Technische
Universität
Braunschweig

Algorithmen und Datenstrukturen II

3. Vorlesung

Linda Kleist, 08.05.2019

Rucksackprobleme...

teilen möglich

ganz oder gar nicht

Spezialfälle von Maximum Knapsack

Problem 2 (Maximum Knapsack)

Maximierungsproblem

Gegeben:
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i und Gewinn p_i
- Größenschranke Z

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i \leq Z$ und

$$\sum_{i \in S} p_i \text{ maximal.}$$

Problem 4

gleiche Gewinndichten

Gegeben:
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i
- Größenschranke Z

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i \leq Z$ und

$$\sum_{i \in S} z_i \text{ maximal.}$$

Spezialfälle von Maximum Knapsack

Problem 5 (Subset Sum)

Gegeben: - Objekte $\{O_1, ..., O_n\}$, je mit Größe z_i
- Zielgröße Z

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i = Z$.

Problem 6 (Partition)

Gegeben: - Objekte $\{O_1, ..., O_n\}$, je mit Größe z_i

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i = \sum_{i \notin S} z_i$.

Partition

Problem 6 (Partition)

Gegeben: - Objekte $\{O_1, ..., O_n\}$, je mit Größe z_i

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i = \sum_{i \notin S} z_i$.

Beispiel 1

Hat die Zahlenreihe $(7, 13, 17, 20, 24, 29, 31, 31, 56)$ mit Summe 228 eine Teilmenge mit Summe 114?

Antwort: Ja! Es gilt $7 + 20 + 31 + 56 = 114 = 13 + 17 + 24 + 29 + 31$.

Partition

Problem 6 (Partition)

Gegeben: - Objekte $\{O_1, ..., O_n\}$, je mit Größe z_i

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i = \sum_{i \notin S} z_i$.

Beispiel 2

Hat die Zahlenreihe $(7, 13, 17, 20, 29, 31, 31, 35, 57)$ mit Summe 240 eine Teilmenge mit Summe 120?

Antwort: Nein! *...Wie kann man das beweisen?...*

Lösen von Subset Sum

Problem 5 (Subset Sum)

Gegeben: - Objekte $\{1, \ldots, n\}$, je mit Größe z_i
- Zielgröße Z

Gesucht: Eine Menge $S \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in S} z_i = Z$.

Wir definieren eine Hilfsfunktion:

$$\mathcal{S} \colon \{0, \ldots, Z\} \times \{0, \ldots, n\} \to \{0, 1\}$$

$$\mathcal{S}(x, i) := \begin{cases} 1, & \text{falls Teilmenge von } z_1, \ldots, z_i \text{ Summe } x \text{ ergibt} \\ 0, & \text{sonst.} \end{cases}$$

Lösen von Subset Sum

Algorithmus 3 (Dynamic Programming für Subset Sum)

Eingabe: Zahlen $z_1, \ldots, z_n, Z$
Ausgabe: 1, falls $S \subseteq \{1, \ldots, n\}$ existiert mit $\sum_{i \in S} z_i = Z$; 0, sonst.

$\mathcal{S}(0,0) := 1$
for $(x = 1)$ to Z **do**
  $\mathcal{S}(x, 0) := 0$;
for $(i = 1)$ to n **do**
  for $(x = 0)$ to $(z_i - 1)$ **do**
    $\mathcal{S}(x, i) := \mathcal{S}(x, i-1)$;
  for $(x = z_i)$ to Z **do**
    if $(\mathcal{S}(x, i-1) = 1)$ OR $(\mathcal{S}(x - z_i, i-1) = 1)$ **then**
      $\mathcal{S}(x, i) := 1$;
    else
      $\mathcal{S}(x, i) := 0$;
return $\mathcal{S}(Z, n)$

$\mathcal{S}(x, i) := \max\{\mathcal{S}(x, i-1), \mathcal{S}(x - z_i, i-1)\}$

Lösen von Subset Sum

Satz 3

Algorithmus 3 entscheidet ob Subset Sum eine Lösung hat, die Laufzeit beträgt $O(nZ)$.

Beweis.

Tafel... $\square$