Algorithmen und Datenstrukturen, Übung 6

Marouane Soussi, Lars Happel, Mustafa Miresh

Mai 2022

Aufgabe 1

siehe .java Datei

Aufgabe 2

a)

Zunächst definieren wir $i := n - k$, da die k größten Elemente gesucht sind.
Über `border := Selection(A, i)` wird die Grenze bestimmt, ab der Elemente aussortiert werden. (Benötigt $O(n)$ Zeit). Anschließend wird das Array linear durchlaufen (wieder $O(n)$) und für jeden Index e geprüft, ob $A[e] > border$. Falls ja, wird dieses Element in ein neues Array B kopiert. Letztlich wird B mittels z.B. Quicksort sortiert in $O(n * log(n)$ Zeit.

```
i := n-k
border := Selection(A, i)
B := [] // Neues, leeres Array
b := 1 // Index fuer B
for e = 1 to A.length:
    if A[e] > border:
        B[b] = A[e]
        b++
Quicksort(B)
return B
```

b)

Zunächst wird der Wert des Medians mit Selection(A, n/2) ermittelt.
Dann wird das Array durchlaufen und für jedes Element, dessen Differenz zum Median in ein zweites Array B geschrieben.
Aus B wird mittels `border := Selection(B, k)` ermittelt, welche Grenze zur Bestimmung der k dichtesten Elemente zum Median verwendet wird.
Zuletzt wird über einen Index j durch A bzw. B iteriert. Der Wert A[j] wird dann in ein drittes Array C kopiert, wenn $B[j] < border$.

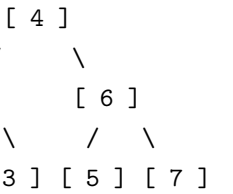
```
m := Selection(A, n/2)
for i = 1 to A.length:
    B[i] = Absolute(m - A[i]) \\ Betrag nehmen
border := Selection(B, k)
C := [] // Neues Array für gesuchte Werte
c := 1 // Index fuer C
for j = 1 to A.length:
    if B[j] < border:
        C[c] := B[j]
        c++
```

Aufgabe 3

a)

1.

Einzige Möglichkeit: 4 als Wurzel mit linkem Kind 2 und rechtem Kind 6.
Linkes Kind (2) hat links 1 und rechts 3 als Kind.
Rehtes Kind (6) hat links 5 und rechts 7 als Kind.



Eine andere Möglichkeit gibt es nicht, da 4 der Median des Arrays ist und der Baum mit dem Median als Wurzel balanciert ist. Der Baum muss balanciert sein, da bei 7 Elementen jeder "Slot" eines 2 Ebenen tiefen Baums belegt ist und jede Unbalancierung dazu führen würde, dass mehr als Tiefe 2 erreicht wird.

2.

Es gibt zwei Bäume, welche die Bedingung erfüllen, aber keine drei.
Damit bei 7 Elementen der Baum eine Höhe von 6 haben kann, darf kein Element mehr als ein Kind haben. Es gibt also die Möglichkeit bei 1 zu beginnen und jeweils das nächstgrößere Element in das rechte Kind zu platzieren, oder bei 7 zu beginnen und das nächstkleinere Element in das linke Kind zu platzieren. Jede Umordnung zweier Elemente würde, damit die Suchbaumeigenschaft erhalten bleibt, voraussetzen, dass ein Knoten zwei Kinder besitzt. Allerdings wäre dann die Höhe < 6.

3.

Ein solcher Baum ist mit Elementen aus $\mathbb{N}$ nicht möglich. Sei i ein beliebiger Knoten und $V(i)$ der Vaterknoten von i. Damit $V(4) = V(5)$ sein kann muss $V(4) \geq 4$ sein und $V(5) \leq 5$. Es blieben also $\{4, 5\}$ als theoretisch mögliche Lösungen. Da 4 und 5 aber je nur einmal in der Menge vorkommen und bereits in den Knoten 4 und 5 selbst repräsentiert sind, gibt es kein anderes Element, das beide Bedingungen erfüllt.

b)

Wert d. Knoten	Intervall im linken Teilbaum
800	$0 < x < 800$
669	$0 < x < 669$
424	$0 < x < 424$
503	$424 < x < 503$
600	$503 < x < 600$

Aufgabe 4