Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Dr. Jens Ernst

WS 2003/2004
Übungsblatt 5
21. November 2003

Grundlegende Algorithmen

Letzter Abgabetermin: Freitag, 28. November 2003 (nach der Vorlesung)

Aufgabe 1 (15 Punkte)

Es soll die in der Vorlesung eingeführte Datenstruktur *Heap* um zwei neue Operationen erweitert werden.

a) Die Operation *insert* fügt für einen gegebenen Schüsselwert x einen neuen Knoten mit diesem Schlüssel in einen bestehenden Heap h der Größe n ein, so daß ein korrekter Heap der Größe $n+1$ entsteht. Beschreiben Sie einen möglichst effizienten Algorithmus für eine Operation *insert* mit der Funktionalität

```
heap insert(heap h, key x, unsigned n)
```

und analysieren Sie seine Komplexität[1]. Vergleichen Sie die Komplexität der Operation *create-heap* zum Aufbau eines Heaps der Größe N (siehe Vorlesung) mit den Kosten einer N-fachen Anwendung der *insert*-Operation auf einen anfangs leeren heap[1].

b) Die Operation *delete* erhält als Operanden einen Heap h, seine Größe n und einen Zeiger v auf einen Knoten V in h. Sie löscht V samt dem an ihm gespeicherten Schüsselwert und sorgt dafür, daß danach wieder ein korrekter heap der Größe $n-1$ vorliegt. Beschreiben Sie einen möglichst effizienten Algorithmus für eine Operation *delete* mit der Funktionalität

```
heap delete(heap h, node* v, unsigned n)
```

und analysieren Sie seine Komplexität[1].

Aufgabe 2 (5 Punkte)

Wieviele Schlüsselwerte können mindestens in einem Heap der Tiefe d gespeichert werden, wieviele höchstens?

Aufgabe 3 (5 Punkte)

Zeigen Sie, daß für jeden Teilbaum T eines Heaps (wie in der Vorlesung definiert) der minimale in T auftretende Schüsselwert an der Wurzel von T gespeichert ist.

[1]Hinweis: Es ist die Anzahl der Vergleiche zu untersuchen. Sie können die in der Vorlesung bewiesenen Aussagen zur Komplexität der bekannten Heap-Operationen verwenden.

Aufgabe 4 (15 Punkte)

Implementieren Sie den Heapsort-Algorithmus in der Variante nach R. W. Floyd, wie er in der Vorlesung besprochen wurde[2]. Es soll eine Funktion der Form

```
void HeapSort(int array[], unsigned n)
```

bereitgestellt werden, die ein gegebenes Array mit n ganzzahligen Schlüsselwerten aufsteigend sortiert. Ferner soll eine `main()`-Funktion programmiert werden, welche sich vom Benutzer den Wert n und dann n ganze Zahlen eingeben läßt, diese in ein Array speichert und darauf die Funktion `Heapsort()` aufruft. Die sortierte Folge soll auf dem Bildschirm ausgegeben werden. Verwenden Sie nach Möglichkeit die Programmiersprache C(++).
Bitte senden Sie Ihren Programmcode mit eventuellen Hinweisen zur Compilierung an die Email-Adresse `griebsch@in.tum.de`.

[2]Eine äquivalente Darstellung findet sich in Abschnitt 2.3.3 des Lehrbuchs von V. Heun.