

2. Präsenzübungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2018/19)

Aufgabe 1 – Probabilistische Laufzeiten

Gegeben sei folgender Algorithmus:

Algo1(int[] A):

n = A.length

for i = 1 to n do

```
┌   rnd = eine generierte Zahl zwischen 1 (inklusive) und n (inklusive)
├   if rnd ≤ i then
├       for j = i to n do
├           └ print A[j]
```

- Gebe der Zufallsgenerator (Zeile 2) stets n zurück. Wie viele Zahlen werden ausgegeben?
- Gebe der Zufallsgenerator (Zeile 2) stets 1 zurück. Wie viele Zahlen werden ausgegeben?
- Im Folgenden gebe der Zufallsgenerator jede Zahl zwischen 1 und n mit gleicher Wahrscheinlichkeit zurück. Definieren Sie eine Indikatorzufallsvariable X_i , die beschreibt ob in Iteration i die Zeile 4 ausgeführt wird. Berechnen Sie außerdem $E[X_i]$.
- Ermitteln Sie nun, wie viele Zahlen X der Algorithmus erwartet ausgibt.
- Geben Sie, basierend auf der vorherigen Teilaufgabe, die erwartete Laufzeit des Algorithmus in Θ -Notation an. Vergleichen Sie sie mit der Worst-Case- und der Best-Case-Laufzeit des Algorithmus.

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.

Aufgabe 3 – Sortieren in Linearzeit

Geben Sie in kommentiertem Pseudocode einen Algorithmus an, der ein Eingabefeld A der Länge n mit Wertebereich $\{1, \dots, n^2 - 1\}$ in $O(n)$ Zeit sortiert!

Tipp: Verwenden Sie Methoden aus der Vorlesung. Wandeln Sie jedoch zunächst die Zahlen in eine geeignete Darstellung um.

PABS

Aufgabe 4 – QuickSort

In dieser Aufgabe sollen Sie QuickSort in Java implementieren. Im folgenden sei stets A ein Feld aus Ganzzahlen und $n = A.length$. Außerdem gilt für $\ell, r \in \mathbb{N}$ stets $0 \leq \ell \leq r < n$ und Ihre Methoden sollen stets eine `IllegalArgumentException` werfen, wenn eine dieser Ungleichungen verletzt ist. Erstellen Sie dazu im Paket `sorting` die Klasse `public class QuickSort` und implementieren Sie dort die folgenden Methoden.

a) Zunächst benötigen Sie die beiden folgenden Subroutinen:

- `public static void swap(int[] A, int i, int j)`
- `public static int partition(int[] A, int l, int r)`

Die Methode `swap` tauscht im Feld A die Einträge i und j aus – nach Aufruf von `swap` steht in $A[i]$ der Wert, der vormals in $A[j]$ stand und umgekehrt. Die Laufzeit von `swap` soll in $O(1)$ sein.

Die Beschreibung der Methode `partition` finden Sie als Pseudocode im entsprechenden Foliensatz der Vorlesung. Die Laufzeit Ihrer Implementation soll in $O(n)$ sein.

b) Implementieren Sie nun, wie aus der Vorlesung bekannt, mit Hilfe Ihrer Subroutinen

- `public static void quickSort(int[] A, int l, int r)`

c) Implementieren Sie nun

- `public static int randomizedPartition(int[] A, int l, int r)`,

welche ein zufälliges Pivot-Element wählt, und nutzen Sie diese Implementation um

- `public static void randomizedQuickSort(int[] A, int l, int r)`

zu realisieren.

Hinweis: Um in Java zufällige Ganzzahlen aus dem Intervall $[0, n]$ zu erzeugen, verwenden Sie ein Random Objekt wie folgt:

```
Random rand = new Random();  
int k = rand.nextInt(n);
```

Diese Aufgaben werden eventuell gemeinsam in den Übungen am 18. und 19. Dezember 2018 gelöst. Sie brauchen Sie nicht vorher zu lösen und auch nicht abzugeben.