

2. Präsenzübungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2019/20)

Aufgabe 1 – Münzwurf

Eine faire Münze wird dreimal geworfen. Die Zufallsvariable X gibt an, wie oft die Münze mit dem Kopf nach oben landet.

- Wie groß ist $E[X^3]$?
- Wie groß ist $(E[X])^3$?

Aufgabe 2 – Shuffle

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

Shuffle(Feld A)

$n_1 = \lfloor (A.length + 1)/2 \rfloor$

$n_2 = A.length - n_1$

$L =$ neues Feld[1.. n_1]

$R =$ neues Feld[1.. n_2]

$L[1..n_1] = A[1..n_1]$

$R[1..n_2] = A[n_1 + 1..A.length]$

$i = j = 1$

for $k = 1$ **to** $A.length$ **do**

$b = \text{Random}(0,1)$ // Liefert mit gleicher Wahrscheinlichkeit 0 oder 1.

if ($b == 0$ **and** $i \leq n_1$) **or** $j > n_2$ **then**

$A[k] = L[i]$

$i = i + 1$

else

$A[k] = R[j]$

$j = j + 1$

- Beschreiben Sie in Worten, was der Algorithmus Shuffle bewirkt. Stellen Sie sich das Feld A dafür als Kartenstapel vor.

b) Nach dem Aufruf von `Shuffle(A)` sei das Feld $A = \langle 4, 1, 8, 5, 3, 2, 6, 7, 9 \rangle$. Die Folge der Zufallszahlen, die bei der Ausführung an die Variable `b` übergeben wurden, sei $\langle 0, 1, 1, 1, 0, 0, 1, 0, 1 \rangle$.

Geben Sie `A` vor der Ausführung von `Shuffle` an.

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

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.

Aufgabe 3 – Sortieren in Linearzeit

Die Eingabe sei eine Liste von t Feldern A_1, \dots, A_t mit einer Gesamtlänge von $n = \sum_{i=1}^t A_i.\text{length}$. Der Wertebereich ist für jedes Feld $\{1, \dots, k\}$ und weiter gilt $t \leq k$.

Geben Sie in Worten einen Algorithmus an, der alle Felder in insgesamt $O(n + k)$ Zeit sortiert! Jedes Feld soll am Ende die gleichen Elemente enthalten wie zu Beginn (nur in sortierter Reihenfolge). Begründen Sie die Laufzeit.

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 $n = A.\text{length}$. 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 17. und 18. Dezember 2019 gelöst. Sie brauchen Sie nicht vorher zu lösen und auch nicht abzugeben.