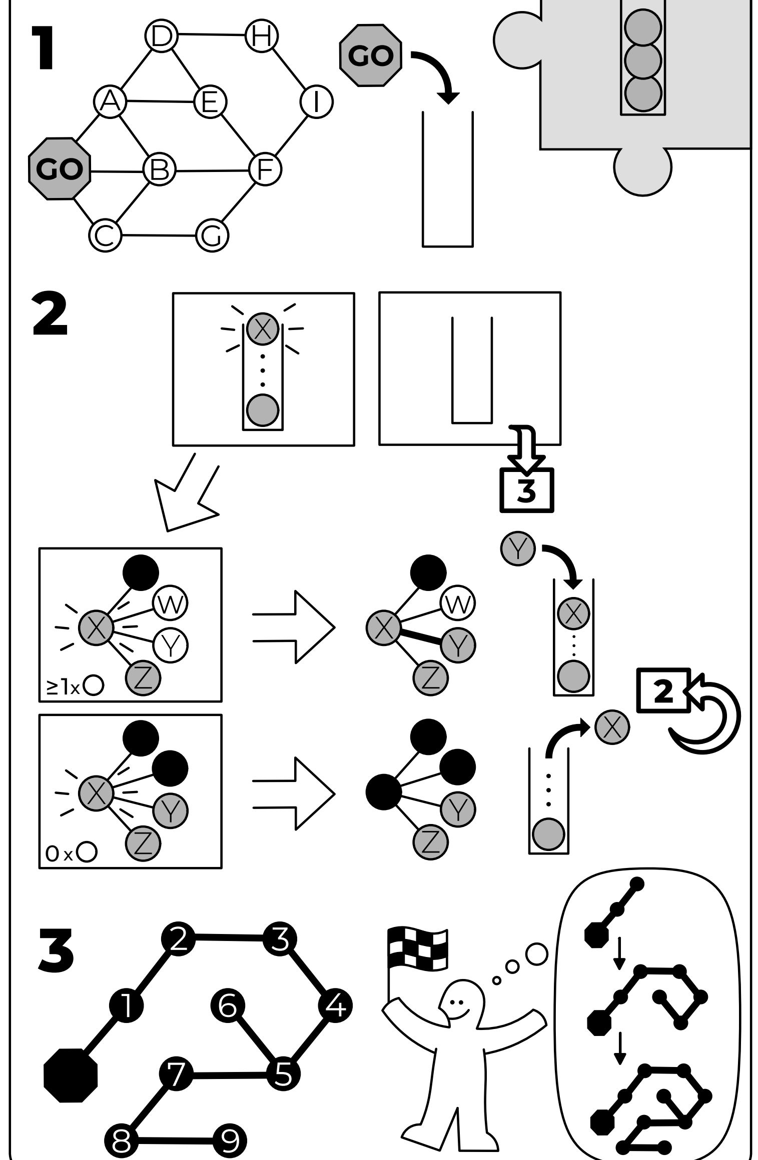
idea-instructions.com/graph-scan/



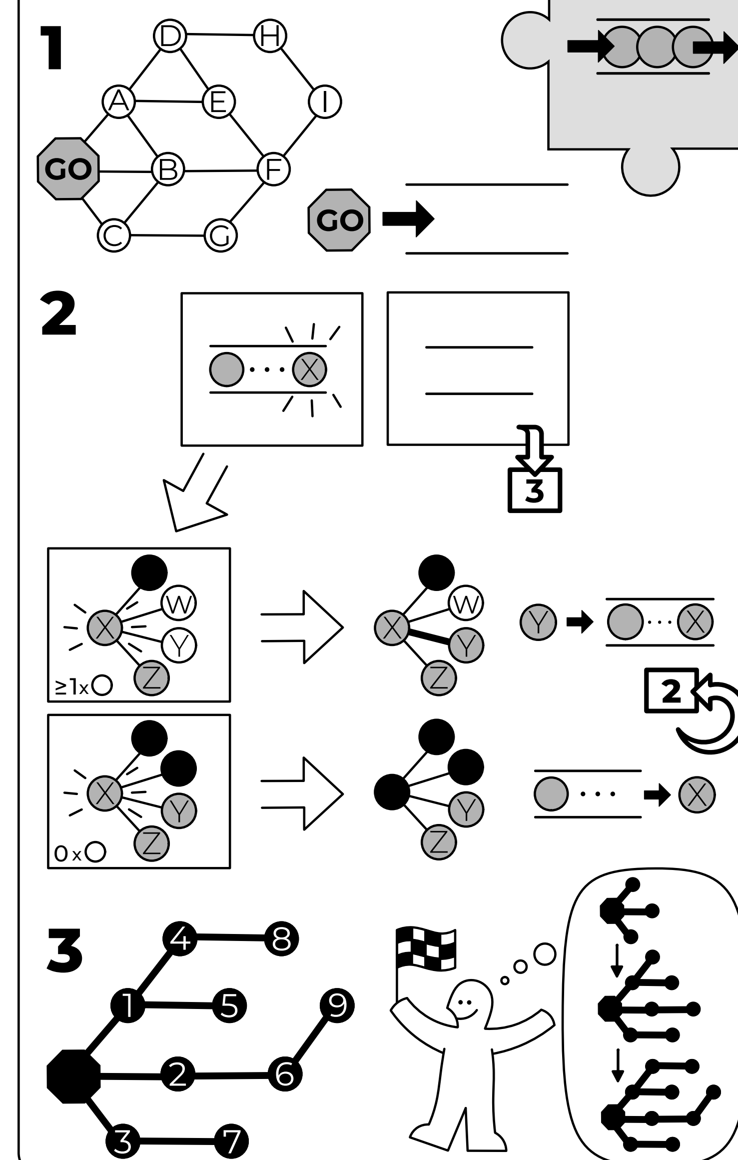
AD-HOC SEARCH



DEEP SEARCH



BROAD SEARCH



Kapitelende!

s.fekete@tu-bs.de