Algorithmen und Datenstrukturen 1

ALGO1 · SoSe-2023 · `tcs.uni-frankfurt.de/algo1/` · 2023-07-06 · *ed6968a*

Gierige Algorithmen (Woche 14)

Eigenständige Vorbereitung:
Lies 📖 E Kapitel 4 und schau die 📺 Videos der Woche.

Zeichenlegende:

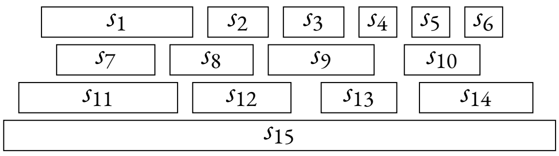
- ✏️ Schriftliche Aufgabe, die du fristgerecht in Moodle abgibst. In der Klausur wirst du alle Aufgaben schriftlich bearbeiten, daher ist das Feedback der Tutoren wichtig, damit du deine Schreibfähigkeiten verbessern kannst.
- 👍 Diese Art von Aufgabe musst du sicher können, um die Klausur zu bestehen.
- 🔑 Diese Art von Aufgabe musst du weitgehend können, um die Klausur zu bestehen.
- 🔭 Diese Art von Aufgabe musst du können, um eine gute Note zu erhalten.
- 🌈 Diese Aufgabe ist als Knobelspaß gedacht, der das algorithmische Verständnis vertieft.

Aufgabe 14.1 (Per Hand laufen lassen).

a) 👍 Verwende den Algorithmus zur Huffman-Codierung aus der Vorlesung, um auf der folgenden Häufigkeitstabelle für das Alphabet $\{\texttt{a}, \dots, \texttt{g}\}$ einen optimalen präfixfreien Binärcode zu berechnen; zeichne dabei den Code-Baum, und gib am Schluss den berechneten optimalen binären Code an:

a	b	c	d	e	f	g
3	3	1	1	7	2	4

b) 👍 Verwende den gierigen Algorithmus aus der Vorlesung, um auf der folgenden Eingabe einen maximalen konfliktfreien Stundenplan $\{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$ zu bestimmen:



c) 👍 Verwende den gierigen Algorithmus aus der Vorlesung, um für die folgenden Präferenzen ein stabiles Matching zu bestimmen:

x	y	z	w
A	C	A	B
C	D	D	C
D	A	B	D
B	B	C	A

A	B	C	D
y	x	y	z
x	z	w	w
w	w	z	y
z	y	x	x

Hierbei sollen A, B, C und D Angebote machen, welche von x, y, z und w angenommen oder abgelehnt werden können.

Aufgabe 14.2 (Stabiles Matching 🔑). Konstruiere Präferenzen für Ärzt:innen und Krankenhäuser, sodass der Gale-Shapley Algorithmus aus der Vorlesung auf derselben Eingabe zwei verschiedene stabile Matchings konstruiert, abhängig davon, ob die Krankenhäuser die Angebote machen oder ob die Ärzt:innen die Angebote machen.

Aufgabe 14.3 (Dateien auf Band). Auf einem Magnetband sind n Dateien gespeichert. Die Längen der Dateien sind durch ein Feld $L[1..n]$ gegeben, das heißt, die i-te auf dem Band gespeicherte Datei hat Länge $L[i]$. Um Datei k zu lesen, muss das Band alle Dateien von 1 bis k lesen.

a) 👍 Betrachte ein Magnetband, auf dem sechs Dateien mit $L = [8, 3, 1, 6, 4, 2]$ gespeichert sind. Zum Beispiel hat Datei 1 die Länge 8. Wie hoch sind die Kosten, um Datei 4 zu lesen?
b) 👍 Wie hoch sind die erwarteten Kosten bei a), wenn ein sechsseitiger Würfel geworfen wird und die damit indizierte Datei gelesen wird?
c) 🔑 Wenn jede Datei k mit Wahrscheinlichkeit $1/n$ angefragt wird, sortieren wir die Dateien am besten der Größe nach. Jetzt aber sind manche Dateien beliebter als andere: Datei k wird mit Wahrscheinlichkeit $F[k]$ angefragt. Das heißt, $F[1..n]$ ist eine Wahrscheinlichkeitsverteilung über $\{1, \ldots, n\}$ und die erwarteten Kosten sind

$$\sum_{k=1}^{n} \sum_{i=1}^{k} F[k] \cdot L[i] \,.$$

In welcher Reihenfolge müssen wir die Dateien jetzt sortieren, damit die erwarteten Kosten so klein wie möglich sind? Begründe deine Antwort.

Aufgabe 14.4 (Scheduling). Gegeben sind Startzeiten $S[1..n]$ und Endzeiten $F[1..n]$ von n Vorlesungen.

a) 👍 Beschreibe mathematisch (also durch eine logische Formel, die S und F benutzt), was es heißt, dass die Zeiten von Vorlesung i und j sich überlappen.
b) 🔑 Professorin S. Chedule hat folgende Idee, um den gierigen Algorithmus für das Scheduling zu vereinfachen: Anstatt nach den Endzeiten, sortieren wir die Vorlesung nach den Startzeiten, und nehmen also immer die erste Vorlesung in den Stundenplan auf, die als Nächstes startet und keinen Konflikt verursacht. Finde ein Gegenbeispiel, in dem dieser Algorithmus fehlschlägt.

Aufgabe 14.5 (Antipathien). In der Algorithmen-Bank gab es während der Betriebsfeier viele Streitigkeiten zwischen den Angestellten. Daher wurde nach der Betriebsfeier ein Antipathie-Graph $G = (V, E)$ erstellt, in dem jede:r Angestellte durch einen Knoten der Menge $V = \{v_1, \ldots, v_n\}$ dargestellt wird. Eine ungerichtete Kante zwischen zwei Knoten bedeutet, dass sich diese beiden Angestellten nicht leiden können.

Nun möchte man einen neuen Standort eröffnen. Um das Konkurrenzdenken zwischen den Standorten zu verstärken, möchte man die Angestellten in zwei Gruppen aufteilen, sodass die Anzahl an Antipathien (Kanten) zwischen den Gruppen maximal ist.

Zur Bestimmung dieser Gruppen schlägt der Angestellte Aglot den folgenden gierigen Algorithmus vor:

1. Weise v_1 der Menge A zu.
2. Für $i = 2, \ldots, n$:
   - Seien a_i und b_i die Anzahlen der Nachbarn von v_i die bereits in A oder B sind
   - Weise v_i der Menge A zu, wenn $a_i \leq b_i$ sonst weise v_i der Menge B zu

Zeige oder **widerlege** folgende Aussagen:

a) 🔑 Der gierige Algorithmus arbeitet optimal.
b) 🔑 Der gierige Algorithmus kann mit Laufzeit $O(n + m)$ implementiert werden, sofern G in Adjazenzlistendarstellung gegeben ist.
c) 🔭 Der gierige Algorithmus findet eine Gruppenkonstellation mit mindestens $\frac{|E|}{2}$ Kanten zwischen den Gruppen.

Aufgabe 14.6 (Balance II 🔭). Wir erinnern uns zurück an die ✏️-Aufgabe von Woche 4 ("Balance"). Diesmal betrachten wir Strings mit nur einem Klammertyp. Ein String w von Klammern (und) ist *balanciert*, wenn er eine der folgenden Bedingungen erfüllt:

- w ist der leere String
- $w =$ (x) für einen balancierten String x
- $w = xy$ für balancierte Strings x und y.

Beispielsweise ist der String (())((()())) balanciert, während weder ()), noch ())(balanciert sind.

Beschreibe einen gierigen Algorithmus mit Laufzeit $O(n)$, welcher die Länge eines längsten balancierten Teilstrings in einem String der Länge n berechnet. Dein Algorithmus soll ein Feld $w[1..n]$ entgegennehmen, wobei für jeden Index i entweder $w[i] =$ (oder $w[i] =$) gilt.

Hinweis: Welche elementare Datenstruktur wurde damals in der ✏️-Aufgabe verwendet?

Aufgabe 14.7 (Schuldenerlass 🌈). Aglot aus der Algorithmen-Bank schuldet seiner Kollegin Aifos 500.000 Traumdollar. Zum Erlassen seiner Schulden bietet sie Aglot folgenden Deal an:

In einem blickdichten Beutel befindet sich eine Murmel, die entweder rot oder blau ist. Aglot kennt die Farbe der Murmel nicht, er weiß lediglich, dass beide Farben mit gleicher Wahrscheinlichkeit auftreten. Nun legt Aifos eine rote Murmel in den Beutel und schüttelt diesen einmal durch. Anschließend zieht sie rein zufällig eine Murmel aus dem Beutel und teilt Aglot die Farbe der gezogenen Murmel mit. Nun muss Aglot die Farbe der im Beutel verbliebenen Murmel erraten. Liegt er richtig, erlässt Aifos seine Schulden, andernfalls werden die Schulden verdoppelt.

Selbstverständlich nimmt Aglot den Deal an. Aifos legt die rote Murmel in den Beutel, mischt einmal durch, und zieht anschließend eine rote Murmel aus dem Beutel. Welche Farbe sollte Aglot nun raten?