Institut für Informatik
der Universität München
Prof. M. Hofmann
Dipl. Inf. Andreas Abel
Dipl. Math. Stefan Schimanski

SS 2006

8. Mai 2006

Übungen zur Vorlesung
Effiziente Algorithmen

Blatt 2

Aufgabe H-5: Entwerfen Sie eine Prozedur HEAP-INCREASE-KEY(A, i, k), die im Heap A das Element $A[i]$ durch $\max(A[i], k)$ ersetzt und die Heap-Eigenschaft erhält, und eine Laufzeit von $\Theta(\log n)$ bei einem Heap der Größe n hat. (4 Punkte)

Aufgabe H-6: Betrachten Sie die folgende alternative Implementierung von BUILD-HEAP:

```
BUILD-HEAP(A)
    heap-size[A] ← 1
    for i ← 2 to length[A]
        do HEAP-INSERT(A, A[i])
```

- Erzeugt diese Prozedur bei jeder Eingabe die selbe Ausgabe wie die in der Vorlesung vorgestellte Prozedur BUILD-HEAP? Beweisen Sie dies, oder geben Sie ein Gegenbeispiel an.
- Zeigen Sie, dass die obige Prozedur BUILD-HEAP im *worst-case* eine Laufzeit von $\Theta(n \log n)$ hat.

(6 Punkte)

Aufgabe H-7: Zeigen Sie, dass QUICKSORT bei Eingabe eines absteigend sortierten Arrays der Größe n eine Laufzeit von $\Theta(n^2)$ benötigt. (4 Punkte)

Aufgabe H-8: Zeigen Sie, dass es *kein* vergleichbasiertes Sortierverfahren gibt, das mindestens die Hälfte der $n!$ möglichen Eingaben der Länge n in linearer Zeit sortiert. Gibt es ein Verfahren, das jede $n$te Eingaben der Länge n in linearer Zeit sortiert? Oder wenigstens jede 2^nte? (6 Punkte)

Abgabe bis Montag, 15. Mai, 14.00 Uhr in einer der Vorlesungen oder Übungen oder im dafür vorgesehenen Briefkasten in der Oettingen- oder Theresienstraße.