Algorithmen und Datenstrukturen
Blatt 3

Präsentation am 27.–29. November 2019

Jede Teilaufgabe (a, b, ...) zählt als ein einzelnes Kreuzchen.

Übung 1.
Betrachten Sie den folgenden Algorithmus für SelectionSort, der eine Liste von Zahlen absteigend sortiert:

```
SelectionSort(list l of integers)
list r := empty list
while l is not empty
    integer min := l.head
    for e ∈ l
        if min > e
            min := e
    Insert min at the beginning of r
    Remove min from l
return r
```

(a) Beweisen Sie die Korrektheit von SelectionSort.

(b) Geben Sie eine Laufzeitabschätzung für SelectionSort an. Ist die Laufzeit von SelectionSort für vergleichsbasiertes Sortieren asymptotisch optimal? Was ist der best-case von SelectionSort?

Übung 2.

(a) Sortieren Sie das Array $(12, 10, 6, 3, 1, 14, 9)$ mithilfe des QuickSort-Algorithmus aus der Vorlesung. Wählen Sie hierzu das Pivotelement wie im QuickSort-Algorithmus aus der Vorlesung vorgegeben. Stellen Sie den Inhalt des Arrays nach jedem Aufruf von Partition dar.

(b) Ist der QuickSort-Algorithmus aus der Vorlesung stabil? Begründen Sie Ihre Antwort.

Übung 3.
Geben Sie eine in Zeit $\Theta(n)$ laufende nichtrekursive Prozedur an, die eine einfach verkettete Liste aus n Elementen umkehrt. Stellen Sie sicher, dass Ihre Prozedur höchstens $\mathcal{O}(1)$ zusätzlichen Platz benötigt (neben dem Platz, die für die Eingabeliste gebraucht wird). Begründen Sie, dass Ihre Prozedur korrekt ist und die angegebenen Eigenschaften hat.

Übung 4.

(a) Beweisen Sie: Jeder nichtleere Binärbaum mit k inneren Knoten, in dem jeder innere Knoten zwei Kinder hat, hat $k + 1$ Blätter.

(b) Aus der Vorlesung ist Ihnen bekannt, dass zum Suchen in einem vollständigen Binärbaum $\mathcal{O}(\log_2 n)$ Zeit benötigt wird. Kann die Laufzeit asymptotisch verbessert werden, wenn jeder innere Knoten bis zu 3 Kinder haben darf? Wie sieht es aus, wenn jeder Knoten bis zu $\log_2 n$ Kinder haben darf?