Lehrstuhl für Informatik I
Algorithmen, Komplexität und wissensbasierte Systeme
Universität Würzburg

Würzburg, den 28. November 2017

Prof. Dr. Alexander Wolff
Fabian Lipp, M. Sc.
Andre Löffler, M. Sc.
Johannes Blum, B. Sc.

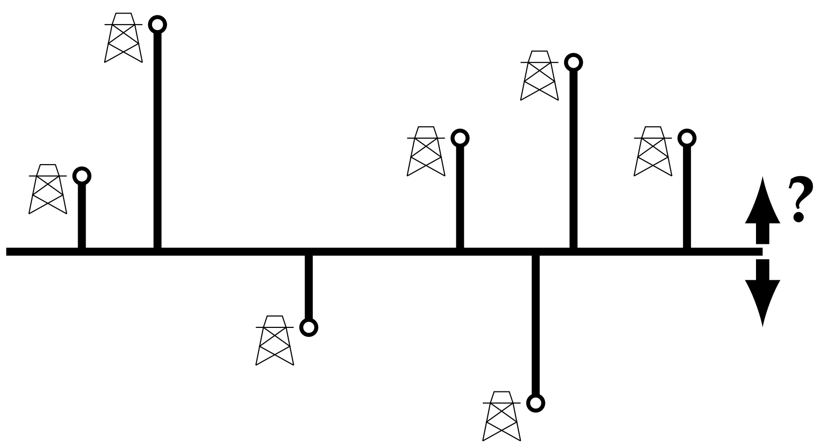
4. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2017/18)

Aufgabe 1 – Pipelines

Jett Rink möchte eine Pipeline in Ost-West-Richtung bauen, an die er seine Ölquellen möglichst kostengünstig anschließen kann. Jede Ölquelle soll über eine Zuleitung, die in Nord-Süd-Richtung verlaufen soll, an die Pipeline angeschlossen werden. Jett Rink fragt sich, wo die billigste Ost-West-Pipeline verläuft, also die, bei der die Gesamtlänge aller Zuleitungen minimiert wird.

Geben Sie einen Algorithmus mit Laufzeit in $O(n)$ an, der für eine Menge von n Ölquellen $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ die y-Koordinate einer billigsten Ost-West-Pipeline zurückgibt. Begründen Sie, warum der Algorithmus korrekt ist.

Hinweis: Sie dürfen davon ausgehen, dass die x-Koordinaten aller Ölquellen paarweise verschieden sind. **4 Punkte**



Aufgabe 2 – Sortieren in Linearzeit

Die Eingabe sei eine Liste von t Feldern $A_1, \ldots, A_t$ mit einer Gesamtlänge von $n = \sum_{i=1}^{t} A_i.length$. Der Wertebereich der Felder ist $\{1, \ldots, k\}$.

Geben Sie in Worten einen Algorithmus an, der alle Felder in insgesamt $O(n + k)$ Zeit sortiert! Begründen Sie die Laufzeit. **4 Punkte**

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

Aufgabe 3 – Schlange-2-Stapel

a) Wie lässt sich eine Schlange durch die Verwendung von zwei Stapeln implementieren?

   Welche asymptotischen Worst-Case-Laufzeiten haben die Methoden Enqueue und Dequeue in Ihrer Implementierung? **2 Punkte**

b) Wie lässt sich ein Stapel durch die Verwendung von zwei Schlangen implementieren?

   Welche asymptotischen Worst-Case-Laufzeiten haben die Methoden Push, Pop und Top in Ihrer Implementierung? **2 Punkte**

Aufgabe 4 – 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, wie der Algorithmus Shuffle vorgeht. Stellen Sie sich das Feld A dafür z.B. als Kartenstapel vor. **2 Punkte**

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. **1 Punkt**

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

d) Geben Sie in Pseudocode einen vergleichsbasierten Algorithmus UnShuffle an, der ein Feld A sortiert, das sortiert war und nur durch einmalige Anwendung von Shuffle gemischt wurde.

   Die Laufzeit des Algorithmus soll $\Theta(n)$ sein, wobei $n = A.length$.

   Begründen Sie, warum Ihr Algorithmus korrekt ist. **4 Punkte**

Bitte werfen Sie Ihre Lösungen bis **Dienstag, 5. Dezember 2017, 10:10 Uhr** in den Vorlesungs-Briefkasten im Informatik-Gebäude. Geben Sie stets die Namen und Übungsgruppen aller BearbeiterInnen sowie die Übungsgruppe, in der das Blatt zurückgegeben werden soll, an.

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.