Karlsruher Institut für Technologie

Institut für Theoretische Informatik
T.T.-Prof. Dr. Thomas Bläsius
https://scale.iti.kit.edu

Übungsblatt 01

Algorithmen I - Sommersemester 2024

Abgabe im ILIAS bis 26.04.2024, 18:00 Uhr

Die Abgabe erfolgt alleine oder zu zweit als eine PDF-Datei über das Übungsmodul in der Gruppe eures Tutoriums im ILIAS. Beschriftet die Abgabe deutlich mit Matrikelnummer(n) und Name(n). Bei Abgaben zu zweit reicht es, wenn eine Person die Abgabe im ILIAS hochlädt.

- Achtet bei handschriftlichen Abgaben auf Lesbarkeit.
- Achtet darauf, ob Algorithmen in Worten oder Pseudocode beschrieben werden sollen.
- Es werden immer asymptotisch möglichst effiziente Algorithmen erwartet, wenn nicht anders angegeben.
- Wenn Korrektheits- oder Laufzeitanalysen gefordert sind, behandelt diese separat von der Algorithmenbeschreibung.

Gesamtpunkte: 20

Aufgabe 1 - O(rdnungs-)-Kalkül (3 Punkte)

Ordne die folgenden Funktionen so an, dass $f \in \mathcal{O}(g)$ gilt, genau dann, wenn f links von g eingeordnet ist. Begründungen sind nicht notwendig.

(i) $2^{n \cdot \log_2(n)}$

(ii) $n!$

(iii) $\log_2(\sqrt{n})$

(iv) $\sqrt{\log_3(n)}$

(v) $\frac{n^2}{\log_2(n)}$

(vi) $\pi^{\ln(n)}$

Hinweis: Achte auf die Basen der Logarithmen.

Aufgabe 2 - Beweisen oder nicht beweisen, das ist hier die Frage (5 Punkte)

Beweise oder widerlege die folgenden Aussagen. Gib hierfür jeweils n_0 und c aus der formalen Definition an und zeige, dass mit diesen Werten die Ungleichung der Definition gilt. Alternativ, nutze die alternative Definition über Grenzwerte, um eine Aussage zu widerlegen.
Beispiel: Für $f(n) \in \mathcal{O}(g(n))$ finde n_0 und c mit $\forall n > n_0 : f(n) \leq c \cdot g(n)$ und zeige, das dies für deine Werte gilt.

a) $7 \cdot n^{42} + 4 \cdot n^{11} + 13 \cdot n^2 - 12 \cdot n + 7 \in \mathcal{O}(n^{42})$ (1 Punkt)

b) $\log(n) \cdot \log(n) \in \mathcal{O}(\log(\log(n)))$ (1 Punkt)

c) $2024 \cdot n^2 \in \mathcal{O}(3^n)$ (1 Punkt)

d) $\sqrt[5]{n^7} \in \Omega(\log^7(n))$ (2 Punkte)

Aufgabe 3 - Die Mär vom großen und bösen ~~Wolf~~ Summenterm (4 Punkte)

In dieser Aufgabe sollst du einige wichtige Techniken zum Abschätzen von Summen im $\mathcal{O}$-Kalkül kennen lernen.
Zeige für jede der folgenden Summen, dass diese in der gegebenen Komplexitätsklasse liegt.

a) $f_1(n) := \sum_{i=0}^{\sqrt{n}} \frac{2^8}{4^i} \in \Theta(1)$ (1 Punkt)

b) $f_2(n) := \sum_{i=0}^{\log_3 n} \frac{3^i}{2^8} \in \Theta(n)$ (1 Punkt)

c) $f_3(n) := \sum_{i=0}^{\sqrt{n}} \frac{i^2}{2^8} \in \mathcal{O}(n \cdot \sqrt{n})$ (1 Punkt)

d) $f_4(n) := \sum_{i=\sqrt{n}}^{2\cdot\sqrt{n}} \left(\frac{n}{\log_6(2^n \cdot 3^n)}\right)^i \in \Theta(\sqrt{n})$ (1 Punkt)

Aufgabe 4 - Hols Holz! (8 Punkte)

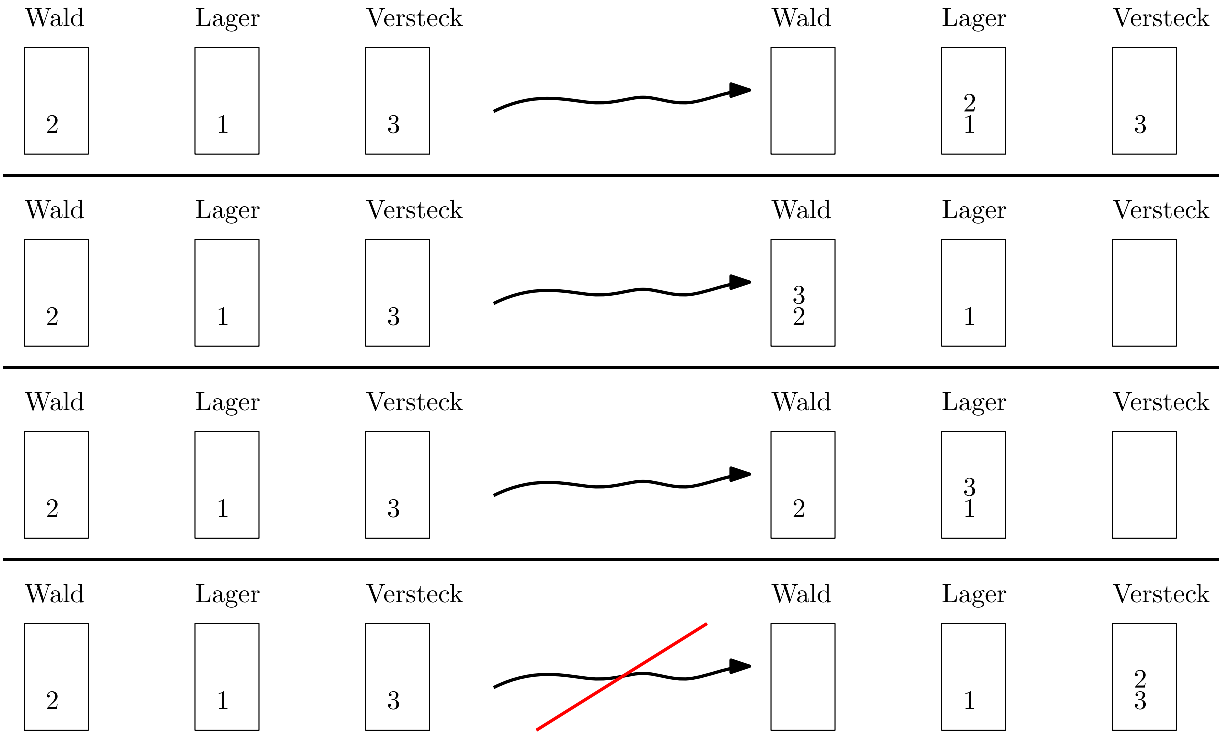
Nach mehreren gescheiterten Versuchen, die Weltherrschaft zu übernehmen, nutzte Dr. Meta die vorlesungsfreie Zeit, um eine neue Strategie zu entwickeln. Während er die Details seines Plans noch geheim hält, hat er enthüllt, dass er für die Umsetzung erhebliche Mengen an Holz in seinem Versteck benötigt. Dafür hat er seine treuen Biber beauftragt, das benötigte Holz aus dem Wald zu seinem Versteck zu transportieren.

Jeder der n Baumstämme ist mit einer eindeutigen Qualitätszahl von 1 bis n markiert. Um die Qualität des Holzes zu bewahren, muss jeder Holzstapel stets so sortiert sein, dass die Stämme mit niedrigerer Qualitätszahl unten und jene mit höherer oben liegen.

Das bedeutet, dass ein Baumstamm immer auf einen anderen gelegt werden muss. Die Holzfällbiber haben das gefällte Holz bereits entsprechend sortiert und gestapelt.

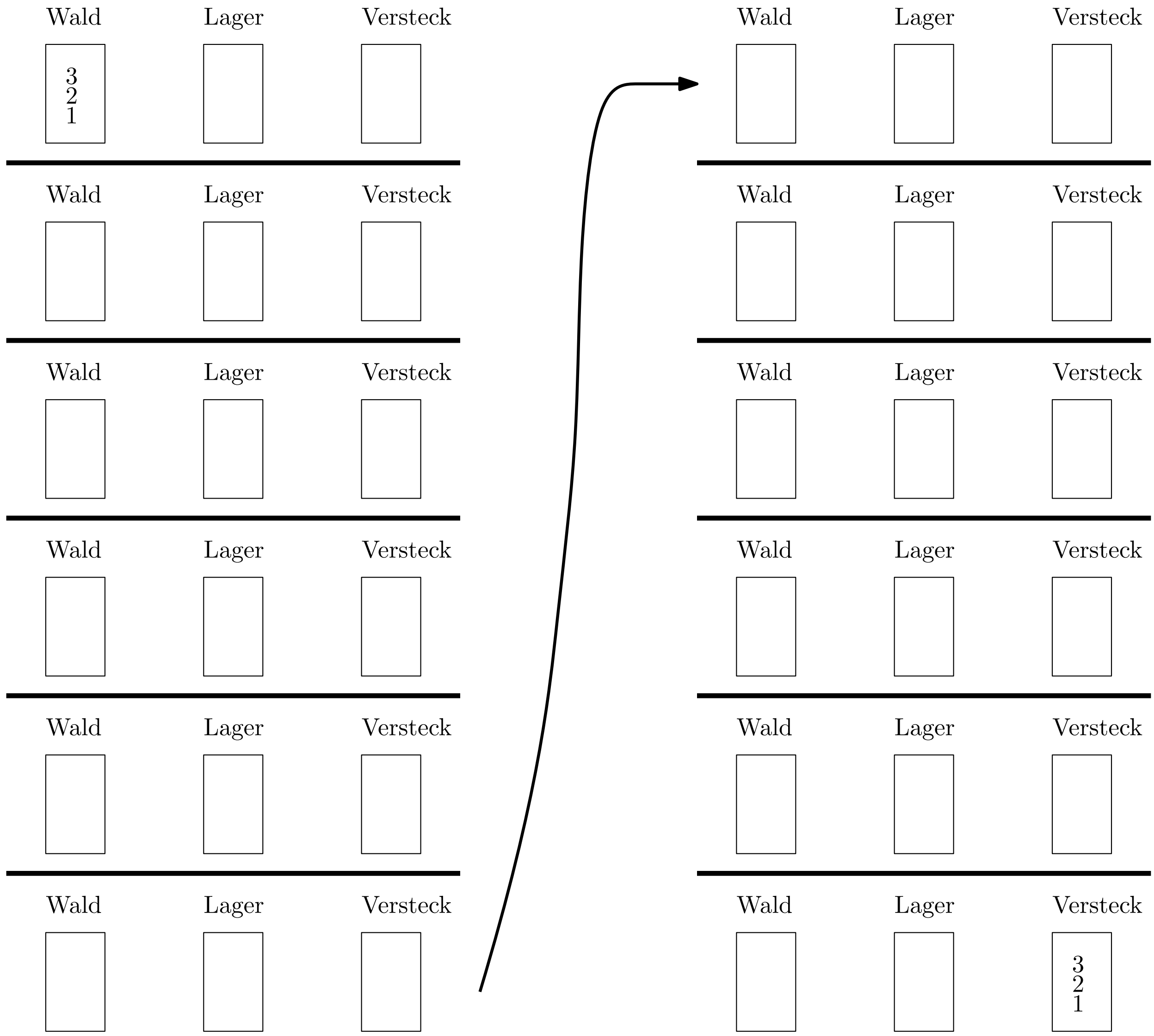
Dr. Meta stellt den Bibern für diese Aufgabe nur einen alten Lastkraftwagen und ein kleines Zwischenlager zur Verfügung. Der LKW kann pro Fahrt nur einen Baumstamm befördern. Sowohl im Wald, im Geheimversteck, als auch im Zwischenlager kann nur ein einziger Stapel Holz platziert werden. Es kann jeweils nur der oberste Baumstamm eines Stapels transportiert werden, welcher dann auf den Stapel am Zielort gelegt wird.

Die Aufgabe besteht darin, die Biber dabei zu unterstützen, das Holz mit möglichst wenigen Fahrten in das Versteck zu bringen. Dabei zählen nur die Fahrten, bei denen tatsächlich ein Baumstamm transportiert wird.



Ein gültiger Zustand, sowie alle durch gültige Fahrten erreichbaren Zustände. Außerdem ein Zustand der nicht durch eine gültige Fahrt erreicht werden kann. Jeder Baumstamm wird durch seine Qualitätszahl repräsentiert.

a) Gib für $n = 3$ eine Folge von Fahrten an, die alle Baumstämme mit minimaler Fahrtenzahl ins Versteck bewegt. Trage dafür den Zustand nach jeder Fahrt für alle drei Lagerorte (Wald (W), (Zwischen-)Lager (L), Versteck (V)) in der untenstehenden Darstellung ein. (1.5 Punkte) Hinweis: Es sind mehr Zustände als nötig angegeben. Streiche die nicht benötigten durch.



Darstellung des Start- und des Endzustandes der geforderten Folge von Fahrten, so wie freie Felder für Zwischenzustände.

b) Angenommen die Biber könnten die obersten $n-1$ Baumstämme eines Stapels auf einmal bewegen. Gib nun für allgemeine und ausreichend große n an, wie du das Problem mit minimaler Fahrtzahl lösen kannst.
Du musst nicht zeigen, dass deine Lösung minimal ist. (1.5 Punkte)

c) Gib einen rekursiven Algorithmus, der das ursprüngliche Problem mit möglichst wenigen Fahrten löst, in Pseudocode an. Du kannst dafür die Erkenntnisse aus b) in abgewandelter Form und die Methode FAHREN(S : Lagerort, Z : Lagerort), die den obersten Baumstamm von S nach Z fährt, verwenden.
Deine Lösung soll hierbei die Methodensignatur HOLSHOLZ(S : Lagerort, Z : La-

gerort, H : Lagerort, $n : \mathbb{N}_+$) haben, wobei die Methode n Baumstämme von S nach Z unter Nutzung von H als Ablage transportiert. (2 Punkte)

d) Erkläre kurz warum es bei deinem Algorithmus zu keinen Problemen kommt, also warum nie ein Baumstamm auf einem qualitativ hochwertigeren gestapelt wird. (1 Punkt)

e) Gebe die von deinem Algorithmus benötigte Anzahl an Fahrten mit einem Baumstamm an. Stelle dafür zuerst eine Rekurrenzgleichung auf und analysiere diese anschließend. (2 Punkte)

Aufgabe 5 - Zum Knobeln (0+1 Punkte)

Zeige, die zu Aufgabe 3 c) gehörende untere Schranke.

Zeige also $\sum\limits_{i=0}^{\sqrt{n}} \frac{i^2}{2^8} \in \Omega(n \cdot \sqrt{n})$. (1 Punkt)