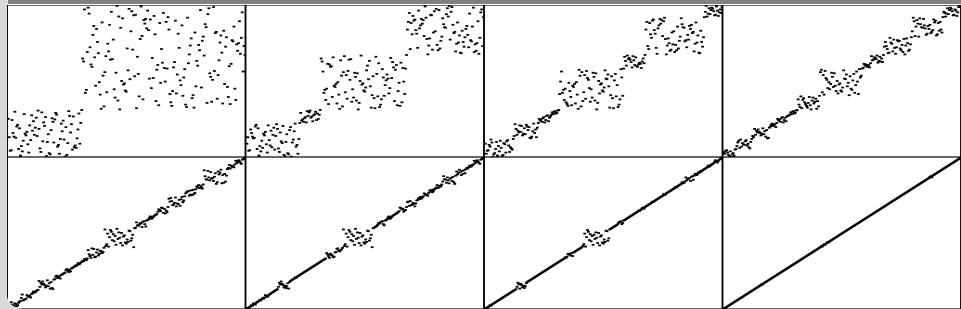


# Tutorium 7: Übungsaufgaben

Holger Ebhart | 3. Juni 2015

TUTORIUM ZUR VORLESUNG ALGORITHMEN I IM SS15



## 1 6. Übungsblatt

## 2 Übungsaufgaben

- Motivation
- O-Kalkül
- Felder, Folgen und Listen
- Hashing
- Sortieren
- Heaps

## 3 Mittsemesterklausur

# 6. Übungsblatt

Gegeben sind zwei Matrizen  $A, B \in \mathbb{N}^{n \times n}$ . Geben Sie Pseudocode an, der das Produkt von  $A$  und  $B$  berechnet und geben Sie die Laufzeit im  $\mathcal{O}$ -Kalkül an. Die Struktur *Matrix* bietet folgende Funktionen:  $M.get(int\ x, int\ y)$  gibt  $m_{x,y}$  zurück,  $M.set(int\ x, int\ y, int\ v)$  setzt  $m_{x,y}$  auf den Wert  $v$ .  $M.size()$  gibt die Spalten/Zeilenanzahl  $n$  der Matrix zurück. Diese Operationen können in  $\mathcal{O}(1)$  durchgeführt werden.

Zeigen Sie:

a)  $\mathcal{O}(n^{n!}) \in \mathcal{O}(n^{n^n})$

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

Finden Sie eine geschlossene Form für folgende Rekurrenz:

$$T(n) = \begin{cases} 1, & \text{falls } n = 1 \\ 4 \cdot T(\lceil \frac{n}{2} \rceil) + 6 \cdot n^2, & \text{falls } n \geq 2 \end{cases}$$

Sie dürfen annehmen, dass  $n$  eine Zweierpotenz ist.

Hinweis: Substitution hilft!

Zeigen Sie:

a)  $\mathcal{O}(n^{n!}) \in \mathcal{O}(n^{n^n})$

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

Finden Sie eine geschlossene Form für folgende Rekurrenz:

$$T(n) = \begin{cases} 1, & \text{falls } n = 1 \\ 4 \cdot T(\lceil \frac{n}{2} \rceil) + 6 \cdot n^2, & \text{falls } n \geq 2 \end{cases}$$

Sie dürfen annehmen, dass  $n$  eine Zweierpotenz ist.

Hinweis: Substitution hilft!

Lösen Sie folgende Rekurrenzen mit dem Mastertheorem:

a)  $T(1) = 1, T(n) = \frac{n}{6} + 4T(\frac{n}{6}), n = 6^k, k \in \mathbb{N}$

b)  $T(1) = 33, S(n) = 6T(\frac{n}{3}) + 89n, n = 3^k, k \in \mathbb{N}$

- a) Warum verwendet man beim verketteten Hashing einfach- / doppelt-verkettete Listen? Nennen Sie mindestens zwei Gründe dafür.
- b) Was ist ein Sentinel?, wofür verwendet man es? und welche Vor- und Nachteile hat es?
- c) Wie lautet die Datenstrukturinvariante für doppelt verkettete Listen aus der Vorlesung?



# Aufgabe 5

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus  $\mathbb{N}$ . Gegeben sei eine Datei/Folge von  $n$  Paaren der Form  $(MitarbeiterID, Urlaubstag) \in \mathbb{N} \times \{1, \dots, 365\}$ . Die Datei  $D$  beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält  $D$  genau ein entsprechendes Paar. Achtung: Die IDs können beliebig groß werden.

- a) Geben Sie einen Algorithmus an der in erwartet  $\mathcal{O}(n)$  Zeit die IDs der -Mitarbeiter ausgibt die in der ersten Jahreshälfte mindestens 30 Tage Urlaub genommen haben. Geben Sie dabei keine ID doppelt aus.
- b) Begründen Sie die Laufzeit.
- c) Geben Sie einen Algorithmus an der anhand von  $D$  in deterministisch  $\mathcal{O}(n)$  Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl an Mitarbeitern Urlaub genommen hat.
- d) Begründen Sie die Laufzeit.

# Aufgabe 5

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus  $\mathbb{N}$ . Gegeben sei eine Datei/Folge von  $n$  Paaren der Form  $(MitarbeiterID, Urlaubstag) \in \mathbb{N} \times \{1, \dots, 365\}$ . Die Datei  $D$  beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält  $D$  genau ein entsprechendes Paar. Achtung: Die IDs können beliebig groß werden.

- a) Geben Sie einen Algorithmus an der in erwartet  $\mathcal{O}(n)$  Zeit die IDs der -Mitarbeiter ausgibt die in der ersten Jahreshälfte mindestens 30 Tage Urlaub genommen haben. Geben Sie dabei keine ID doppelt aus.
- b) Begründen Sie die Laufzeit.
- c) Geben Sie einen Algorithmus an der anhand von  $D$  in deterministisch  $\mathcal{O}(n)$  Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl an Mitarbeitern Urlaub genommen hat.
- d) Begründen Sie die Laufzeit.

# Aufgabe 5

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus  $\mathbb{N}$ . Gegeben sei eine Datei/Folge von  $n$  Paaren der Form  $(\text{MitarbeiterID}, \text{Urlaubstag}) \in \mathbb{N} \times \{1, \dots, 365\}$ . Die Datei D beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält D genau ein entsprechendes Paar. Achtung: Die IDs können beliebig groß werden.

- a) Geben Sie einen Algorithmus an der in erwartet  $\mathcal{O}(n)$  Zeit die IDs der -Mitarbeiter ausgibt die in der ersten Jahreshälfte mindestens 30 Tage Urlaub genommen haben. Geben Sie dabei keine ID doppelt aus.
- b) Begründen Sie die Laufzeit.
- c) Geben Sie einen Algorithmus an der anhand von D in deterministisch  $\mathcal{O}(n)$  Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl an Mitarbeitern Urlaub genommen hat.
- d) Begründen Sie die Laufzeit.

# Aufgabe 5

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus  $\mathbb{N}$ . Gegeben sei eine Datei/Folge von  $n$  Paaren der Form  $(MitarbeiterID, Urlaubstag) \in \mathbb{N} \times \{1, \dots, 365\}$ . Die Datei  $D$  beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält  $D$  genau ein entsprechendes Paar. Achtung: Die IDs können beliebig groß werden.

- Geben Sie einen Algorithmus an der in erwartet  $\mathcal{O}(n)$  Zeit die IDs der -Mitarbeiter ausgibt die in der ersten Jahreshälfte mindestens 30 Tage Urlaub genommen haben. Geben Sie dabei keine ID doppelt aus.
- Begründen Sie die Laufzeit.
- Geben Sie einen Algorithmus an der anhand von  $D$  in deterministisch  $\mathcal{O}(n)$  Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl an Mitarbeitern Urlaub genommen hat.

d) Begründen Sie die Laufzeit.

# Aufgabe 5

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus  $\mathbb{N}$ . Gegeben sei eine Datei/Folge von  $n$  Paaren der Form  $(MitarbeiterID, Urlaubstag) \in \mathbb{N} \times \{1, \dots, 365\}$ . Die Datei D beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält D genau ein entsprechendes Paar. Achtung: Die IDs können beliebig groß werden.

- Geben Sie einen Algorithmus an der in erwartet  $\mathcal{O}(n)$  Zeit die IDs der -Mitarbeiter ausgibt die in der ersten Jahreshälfte mindestens 30 Tage Urlaub genommen haben. Geben Sie dabei keine ID doppelt aus.
- Begründen Sie die Laufzeit.
- Geben Sie einen Algorithmus an der anhand von D in deterministisch  $\mathcal{O}(n)$  Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl an Mitarbeitern Urlaub genommen hat.
- Begründen Sie die Laufzeit.

Gegeben Sei eine Hashtabelle mit Puffergröße 3 und eine mit einem cyclic Array sowie die Hashfunktion  $h(x) = x \text{DIV} 10$ . Fügen Sie folgende Elemente mit Hashing mit linearer Suche in die Tabellen ein:

81, 34, 87, 17, 80, 35, 20, 86, 18

Sortieren durch Auswählen (Selection Sort) ist ein weiterer vergleichsbasierter Sortier-Algorithmus. In jedem Schritt sucht er das kleinste Element der Sequenz und verschiebt es an den Anfang. Danach wird der Rest der Sequenz um eins nach rechts verschoben. Gehen Sie im Folgenden von einer Array-Implementierung aus.

- a) Konstruieren Sie eine Worst-Case-Eingabe für  $n=6$ .
- b) Geben Sie die Laufzeit für eine Sequenz von  $n$  Elementen im schlimmsten Fall an (genau und in Landau-Notation).
- c) Welchem Sortier-Algorithmus ähnelt Selection Sort?

# Aufgabe 8

Sortieren Sie folgende Menge mit Radixsort:

$Z = (111, 76, 223, 567, 349, 496, 201, 872, 3)$

Geben Sie alle Zwischenschritte an. Welche Laufzeit hat Radixsort?



Es sei folgendes Array gegeben:

$A := [3, 7, 9, 6, 13, 10, 11, 8, 7, 14, 17, 16, 15]$

- a) Geben Sie die Invariante für Min-Heaps an.
- b) Geben Sie den binären Baum an den A aufspannt.
- c) Ist a ein Min-Heap? Falls ja begründen Sie warum, falls nein so geben Sie an wo der Fehler liegt und korrigieren Sie ihn.
- d) Führen Sie *deleteMin()* auf dem Heap aus c) aus.
- e) Bauen Sie aus A einen Max-Heap in  $\mathcal{O}(n)$  Zeit.

Es sei folgendes Array gegeben:

$A := [3, 7, 9, 6, 13, 10, 11, 8, 7, 14, 17, 16, 15]$

- a) Geben Sie die Invariante für Min-Heaps an.
- b) Geben Sie den binären Baum an den A aufspannt.
- c) Ist a ein Min-Heap? Falls ja begründen Sie warum, falls nein so geben Sie an wo der Fehler liegt und korrigieren Sie ihn.
- d) Führen Sie *deleteMin()* auf dem Heap aus c) aus.
- e) Bauen Sie aus A einen Max-Heap in  $\mathcal{O}(n)$  Zeit.

Es sei folgendes Array gegeben:

$A := [3, 7, 9, 6, 13, 10, 11, 8, 7, 14, 17, 16, 15]$

- a) Geben Sie die Invariante für Min-Heaps an.
- b) Geben Sie den binären Baum an den A aufspannt.
- c) Ist a ein Min-Heap? Falls ja begründen Sie warum, falls nein so geben Sie an wo der Fehler liegt und korrigieren Sie ihn.
- d) Führen Sie *deleteMin()* auf dem Heap aus c) aus.
- e) Bauen Sie aus A einen Max-Heap in  $\mathcal{O}(n)$  Zeit.

# Aufgabe 10

Geben Sie eine Implementierung für *deleteMax* für einen Max-Heap an (Pseudocode).

- **zeigen, angeben, beweisen, nachweisen** → formaler Beweis erforderlich
- Aufgabenstellung genau durchlesen
- Antwortaufwand in Verhältnis zu Punkten setzen
- Standardaufgaben **müssen** schnell von der Hand gehen
- Rückgabe im Tutorium

- **zeigen, angeben, beweisen, nachweisen** → formaler Beweis erforderlich
- Aufgabenstellung genau durchlesen
- Antwortaufwand in Verhältnis zu Punkten setzen
- Standardaufgaben **müssen** schnell von der Hand gehen
- Rückgabe im Tutorium

- **zeigen, angeben, beweisen, nachweisen** → formaler Beweis erforderlich
- Aufgabenstellung genau durchlesen
- Antwortaufwand in Verhältnis zu Punkten setzen
- Standardaufgaben **müssen** schnell von der Hand gehen
- Rückgabe im Tutorium

- **zeigen, angeben, beweisen, nachweisen** → formaler Beweis erforderlich
- Aufgabenstellung genau durchlesen
- Antwortaufwand in Verhältnis zu Punkten setzen
- Standardaufgaben **müssen** schnell von der Hand gehen
- Rückgabe im Tutorium



- **zeigen, angeben, beweisen, nachweisen** → formaler Beweis erforderlich
- Aufgabenstellung genau durchlesen
- Antwortaufwand in Verhältnis zu Punkten setzen
- Standardaufgaben **müssen** schnell von der Hand gehen
- Rückgabe im Tutorium

**Vielen Dank für eure Aufmerksamkeit!**  
**Bis zum nächsten Mal.**

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

stackoverflow.com