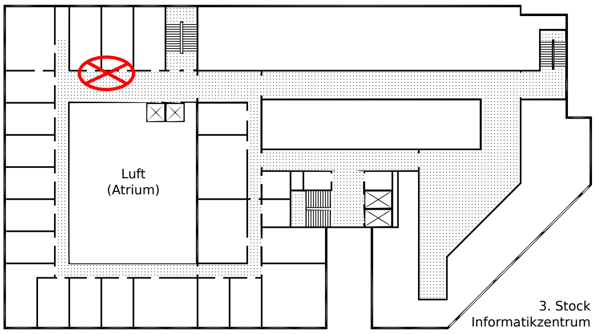
Algorithmen und Datenstrukturen II **SoSe 2019**
Dr. Linda Kleist **Abgabe:** 06.06.19
Arne Schmidt **Rückgabe:** 19./21.06.19

Übungsblatt 3

Abgabe der Lösungen bis zum 06.06.19 um 13:15 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!**



Dieses Blatt besteht aus einer Präsenzaufgabe, die in der kleinen Übung besprochen wird, sowie aus drei Hausaufgaben, die abgegeben werden müssen und bewertet werden.

Präsenzaufgabe:
In dieser Aufgabe betrachten wir den Branch-and-Bound-Algorithmus für Maximum Knapsack aus der Vorlesung. Sei $I := (z_1, \dots, z_n, Z, p_1, \dots, p_n)$ ein Instanz von Maximum Knapsack.

a) Betrachte einen Knoten v in einem Entscheidungsbaum T, für den die gültige Belegung $b_1, \dots, b_{l-1}$ gilt. Sei $U := UB(I, \ell, b)$ eine dazu passende **obere Schranke**. Für welche Teilbäume von T gilt U **immer** als obere Schranke?

b) Betrachte einen Knoten v in einem Entscheidungsbaum T, für den die gültige Belegung $b_1, \dots, b_{l-1}$ gilt. Sei $P := LB(I, \ell, b)$ eine dazu passende **untere Schranke**. Für welche Teilbäume von T gilt P **immer** als untere Schranke?

Im folgenden benutzen wir Greedy$_0$ (siehe Blatt 1) als untere Schranke und den Greedy-Algorithmus für Fractional Knapsack als obere Schranke.

c) Zeige: Für jedes $n \geq 1$ gibt es mindestens eine Instanz mit n Objekten, bei der der Branch-and-Bound-Algorithmus keine rekursiven Aufrufe startet.

d) Der Entscheidungsbaum, den der Branch-and-Bound-Algorithmus ablaufen kann, besitzt $2^{n+1} - 1$ viele Knoten. Da der Algorithmus Teilbäume abschneidet, besteht die Hoffnung, dass deutlich weniger Knoten besucht werden müssen.

   Zeige: Für jedes $n \geq 1$ gibt es eine Instanz mit n Objekten, bei der der Branch-and-Bound-Algorithmus mindestens $2^{\frac{n}{2}} - 1$ Knoten des gesamten Entscheidungsbaumes besucht.
   (Hinweis: Ein Entscheidungsbaum der Höhe h besitzt $2^h - 1$ Knoten. Finde also eine Instanz für jedes n, sodass alle Knoten auf den ersten $\frac{n}{2}$ Ebenen besucht werden müssen.)

Hausaufgabe 1 (Dynamische Programmierung für Knapsack): (6 Punkte)
Wende das dynamische Programm für Maximum Knapsack aus der Vorlesung auf folgende Instanz an:

i	1	2	3	4	5
z_i	5	4	3	5	6
p_i	1	3	1	2	4

und $Z = 11$.

Fülle dazu die folgende Tabelle aus, wobei der Eintrag in Zeile i und Spalte x dem Wert $\mathcal{P}(x, i)$ entspricht.

i \ x	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												

Hausaufgabe 2 (Gewichtetes Hörsaal-Problem): (2+3+3 Punkte)
Betrachte das gewichtete Hörsaal-Belegungs-Problem:
Gegeben Intervalle $(s_1, e_1), \ldots, (s_n, e_n)$ mit $e_1 \leq e_2 \leq \cdots \leq e_n$ und Gewichte $w_1, \ldots, w_n$
Gesucht Indexmenge $S \subseteq \{1, \ldots, n\}$ an disjunkten Intervallen mit $\sum\limits_{i \in S} w_i$ maximal

a) Sei $\mathcal{G}(i)$ der größte Wert, wenn die ersten i Intervalle zur Verfügung stehen. Begründe, dass die folgende Rekursionsgleichung einen optimalen Wert zurückgibt.
$$\mathcal{G}(i) := \begin{cases} 0 & \text{, falls } i = 0 \\ \max\left(\mathcal{G}(i-1), \mathcal{G}(\mathrm{p}(i)) + w_i\right) & \text{, sonst} \end{cases}$$
Dabei ist $\mathrm{p}(i) := \max\left(\{j \mid e_j \leq s_i\} \cup \{0\}\right)$. ($\mathrm{p}(i)$ gibt also den Index des Intervalls zurück, das am spätesten, aber noch vor Intervall I_i endet.)

b) Entwirf einen Algorithmus in Pseudocode, der $\mathrm{p}(0), \ldots, \mathrm{p}(n)$ bestimmt.

c) Entwirf einen Algorithmus in Pseudocode, der $\mathcal{G}(n)$ berechnet.

Hausaufgabe 3 (Branch-And-Bound): (6 Punkte)
In dieser Aufgabe betrachten wir den Branch-and-Bound-Algorithmus für Maximum Knapsack aus der Vorlesung. Wir benutzen Greedy_0 (siehe Blatt 1) als untere Schranke und den Greedy-Algorithmus für Fractional Knapsack als obere Schranke. Wende den Branch-and-Bound-Algorithmus auf folgende, nach $\frac{z_i}{p_i}$ aufsteigend sortierte Instanz für Maximum Knapsack an.

i	1	2	3	4
z_i	24	28	12	18
p_i	25	28	11	16

und $Z = 42$

Beachte folgende Punkte:

- Benutze den Enumerationsbaum aus Abbildung 1
- Beschrifte die Kanten mit der Auswahl, die getroffen wurde.
- Beschrifte die Knoten mit den aktuell besten Schranken (obere und untere).
- Sollte eine Auswahl unzulässig sein, beschrifte den jeweiligen Knoten mit *unzulässig*.
- Sollten Kanten nicht benutzt werden, streiche sie durch.
- Halte in einer Tabelle fest, welche Objekte eine neue, beste Lösung liefern.

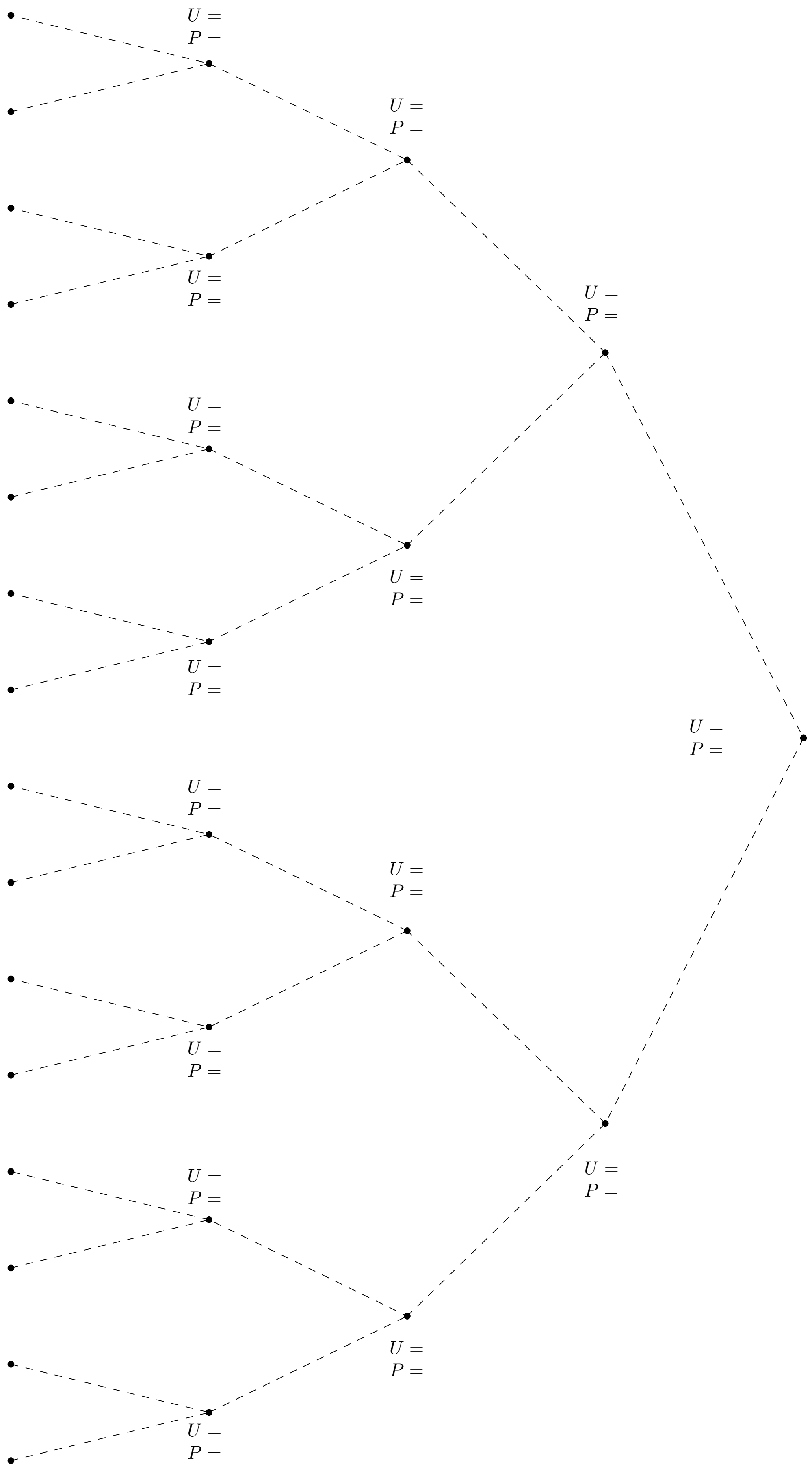


Abbildung 1: Ein Entscheidungsbaum.