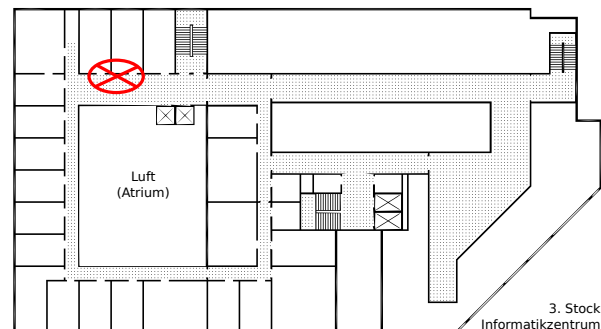


Hausaufgabenblatt 3

Abgabe der Lösungen bis zum 06.06.2025 um 13:30 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Auf deine Abgabe unbedingt Namen, Matrikelnummer und deine Gruppennummer schreiben! Die Blätter bitte zusammenheften!**



Hausaufgabe 1 (Branch-and-Bound):

(8+2+3+3 Punkte)

In dieser Aufgabe betrachten wir den Branch-and-Bound-Algorithmus für MAXIMUM KNAPSACK aus der Vorlesung. Wir benutzen GREEDY₀ (siehe Blatt 1) als untere Schranke (P) und den (abgerundeten Wert des) Greedy-Algorithmus für FRACTIONAL KNAPSACK als obere Schranke (U). Im Folgenden nennen wir ein Item i *fixiert*, wenn der dazugehörigen Variable b_i durch Branch & Bound bereits ein Wert zugewiesen wurde.

- a) Wende den Branch-and-Bound-Algorithmus auf folgende Instanz für MAXIMUM KNAPSACK an. (Hinweis: Die Reihenfolge der Elemente soll im Branch-and-Bound Baum erhalten bleiben, unabhängig von der möglicherweise abweichenden Sortierung für Greedy.)

i	1	2	3	4	
z_i	15	15	8	11	und $Z = 28$
p_i	14	16	8	10	

Beachte folgende Punkte:

- Benutze den Enumerationsbaum¹ auf Seite 4 aus Abbildung 1.
 - Beschrifte die Kanten mit der Auswahl, die getroffen wurde und die Knoten mit den aktuell besten Schranken (obere (U) und untere (P)).
 - 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.
- b) Zeige: Für jedes $n \geq 1$ gibt es eine Instanz mit n Objekten, bei der der Branch-and-Bound-Algorithmus keine rekursiven Aufrufe startet.

¹Auf der Kursseite gibt es den Baum zusätzlich als .pdf, falls ihr lieber darin eure Eintragungen machen wollt.

- c) Der Algorithmus aus der Vorlesung trifft Entscheidungen auf den einzelnen Variablen stets in der Reihenfolge, in der sie in der Instanz auftauchen. Insbesondere kommen auf jedem Pfad von der Wurzel des Entscheidungsbaums zu dessen Blättern die Variablen immer in der gleichen Reihenfolge vor.

Der Branch & Bound-Algorithmus bleibt korrekt, wenn man an jedem Suchbaumknoten nach irgendeinem Verfahren individuell eine noch nicht fixierte Variable b_i auswählt und hierfür als nächstes die Möglichkeiten $b_i = 0$ und $b_i = 1$ ausprobiert.

Zeige oder widerlege: Das Verfahren, nach dem wir die nächste fixierte Variable wählen, hat keinen Einfluss auf die Anzahl Knoten des Suchbaums, die wir besuchen. Gib also entweder eine Beispielinstant an, bei der sich durch eine Änderung des Verfahrens zur Auswahl der nächsten fixierten Variablen die Anzahl besuchter Knoten des Suchbaums ändert, oder begründe, warum es eine solche Instanz nicht geben kann.

- d) Nach Fixierung einer Variable b_i untersucht der Algorithmus aus der Vorlesung stets den Teilbaum mit $b_i = 0$ zuerst. Der Branch & Bound-Algorithmus bleibt korrekt, wenn man sich an jedem Suchbaumknoten individuell nach irgendeinem Verfahren entscheidet, ob zunächst der Fall $b_i = 0$ oder der Fall $b_i = 1$ untersucht werden soll.

Zeige oder widerlege: Das Verfahren, nach dem wir den zuerst untersuchten Fall wählen, hat keinen Einfluss auf die Anzahl Knoten des Suchbaums, die wir besuchen. Gib also entweder eine Beispielinstant an, bei der sich durch eine Änderung des Verfahrens zur Auswahl des zuerst untersuchten Falls die Anzahl besuchter Knoten des Suchbaums ändert, oder begründe, warum es eine solche Instanz nicht gibt.

Hausaufgabe 2 (Linear-/Integer Programming):

(4 Punkte)

Wir betrachten das folgende Integer Program ($x_i \in \mathbb{N}$).

$$\begin{aligned} \max \quad & 3x_1 + 5x_2 \\ \text{sodass} \quad & 2x_1 + 4x_2 \leq 25 \\ & x_1 \leq 8 \\ & 2x_2 \leq 10 \\ & x_1, x_2 \in \mathbb{N}_0^+ \end{aligned}$$

Löse das obige Integer Program mit Hilfe von dem Branch-and-Bound-Algorithmus. Löse dafür in jedem Knoten das relaxierte Linear Program ($x_i \in \mathbb{R}$). Wenn eine Variable x_i in der Lösung nicht ganzzahlig ist, dann betrachte die beiden Fälle $x_i \leq \lfloor x_i \rfloor$ und danach $x_i \geq \lceil x_i \rceil$. Gib in jedem Knoten an, welche obere Schranke U und welche untere Schranke P gilt. Wir erhalten eine untere Schranke, wenn die Variablen der optimalen LP-Lösung ganzzahlig ist. Ist noch keine untere Schranke bekannt, gilt $P = 0$.

(Hinweis: <https://online-optimizer.appspot.com> bietet einen online-Solver an, der lineare Programme lösen kann. Siehe auch folgende Seite.)

Für die Webseite <https://online-optimizer.appspot.com> kann folgendes Modell genutzt werden, welches auf der Webseite unter „Model“ eingetragen werden kann.

```
var x1 >= 0;
var x2 >= 0;

maximize z:      3*x1 + 5*x2;

subject to c11: 2*x1 + 4*x2 <= 25;
subject to c13:  x1          <=  8;
subject to c12:          2*x2 <= 10;

end;
```

Um das LP zu lösen muss man dann unten links auf „Solve Model“ klicken. Unter „Solution“ → „Variables“ steht dann die Belegung der Variablen einer optimalen Lösung. Unter „Model Overview“ ist dann der Wert der Lösung „Optimal objective value“

Um einen Branch (z.b. $x_1 \leq 1$) weiter zu betrachten, kann im Model ein neuer Constraint hinzugefügt werden:

```
subject to branch1: x1 <= 1;
```

Möchte man den anderen Branch (also $x_1 \geq 2$) testen, muss der vorherige Constraint gelöscht werden und ein neuer Constraint wie folgt hinzufügen:

```
subject to branch1: x1 >= 2;
```

Wenn es mit den gegebenen Constraints keine gültige Lösung mehr gibt, gibt der Solver ‘infeasible’ aus.

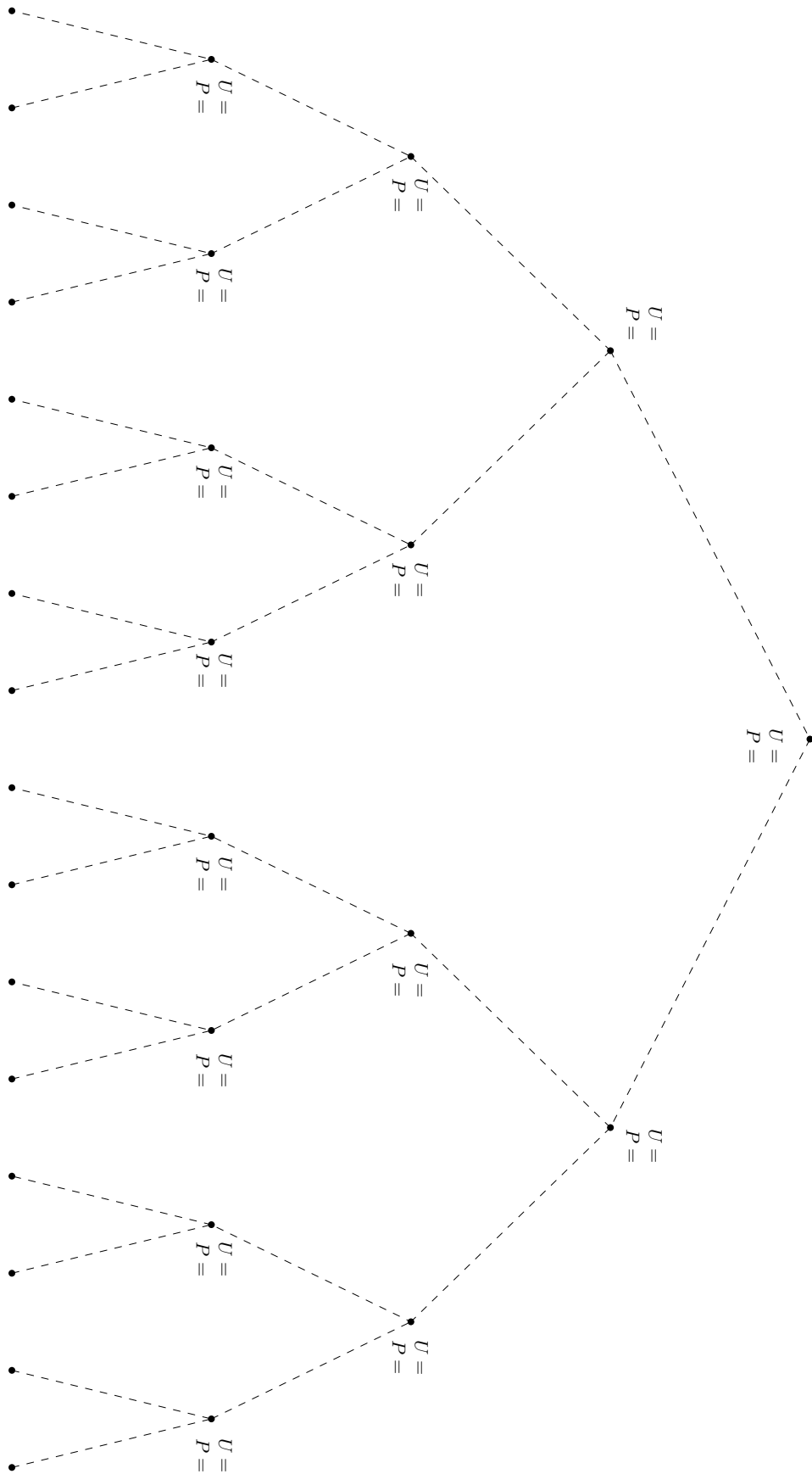


Abbildung 1: Ein Entscheidungsbaum.