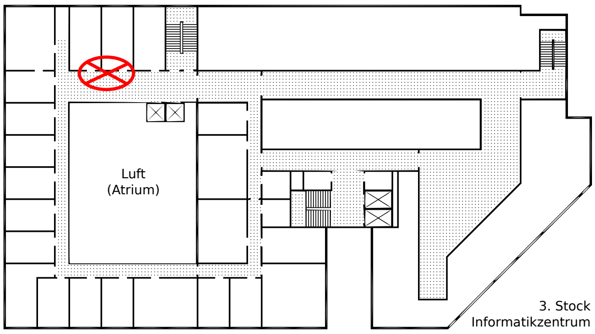
Algorithmen und Datenstrukturen 2 — **SoSe 2022**

Prof. Dr. Sándor Fekete — **Abgabe:** 22.06.22

Matthias Konitzny — **Rückgabe:** ab 29.06.22

Hausaufgabenblatt 3

Abgabe der Lösungen bis zum 22.06.22 um 12:00 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 beiden Namen, Matrikel- Übungs- und Gruppennummer versehen!**



Hausaufgabe 1 (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 (abgerundeten Wert des) Greedy-Algorithmus für Fractional Knapsack als obere Schranke.

a) 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	18	19	9	13
p_i	21	19	8	11

und $Z = 32$

Beachte folgende Punkte:

- Benutze den Enumerationsbaum aus Abbildung 1.[1]
- Beschrifte die Kanten mit der Auswahl, die getroffen wurde.
- Beschrifte die Knoten mit den aktuell besten Schranken (obere und untere).
- Ist eine Auswahl unzulässig, 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.

[1] Unter `https://aud2.ibr.cs.tu-bs.de` gibt es den Baum zusätzlich als .ipe und .svg, um die Daten direkt einzutragen.

Hausaufgabe 2 (Gewichtetes Hörsaal-Problem): (3+3 Punkte)
Betrachte das gewichtete Hörsaal-Problem:

Gegeben Intervalle $(s_1, e_1), \ldots, (s_n, e_n)$ mit $e_1 \leq e_2 \leq \cdots \leq e_n$ und Gewichten $w_1, \ldots, w_n$

Gesucht Indexmenge $S \subseteq \{1, \ldots, n\}$ an disjunkten Intervallen mit $\sum_{i \in S} w_i$ maximal.

Sei $\mathcal{G}(i)$ der größte Wert, wenn die ersten i Intervalle zur Verfügung stehen. Betrachte die folgende Rekursionsgleichung, die für $\mathcal{G}(i)$ immer einen optimalen Wert zurückgibt:

$$\mathcal{G}(i) = \begin{cases} 0 & \text{, falls } i = 0 \\ \max\left(\mathcal{G}(i-1), \mathcal{G}\left(p(i)\right) + w_i\right) & \text{, sonst} \end{cases}$$

Dabei ist $p(i) = \max(\{j | e_j \leq s_i\} \cup \{0\})$. ($p(i)$ gibt also den größten Index des Intervalls zurück, das am spätesten, aber noch vor Intervall I_i endet.)

a) Entwirf einen Algorithmus in Pseudocode, der $p(0), \ldots, p(n)$ bestimmt.

b) Entwirf einen effizienten Algorithus in Pseudocode, der $\mathcal{G}(n)$ berechnet.

Hausaufgabe 3 (Lageroptimierung): (3+5 Punkte)
Der Versandhändler Pesos benötigt Lagerhallen für seinen Betrieb Anasom, um Waren zu lagern und dann zu versenden. Allerdings möchte Anasom nur ein festes Sortiment anbieten, sodass die Waren dauerhaft in den Lagerhallen gelagert werden. Nun stellt sich die Frage, welche Waren angeboten werden sollen und welche Lagerhallen dafür benötigt werden, um möglichst viel Profit zu machen. Eine Ware i nimmt Raum $z_i > 0$ in Anspruch und liefert monatlich einen Erlös $p_i > 0$; eine Lagerhalle i liefert zusätzlichen Raum (entsprechend $z_i < 0$), kostet aber monatlich Miete (entsprechend $p_i < 0$).

Formal lässt sich das LAGERHALLEN-PROBLEM so beschreiben:

Gegeben: $z_1, \ldots, z_n, p_1, \ldots, p_n \in \mathbb{Z}$ mit $z_i/p_i \in \mathbb{R}^+$ für alle $i = 1, \ldots, n$.
Objekte mit $z_i < 0$ entsprechen Lagerhallen, Objekte mit $z_i \geq 0$ entsprechen Waren.

Gesucht: Eine Auswahl $S \subset \{1, \ldots, n\}$ mit

- $\sum_{i \in S} z_i \leq 0$
- $\sum_{i \in S} p_i$ maximal

a) Zeige: Es gibt Instanzen dieses Problems, für die der Branch-and-Bound-Algorithmus für MAXIMUM KNAPSACK aus der Vorlesung keine optimale Lösung liefert. Wir nehmen dabei an, dass GREEDY_0 (siehe Hausaufgabenblatt 1) als untere Schranke und der Greedy-Algorithmus für FRACTIONAL KNAPSACK als obere Schranke genutzt wird. Der Test der Zulässigkeit (zweite Zeile im Algorithmus 3.1 in der Vorlesung) wird auf

$$\textbf{\textit{if}} \sum_{j=1}^{\ell-1} b_j z_j > 0 \quad \textbf{\textit{then return}}$$

geändert.

b) Jetzt überlegt Anasom-Chef Pesos, einen Informatiker anzustellen, der einen Algorithmus entwirft, der das LAGERHALLEN-PROBLEM löst. Dieser sieht, dass man den Branch-and-Bound-Algorithmus für MAXIMUM KNAPSACK doch noch verwenden kann, indem man den Input anpasst.

Zeige: Jede Instanz I des LAGERHALLEN-PROBLEMS lässt sich in eine Instanz I' von MAXIMUM KNAPSACK in $O(n)$ Zeit transformieren, so dass umgekehrt aus einer optimalen Lösung P'_{OPT} für I' eine optimale Lösung P_{OPT} für I abgelesen werden kann. Begründe außerdem kurz, dass deine Transformation korrekt ist und die Laufzeit einhält.

(Hinweis:
(1) MAXIMUM KNAPSACK benötigt eine Kapazität und keine negativen Gewichte/Werte.
(2) Kapazität und Kosten der Lagerhallen müssen verändert werden.)

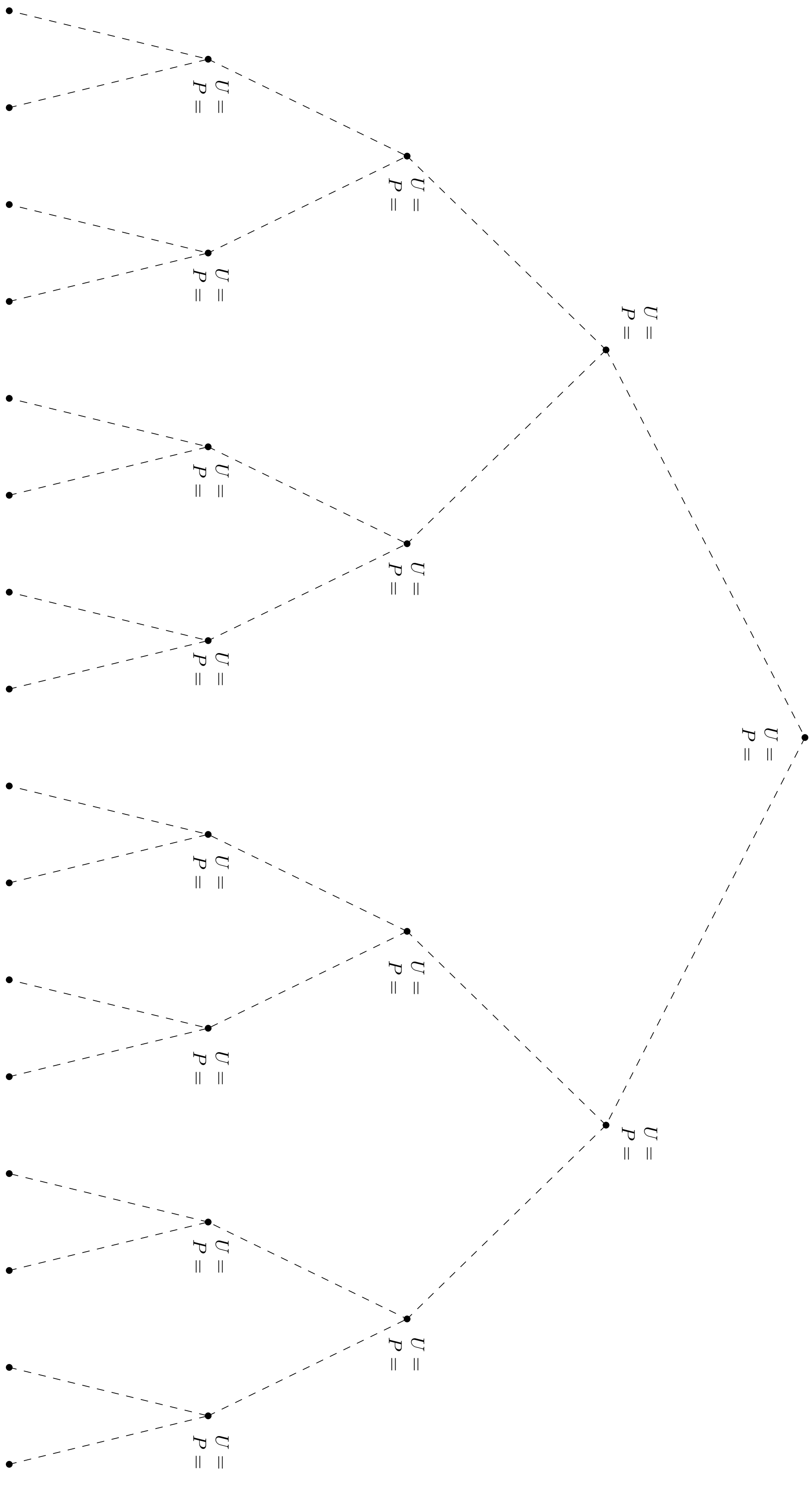


Abbildung 1: Ein Entscheidungsbaum.