

6. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2020/21)

Aufgabe 1 – MinHeaps

Gegeben sei eine Menge $S \subseteq \mathbb{Z}$ von n Zahlen und eine Zahl $z \in \mathbb{Z}$. Ein Algorithmus soll alle Zahlen in S ausgeben, die echt kleiner als z sind.

- a) Zeigen Sie: jeder Algorithmus für dieses Problem braucht im Worst-Case $\Omega(n)$ Zeit. **2 Punkte**
- b) Geben Sie in Pseudocode einen Algorithmus an, der dieses Problem auf einem MinHeap S löst. Der Algorithmus soll echt schneller als linear in n sein, wenn die Mächtigkeit k der Ergebnismenge klein ist (das heißt $k \in o(n)$). Am Ende soll S immer noch ein MinHeap sein, der die gleichen Elemente wie am Anfang enthält. Hierfür können 3 Punkte erreicht werden. Es gibt den 4. Punkt, wenn die Laufzeit nicht von n , sondern nur von k abhängt. **4 Punkte**
- c) Geben Sie die Laufzeit Ihres Algorithmus aus Aufgabenteil b) in Abhängigkeit von n und k an. Begründen Sie Ihre Antwort. **2 Punkte**

Aufgabe 2 – Tödlicher Bocksbeutel

Vor langer Zeit lebte in Franken ein guter König, der in seinem Weinkeller tausend Bocksbeutel kostbaren Weines verwahrte. Eines Tages ertappten seine Wachen einen (bayerischen) Agenten, der gerade dabei war des Königs Wein zu vergiften. Zwar wusste man, dass der Agent es nur geschafft hatte eine einzige Flasche zu präparieren, allerdings war es ihm im Handgemenge bei seiner Festnahme gelungen, die Flasche zu den anderen zu stellen, so dass niemand wusste, welcher Bocksbeutel vergiftet war. Das Gift war tödlich; schon ein Tropfen des vergifteten Weines führte unweigerlich zum Tod. Dabei wirkte das Gift sehr langsam, nach einem Monat erlag man an seinen Folgen.

Da der König seinen Wein sehr liebte, wollte er das Ergebnis innerhalb eines Monats kennen. Leider stehen ihm dafür nur zehn mutige Vorkoster zur Verfügung. Zeigen Sie dem König, dass diese zehn Vorkoster ausreichen. **4 Punkte**

Aufgabe 3 – Shuffle

Gegeben sei der folgende Algorithmus, der ein Feld A mischt.

Shuffle(Feld A)

```
n1 = ⌊(A.length + 1)/2⌋  
n2 = A.length - n1  
L = neues Feld[1..n1]  
R = neues Feld[1..n2]  
L[1..n1] = A[1..n1]  
R[1..n2] = A[n1 + 1..A.length]  
i = j = 1  
for k = 1 to A.length do  
    b = Random(0,1)           // Liefert mit gleicher Wahrscheinlichkeit 0 oder 1.  
    if (b == 0 and i ≤ n1) or j > n2 then  
        A[k] = L[i]  
        i = i + 1  
    else  
        A[k] = R[j]  
        j = j + 1
```

- a) Beschreiben Sie in Worten, was der Algorithmus Shuffle bewirkt. Stellen Sie sich das Feld A dafür als Kartenstapel vor. **2 Punkte**
- b) Nach dem Aufruf von Shuffle(A) sei das Feld A = ⟨4, 1, 8, 5, 3, 2, 6, 7, 9⟩. Die Folge der Zufallszahlen, die bei der Ausführung an die Variable b übergeben wurden, sei ⟨0, 1, 1, 1, 0, 0, 1, 0, 1⟩.

Geben Sie A vor der Ausführung von Shuffle an. **1 Punkt**

- c) Kann man durch einmalige Anwendung von Shuffle jede Permutation der Zahlen in A erzeugen? Begründen Sie Ihre Antwort. **1 Punkt**

- d) Sei A ein Feld der Länge n, das sortiert war und nur durch einmalige Anwendung von Shuffle gemischt wurde. Geben Sie in Pseudocode einen vergleichsbasierten Algorithmus UnShuffle an, der A in $\Theta(n)$ Zeit rekonstruiert.

Begründen Sie, warum Ihr Algorithmus korrekt ist und die richtige Laufzeit hat.

Hinweis: Verwenden Sie den Median. **4 Punkte**

Bitte geben Sie Ihre Lösungen bis **Montag, 11. Januar 2021, 16:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit **PABS** gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.