

Übungen zur Vorlesung  
**Datenstrukturen und Algorithmen**

SS 2004

Blatt 4

**AUFGABE 1** (6 Punkte):

Die Operation `HEAP-DELETE( $A, i$ )` soll aus einem max-Heap den  $i$ -ten Eintrag  $A[i]$  entfernen und dabei die max-Heap-Eigenschaft aufrecht erhalten. Beschreiben Sie in Pseudocode einen Algorithmus mit Laufzeit  $\mathcal{O}(\log(n))$ , der diese Operation realisiert. Hierbei ist  $n$  die Größe des Eingabearrays. Zeigen Sie die Korrektheit Ihres Algorithmus und analysieren Sie seine Laufzeit!

**AUFGABE 2** (8 Punkte):

Gegeben sind  $k$  absteigend sortierte Arrays  $A_1, \dots, A_k$  mit insgesamt  $n$  Zahlen. Beschreiben Sie informell einen Algorithmus, der die  $k$  Arrays zu einem absteigend sortiertem Array zusammenfasst. Sie müssen den Algorithmus allerdings nicht in Pseudocode beschreiben. Der Algorithmus muss Laufzeit  $\mathcal{O}(n \log(k))$  besitzen. Wie üblich müssen Sie die Korrektheit des Algorithmus zeigen und seine Laufzeit analysieren!

*Hinweis:* Arbeiten Sie mit einem max-Heap, der zu jedem Zeitpunkt höchstens  $k$  Elemente enthält. Der Algorithmus `MERGE`, den wir bei `MERGE-SORT` kennengelernt haben, sollte ein einfacher Spezialfall Ihres Algorithmus sein.

**AUFGABE 3** (6 Punkte):

Betrachten Sie den folgenden Algorithmus zum Aufbau eines max-Heaps aus einem beliebigen Array  $A$ .

`BUILD-MAX-HEAP'(A)`

```
1 heap-size[A] ← 1
2 for  $i \leftarrow 1$  to length[A]
3   do MAX-HEAP-INSERT(A, A[i])
```

Zeigen Sie, dass dieser Algorithmus in Zeit  $\mathcal{O}(n \log(n))$  einen max-Heap aufbaut. Hierbei ist  $n$  die Größe des Eingabearrays.