



Informatik II: Algorithmen und Datenstrukturen

SS 2002

8. Übungsblatt

Aufgabe 24 (Binäre Suchbäume, 2T+5P)

Betrachten Sie die in der Vorlesung vorgestellten Operationen für binäre Suchbäume (Programme auf der Web-Seite).

- a) Ergänzen Sie eine Funktion

```
bs_baum bs_max(bs_baum b)
```

welche einen Zeiger auf einen Knoten mit maximalem Schlüssel zurückliefert. Analysieren Sie den Zeitaufwand in Abhängigkeit von der Höhe h des Baumes ($\mathcal{O}$ -Analyse genügt).

- b) Ergänzen Sie eine Funktion

```
int bs_hoehe(bs_baum b)
```

welche die Höhe des über b zugänglichen binären Suchbaums liefert. Analysieren Sie den Zeitaufwand dieser Funktion in Abhängigkeit von der Anzahl n der Knoten im Baum ($\mathcal{O}$ -Analyse genügt).

- c) Wenden Sie die Routinen auf folgendes Beispiel an:

- leeren Baum erzeugen
- nacheinander die Zahlen 5, 3, 8, 1, 10, 6, 9, 4 in dieser Reihenfolge einfügen
- Element mit Schlüssel = 5 löschen
- Element mit maximalem Schlüssel löschen

Nach jeder dieser Operationen (jedes Einfügen ist eine eigene Operation) soll die Höhe, der maximale Schlüssel sowie der komplette Baum ausgegeben werden.

Aufgabe 25 (Nachfolger in binären Suchbäumen, 9T)

Gegeben ist ein binärer Suchbaum b der Höhe h , dessen Knoten aus paarweise verschiedenen, ganzzahligen Schlüsseln bestehen. Als *Nachfolger* $n(s)$ eines Schlüssels s aus b bezeichnen wir den nächst größeren Schlüssel aus b , also

$$n(s) = \min\{t \in b \mid t > s\}.$$

- a) Skizzieren Sie für jeden der folgenden Fälle je einen binären Suchbaum und markieren Sie darin einen Schlüssel s , so dass gilt:
 - 1. $n(s)$ existiert nicht,
 - 2. $n(s)$ befindet sich im rechten Teilbaum von s ,
 - 3. $n(s)$ ist der Vater von s ,
 - 4. $n(s)$ ist ein Vorfahre – aber nicht der Vater – von s .
- b) Geben Sie einen Algorithmus mit Aufwand $\mathcal{O}(h)$ an zur Bestimmung des kleinsten Schlüssels in b .
- c) Zeigen Sie:
 - 1. Existiert $n(s)$, so ist $n(s)$ der kleinste Knoten im rechten Teilbaum von s oder ein Vorfahre von s .
 - 2. Besitzt s einen rechten Teilbaum, so liegt $n(s)$ darin.
 - 3. Besitzt s keinen rechten Teilbaum, so ist $n(s)$ sogar der erste Vorfahre von s , in dessen linken Teilbaum s liegt, falls ein solcher Vorfahre existiert.
- d) Formulieren Sie einen Algorithmus, der für gegebenes s mit Aufwand $\mathcal{O}(h)$ den Nachfolger $n(s)$ bestimmt, sofern er vorhanden ist. (Den Vorbereitungsschritt, in welchem der Schlüssel s im Baum lokalisiert wird, brauchen Sie nicht anzugeben).

Hinweis: Gehen Sie bei Aufwandsbetrachtungen davon aus, dass die Bestimmung des Vaters, des linken oder des rechten Sohnes eines Knotens den Aufwand $\mathcal{O}(1)$ besitzt.

Abgabe: Mi., 12.06.2002, 14 Uhr