

DAP2 Praktikum – Blatt 6

Abgabe: KW 21

Studienleistung

- Zum Bestehen des Praktikums muss jeder Teilnehmer*innen die folgenden Leistungen erbringen:
 - Es müssen mindestens 50 Prozent der Punkte in den Kurzaufgaben erreicht werden.
 - Es müssen mindestens 50 Prozent der Punkte in den Langaufgaben erreicht werden.
- Im Krankheitsfall kann ein Testat bei Vorlage eines Attests in der folgenden Woche nachgeholt werden.
- Wenn ein Praktikumstermin auf einen Feiertag fällt, müssen Sie sich an einem beliebigen anderen Praktikumstermin in der gleichen Woche testieren lassen.
- **Hinweis:** Notieren Sie sich Ihre Punkte nach jedem Testat! Dies dient der eigenen Kontrolle. (Ihr Punktestand kann Ihnen während des Semesters nicht genannt werden.)

Wichtige Information (im Moodle verfügbar)

- Beachten Sie die Erklärung des **Ablaufs (Blatt A)**.
- Beachten Sie die **Regeln und Hinweise (Blatt R)** in der aktuellsten Version!
- Beachten Sie die **Hilfestellungen (Blatt H)** in der aktuellsten Version!

Languaufgabe 6.1: Counting-Sort

(4 Punkte)

Frequenzen Zählen. Schreiben Sie ein Programm, das eine Ganzzahl n und eine Liste A von Ganzzahlen a_0, a_1, \dots, a_{n-1} via Standard-In bekommt, und die Anzahl der Vorkommen jeder Zahl in der Liste A bestimmt. Dies soll in folgenden Schritten umgesetzt werden:

- Zuerst werden die Zahlen a_0, a_1, \dots, a_{n-1} in ein Array `data` eingelesen.
- Anschließend werden Maximum und Minimum des Arrays bestimmt. Implementieren Sie dafür die folgenden zwei Methoden:

```
public static int getMin(int[] data)
public static int getMax(int[] data) .
```

- Schreiben Sie eine Methode `count`, die ein Array `data` von Ganzzahlen sowie zwei zusätzliche Schlüssel `min` und `max` erhält. Ausgabe der Methode ist ein Array C der Länge $\text{max} - \text{min} + 1$, sodass $C[i]$ die Anzahl der Vorkommen von $i + \text{min}$ in der Liste A ist.

```
public static int[] count(int[] data, int min, int max)
```

Für die volle Punktzahl sollte Ihr Programm (abgesehen vom Lesen der Eingabe und Schreiben der Ausgabe) **nicht mehr als $\mathcal{O}(n)$ Rechenschritte benötigen.**

Counting-Sort. In der Vorlesung haben Sie *CountingSort* kennengelernt.

- Für dieses Problem erhalten sie erneut eine Ganzzahl n und eine Liste von Ganzzahlen a_0, a_1, \dots, a_{n-1} via Standard-In. Implementieren Sie den Algorithmus *Counting-Sort* aus der Vorlesung. Dieser Algorithmus soll die Liste der Ganzzahlen in **absteigende Reihenfolge** bringen. Dies geschieht mit einer Methode

```
public static int [] countingSort(int[] data).
```

Zum Zählen der Frequenzen dürfen Sie ein Hilfsarray der Länge $\text{max} - \text{min} + 1$ verwenden (und weitere Hilfsarrays sind nicht erlaubt). Dabei sind `max` und `min` der maximale bzw. minimale Wert in der gegebenen List von Ganzzahlen.

Ausgabe der Methode ist eine absteigend sortierten Liste B der Länge n von Ganzzahlen von `data`.

Für die volle Punktzahl sollte Ihr Programm (abgesehen vom Lesen der Eingabe und Schreiben der Ausgabe) **nicht mehr als $\mathcal{O}(n + \text{max} - \text{min})$ Rechenschritte benötigen.**

- Schreiben Sie eine geeignete `main`-Methode, welche das Frequenzarray und die absteigend sortierte Liste ausgibt. Wie immer können Sie davon ausgehen, dass die Eingabe korrekt ist und müssen den Code nicht um diese Eingabe Robust gestalten.

Beispielausgaben des Programms:

```
java B6A1.java <Enter>
6 1 2 2 3 4 4
Input Array: [1, 2, 2, 3, 4, 4]
The minimum value: 1
The maximum value: 4
Frequencies:  [1, 2, 1, 2]
After sorting: [4, 4, 3, 2, 2, 1]
```

(4 Punkte)

$$1661914940 = 99 \cdot 256^3 + 14 \cdot 256^2 + 207 \cdot 256^1 + 60 \cdot 256^0.$$

After sorting: [692092, 322434, 47676, 22223, 4999, 2777, 122]