Institut für Informatik
der Universität München
Dr. J. Johannsen
Dr. R. Matthes

SS 2003

8. Mai 2003

Übungen zur Vorlesung
Effiziente Algorithmen

Blatt 4

Aufgabe 14: Betrachten Sie die folgende alternative Implementierung von BUILD-HEAP:

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

(a) 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.

(b) Zeigen Sie, dass die obige Prozedur BUILD-HEAP im *worst-case* eine Laufzeit von $\Theta(n \log n)$ hat.

Aufgabe 15: Für eine Liste von n Zahlen sollen die k größten Elemente nach der Größe sortiert ausgegeben werden. Geben Sie für die beiden folgenden Methoden jeweils einen asymptotisch optimalen Algorithmus an, und analysieren Sie dessen Laufzeit in Abhängigkeit von n und k:

(a) Erzeuge aus der Liste eine Prioritätswarteschlange, und wende k mal EXTRACT-MAX an.

(b) Verwende einen Algorithmus zum Auffinden des k-größten Elements, partitioniere die Liste mit diesem als Pivot, und sortiere die so gefundene Liste der k größten Elemente.

Vergleichen Sie die Ergebnisse. Ist wirklich eine der Methoden besser?

siehe nächste Seite

Aufgabe 16:

(a) Für binäre Bäume gibt es die Heap-Eigenschaft und die Eigenschaft, binärer Suchbaum zu sein. Zeigen Sie deren Unterschied anhand der Aufgabenstellung, in linearer Zeit die Schlüssel eines binären Baums sortiert auszugeben.

(b) Zeigen Sie, dass jeder vergleichsbasierte Algorithmus zur Erzeugung eines binären Suchbaums aus einer beliebigen Liste von n Elementen Zeit $\Omega(n \log n)$ braucht.

Aufgabe 17: Zeigen Sie die Rot-Schwarz-Bäume, die entstehen, wenn man in einen anfänglich leeren Rot-Schwarz-Baum der Reihe nach die Schlüssel 41, 38, 31, 12, 19, 8 einfügt.

Geben Sie anschließend die Rot-Schwarz-Bäume an, die entstehen, wenn Sie aus dem oben erhaltenen Baum nacheinander die Schlüssel 12, 31, 38 entfernen.

Hinweis: Nach den Einfügungen hat man die schwarze Wurzel 38. Die Kinder sind 19 (rot) und 41 (schwarz). 19 hat schwarze Kinder 12 und 31. 12 hat links das Kind 8 (rot). Nach den Löschungen hat man einen schwarzen Baum mit Wurzel 19 und Kindern 8 und 41.

Aufgabe 18: Ausgehend von einem Rot-Schwarz-Baum T wird ein neues Element x mittels Insert eingefügt und dann mittels Delete wieder gelöscht. Ist der resultierende Rot-Schwarz-Baum wieder T? Beweisen Sie dies oder geben Sie ein Gegenbeispiel an.