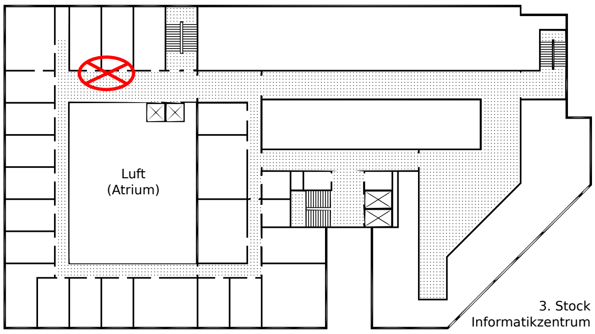
Prof. Dr. Sándor Fekete
Dr. Phillip Keldenich
Abgabe: 14.06.2023
Rückgabe: ab 20.06.2023

Hausaufgabenblatt 3

Abgabe der Lösungen bis zum 14.06.2023 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. **Bitte die Blätter zusammenheften und vorne deutlich mit Namen, Matrikel- Übungs- und Gruppennummer versehen!**



Hausaufgabe 1 (Branch-and-Bound): **(6+2+3+3+3+3 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. 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.

i	1	2	3	4
z_i	13	19	9	18
p_i	12	20	9	22

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.

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.

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

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 sich an jedem Suchbaumknoten nach irgendeinem Verfahren individuell eine noch nicht fixierte Variable b_i auswählt und als nächstes für diese Variable 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 Beispielinstanz 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 Beispielinstanz 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 geben kann.

e) Eine verbreitete Erweiterung von Branch & Bound führt die sogenannte Propagation ein. Dabei wird vor jedem Baumknoten die aktuelle Teilbelegung der Variablen analysiert, um basierend auf bestimmten Regeln indirekte Konsequenzen der bisherigen Teilbelegung zu finden und die Teilbelegung zu erweitern.

   Für Maximum Knapsack mit nicht-negativen Werten und nicht-negativen Gewichten kann man beispielsweise die folgenden Regeln benutzen:

   - Falls ein Item unter der aktuellen Teilbelegung nicht mehr genommen werden kann, weil es nicht mehr in die verbleibende Kapazität passt, setze die dazugehörige Variable auf 0. Für die Instanz aus Aufgabenteil a) würde zum Beispiel die Teilbelegung $b_4 = 1$ automatisch um $b_2 = 0$ ergänzt, weil $z_2 + z_4 = 37 > 32 = Z$ gilt.
   - Falls alle nicht fixierten Items gleichzeitig in den Rucksack passen, setze $b_i = 1$ für all diese Items.

   Diese Regeln werden auf der Teilbelegung jedes Suchbaumknoten so lange angewendet, bis sich keine Regel mehr anwenden lässt. Führe Branch & Bound mit diesen Erweiterungen auf der Instanz aus Aufgabenteil a) aus. Verzweige dabei analog zum Algorithmus aus der Vorlesung stets auf der nicht-fixierten Variablen b_i mit niedrigstem Index und betrachte stets den Fall $b_i = 0$ zuerst. Notiere dabei die durch Propagation entstandenen Zuweisungen in der Form $b_2 = 1 \to b_4 = 0, b_3 = 0$ an den

Kanten.
Beachte: Knoten ohne nicht-fixierte Variablen sind Blätter!

f) Gib eine kurze Begründung an, warum der Branch & Bound-Algorithmus für MAXIMUM KNAPSACK mit den Propagationsregeln aus Aufgabenteil e) weiterhin korrekt ist, d.h., weiterhin immer eine optimale Lösung bestimmt.

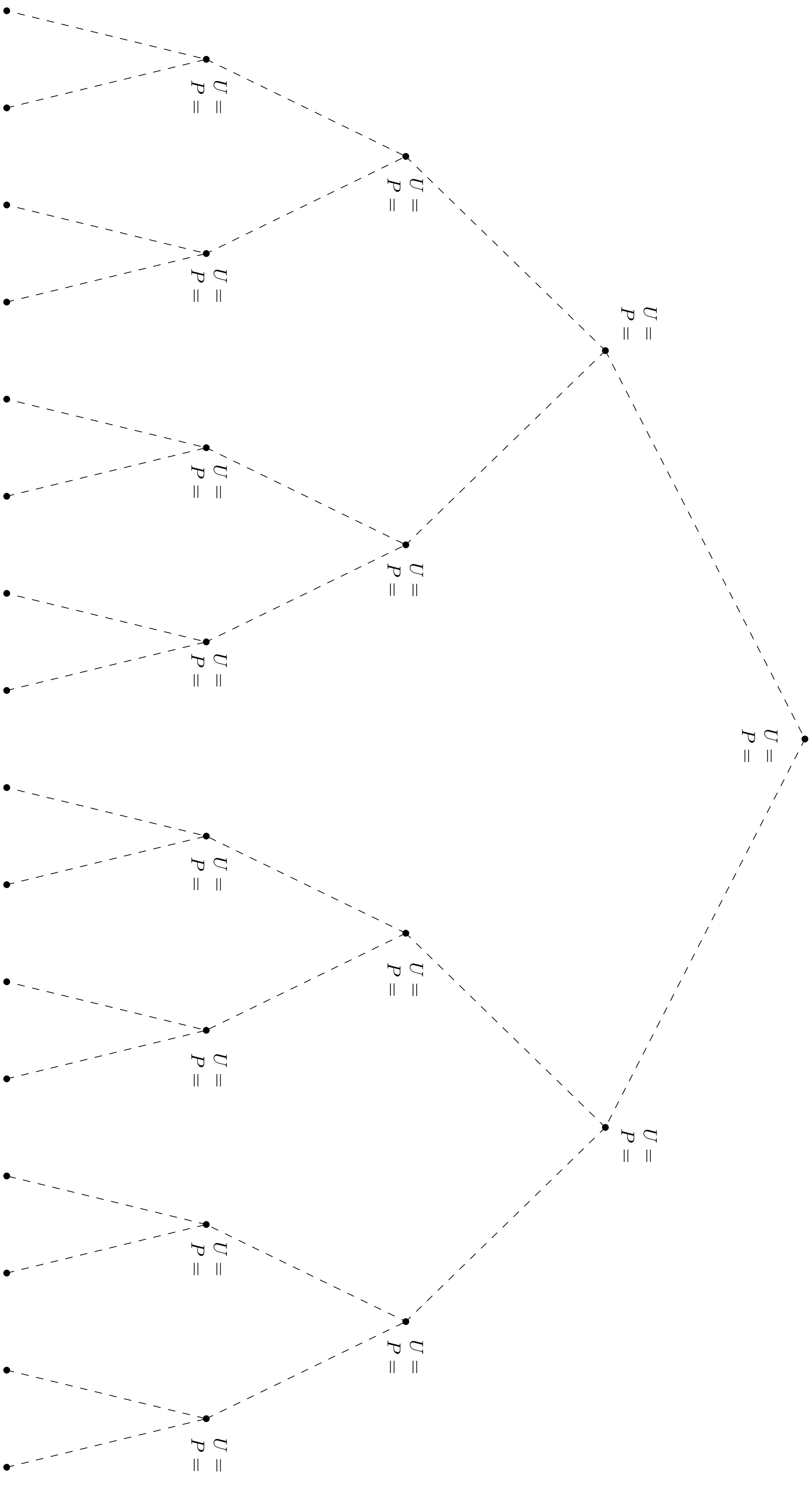
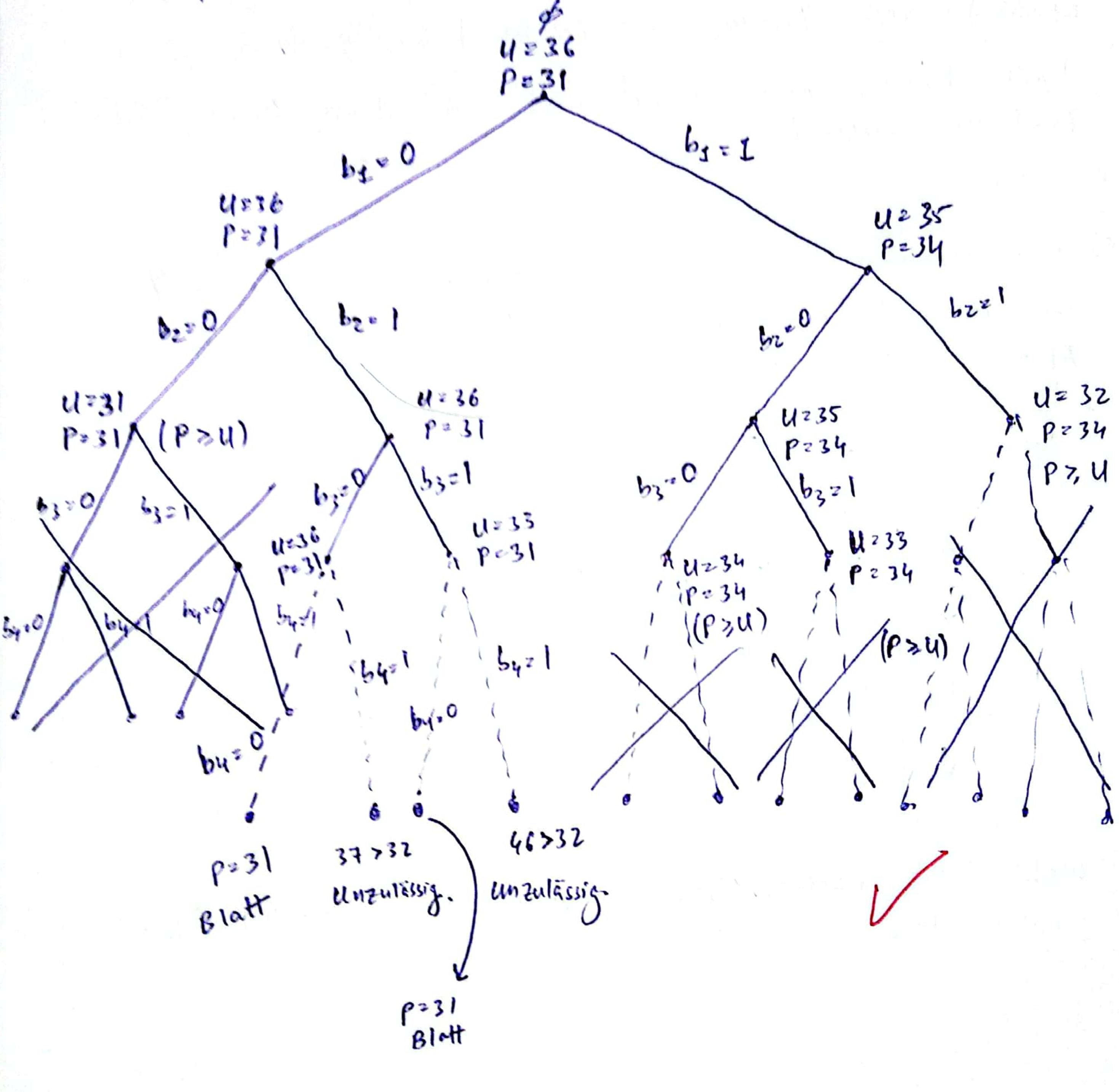


Abbildung 1: Ein Entscheidungsbaum.

a)

i	1	2	3	4
z_i	13	19	9	18
p_i	12	20	9	22
RF	4	2	3	1

Menge	Wert
$\{3,4\}$	31
$\{1,4\}$	34



b) Sei eine Instanz mit n sortierten Objekten und $Z \in \mathbb{N}$.
Hier sei die Summe der Kostenwerte von den ersten i Objekten gleich Z.

Daher sind Upper bound U und bester bekannter Lösungswert P gleich, also $P = U$. Anders ausgedruckt, der Resultat von Greedy$_0$ und Fractional Greedy gleich.

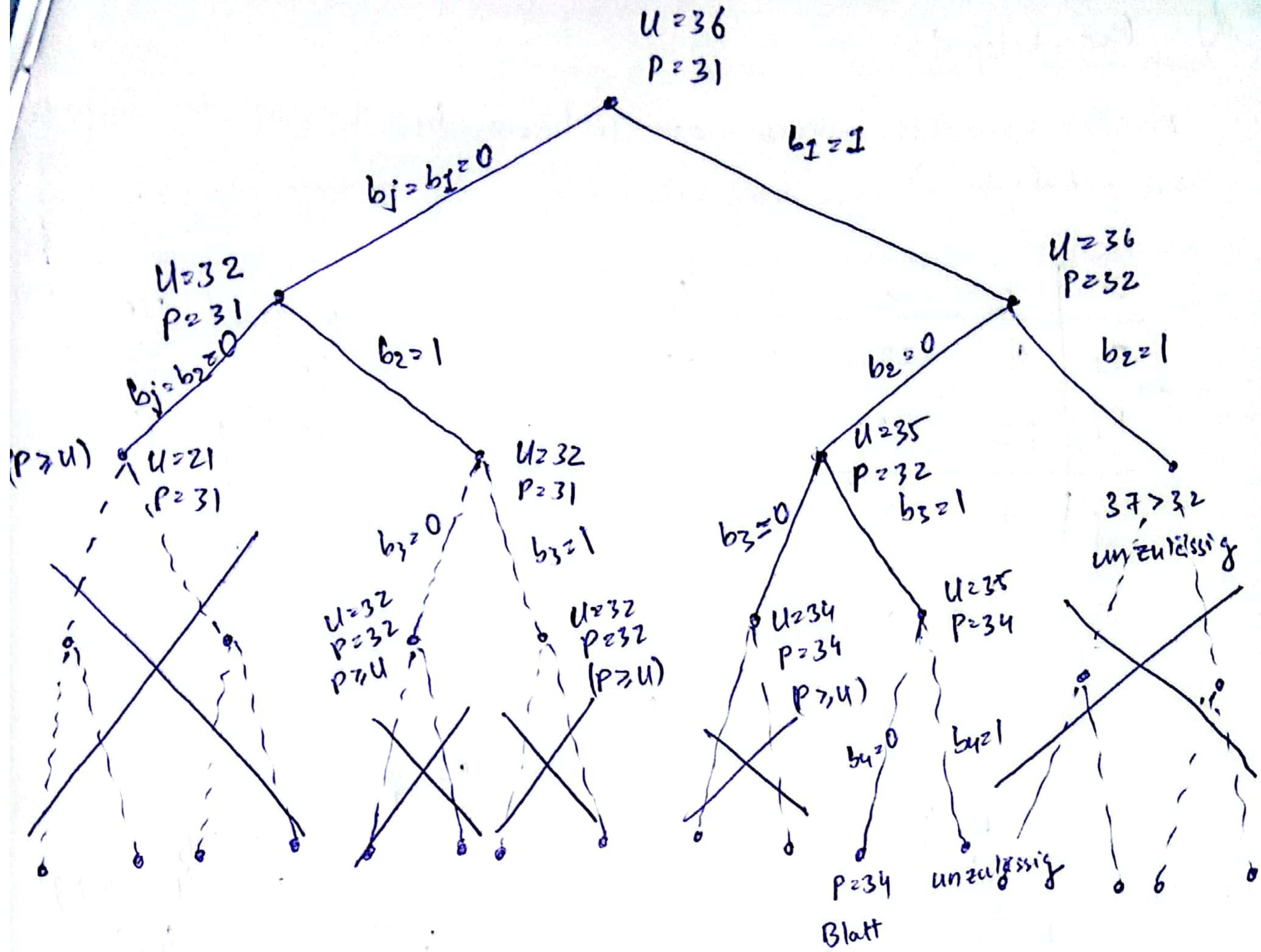
Laut Branch-and-Bound Algo wird dann keine weitere Funktion aufgerufen.

c) Das Verfahren von der Auswahl der nächsten fixierten Variablen hat einen Einfluss auf die Anzahl Knoten des Suchbaums.
Also Gegenbeispiel:

Sei Instanz von Aufgabe a)

i	1	2	3	4
z_i	13	19	9	18
P_i	12	20	9	22
j	4	2	3	1

Wähle die nächste Variable von schon sortierten Objekten, also b_j.



D.h. Wenn die nächste Variable in der Instanz mit sortierten Objekten ausgewählt wird, werden 13 ~~Ba~~ Knoten besucht.

Im Fall der Auswahl von ~~Objekten in~~ unsortierten Objekten müssen 15 Knoten besucht werden.

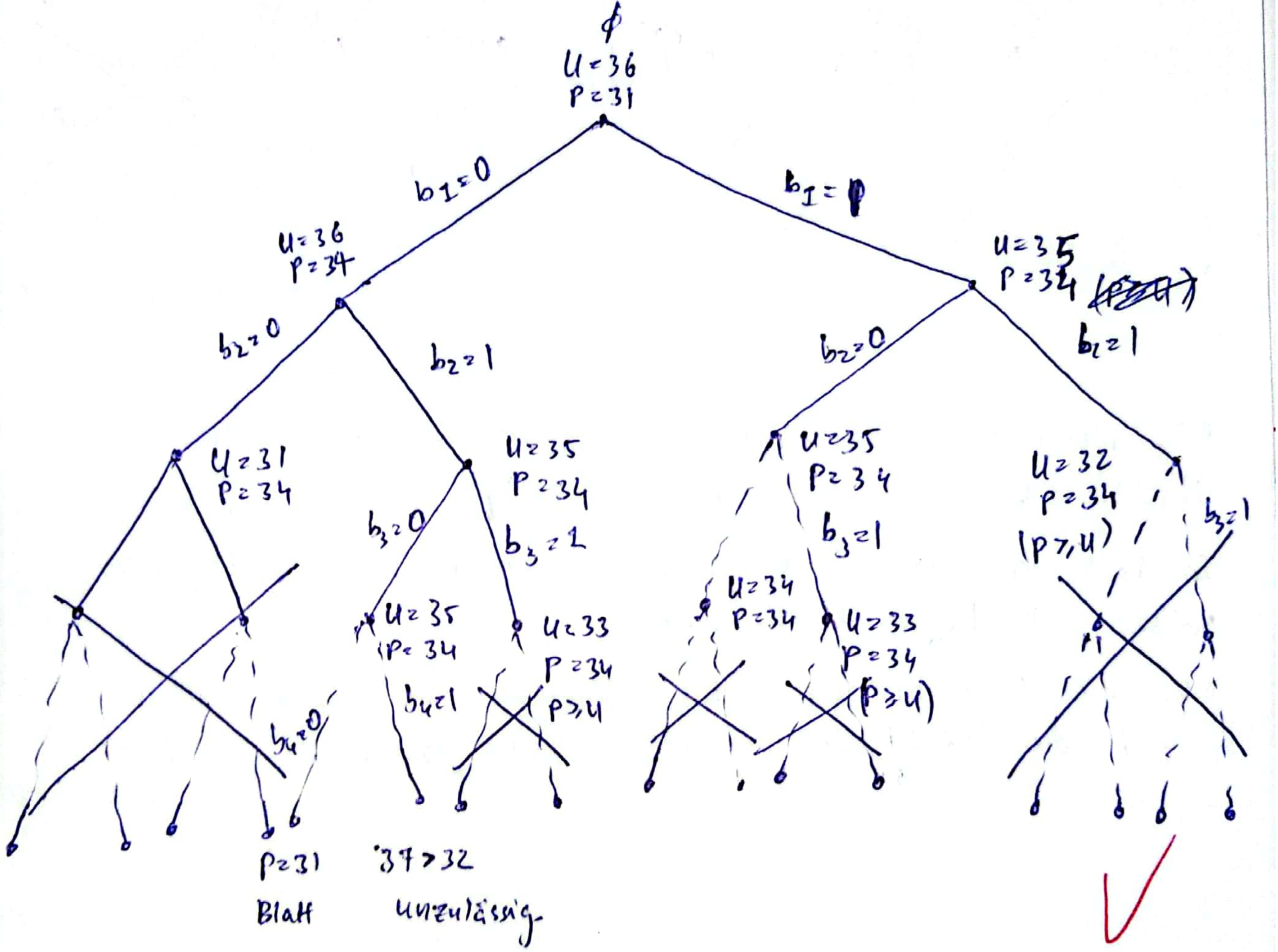
D.h. die Auswahl der nächsten Variable hat einen Einfluss auf die Anzahl der besuchten Suchbaumknoten.

d) <u>Gegenbeispiel:</u>

Wähle zunächst immer den Teilbaum $b_i = 1$ in der Instanz von Aufgabe a)

i	1	2	3	4
z_i	13	19	9	18
p_i	12	20	9	22
j	4	2	3	1

Menge	Wert
$\{2,3\}$	31
$\{4,1\}$	34



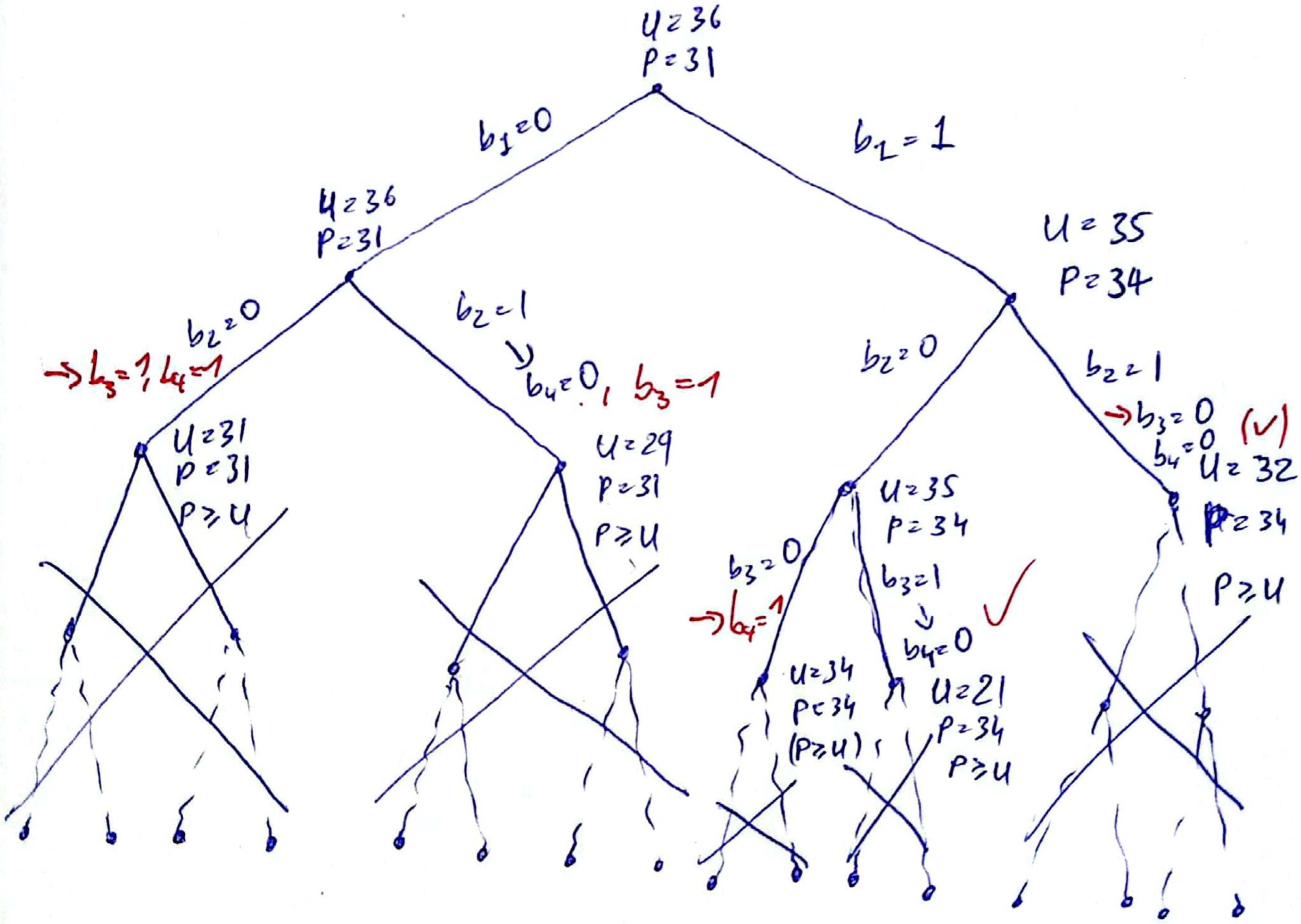
Wenn immer der Teilbaum $b_i = 1$ besucht wird, werden 13 Baumknoten besucht. Dagegen werden mit $b_i = 0$ 15 Baumknoten besucht.

Die Änderung der Auswahl des zunächst untersuchten Falls beeinflusst die Anzahl der besuchten Suchbaumknoten.

e)

i	1	2	3	4
z_i	13	19	9	18
p_i	12	20	9	22
j	4	2	3	1

Menge	Wert
$\{4, 3\}$	31
$\{1, 4\}$	34



1/3

Falls ein Item unter der aktuellen Teilbelegung nicht mehr genommen werden kann, weil es nicht mehr in die verbleibende Kapazität passt, dann gehört das Item nicht zu der Menge, die gültige bessere Lösung in der aktuellen bisherigen Teilbelegungen liefert.

Falls obere Schranke U kleiner als unten Schranke P, kann man keine bessere Lösung beim diesen Teilbaum finden.

D.h. die beste Lösung wird gefunden.

Falls alle nicht fixierten Items gleichzeitig in den Rucksack passen, werden alle diese Items $b_i = 1$ gesetzt.

Diese Items gehören dann zur Menge, die mögliche beste Lösung in der aktuellen Teilbelegung liefert.

Wenn jetzt $P \leq U$, dann liegt keine weitere bessere Lösung in dem Teilbaum und wird er damit abgeschnitten.

D.h. die beste Lösung ist ebenso erreichbar.