Übungen zur Vorlesung Einführung in die Informatik II

Ausgabe: 25. Juni 2002 Abgabe: 1. Juli 2002, 13.00 Uhr

Die Abgabe erfolgt ausschließlich über das System eClaus! Siehe
http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ss02/info_II_02.html

Denken Sie die Vorlesungsumfrage: http://fachschaft.informatik.uni-stuttgart.de/VU/

Aufgabe 1: (Digitale Suchbäume) zum Votieren

Gegeben sei folgender Algorithmus für das Einfügen von Elementen in Digitale Suchbäume (siehe auch im Skript Plödereder Folie 176).

Einfügen und Löschen: Algorithmus für das Einfügen

```
function digitalinsert (v: INTEGER; x: in out NodeRef) return NodeRef is
-- Fuegt Element mit Schluessel v an richtiger Stelle im Baum x ein und liefert
-- Referenz auf dieses Element zurueck
   hilf1, hilf2 : NodeRef;
   b: Integer := maxb;
begin
   hilf1 := x; hilf2 := null;
   while hilf1 /= null loop
      hilf2 := hilf1;
      if bits (v, b, 1) = 0 then
         hilf1 := hilf1.l;
      else
         hilf1 := hilf1.r;
      end if;
      b := b - 1;
   end loop;
   hilf1 := new Node'(key => v, l => null, r => null);
   if hilf2 /= null then            -- Element ist nicht das einzige im Baum
      if bits (v, b+1, 1) = 0 then hilf2.l := hilf1;
                              else hilf2.r := hilf1;
      end if;
   else x := hilf1;
   end if;
   return x;
end digitalinsert;
```

4.4. Digitale Suchbäume (Radix-Suche) 176

(3 Punkte) Dieser Algorithmus arbeitet nicht völlig korrekt. Finden und begründen Sie die Fehler und schreiben Sie einen Algorithmus mit der gewünschten Funktionalität.

Aufgabe 2: (Hashing) schriftlich

In eine Hash-Tabelle der Größe 11 (Indizes 0 bis 10) sollen folgende neun Wörter in der angegebenen Reihenfolge eingetragen werden:
ANANAS, APFEL, BANANE, BIRNE, KIRSCHE, KIWI, MANGO, MELONE, PFLAUME.

a. (3 Punkte) Verwenden Sie als Hash-Funktion

$$h(w) = ((\text{Character}'\text{POS}(w_1) - \text{Character}'\text{POS}('A')) + 2 * \text{Länge}(w)) \bmod 11$$

wobei w_1 der erste Buchstabe des Wortes und Länge(w) die Zahl der Buchstaben in w ist (der Ausdruck $(\text{Character}'\text{POS}(w_1) - \text{Character}'\text{POS}('A'))$ bildet dabei die Buchstaben A bis Z in alphabetischer Reihenfolge auf die Werte 0 bis 25 ab).

Kollisionen sollen durch lineares Sondieren aufgelöst werden.

Geben Sie die Hashtabelle und Sondierungsfolgen an.

Geben Sie die Hashtabelle und Sondierungsfolgen an, wenn quadratisches Sondieren zur Kollisionsauflösung verwendet wird.

b. (2 Punkte) Löschen Sie in der Hash-Tabelle, die Sie mit linearem Sondieren erstellt haben, das Wort ANANAS. Erläutern Sie Ihre Vorgehensweise genau. (Nicht mit dem "gelöscht"-Bit arbeiten, sondern den Wert tatsächlich entfernen!)

c. (2 Punkte) Betrachten Sie die veränderte Hash-Funktion

$$h(w) = (a * (\text{Character}'\text{POS}(w_4) - \text{Character}'\text{POS}('A')) + b * \text{Länge}(w)) \bmod 11$$

Wie müssen Sie die Parameter a und b wählen, um ein perfektes Hashing zu ermöglichen (Hinweis: es gibt mehrere Lösungen)? Beachten Sie, dass hier in der Formel w_4, der vierte Buchstabe jedes Wortes, verwendet wird. Warum kann es kein perfektes Hashing geben, wenn man weiterhin w_1 verwenden würde?

Begründen Sie Ihre Antworten und beschreiben Sie Ihr Vorgehen.

Aufgabe 3: (Sortierverfahren) **zum Votieren**

(2 Punkte) Sortieren Sie die Zahlenreihe 2, 4, 10, 8, 6, 1, 5, 3, 7 und 9 mit den Verfahren *Direktes Aussuchen (Selection Sort)*, *Direktes Einfügen (Insertion Sort)* und *Bubblesort*. Geben Sie dabei jeweils alle Zwischenschritte an.
Zeigen Sie anhand eines Beispiels, dass das Sortierverfahren *Direktes Aussuchen (Selection Sort)* nicht stabil ist. Warum sind Verfahren wie *Direktes Einfügen (Insertion Sort)* und *Bubblesort* dagegen stabil?

Aufgabe 4: (Bubblesort) **zum Votieren**

Das in der Vorlesung behandelte Sortierverfahren *Bubblesort* hat die Eigenschaft, dass es sich asymmetrisch bezüglich der Durchlaufrichtung verhält, d.h., das Verfahren verhält sich bei zwei Folgen $a_1, ..., a_n$ und $a_n, ..., a_1$ evtl. sehr unterschiedlich.

a. (1 Punkt) Geben Sie ein Beispiel mit fünf zu sortierenden Zahlen an, an dem diese Asymmetrie besonders deutlich sichtbar wird. Beschreiben Sie, wie sich das Programm im Skript für Ihr Beispiel verhält.

b. (1 Punkt) Formulieren Sie eine verbesserte Version des im Skript angegebenen Programms indem Sie die Schleifengrenzen dynamisch möglichst präzise anpassen.

c. (1 Punkt) Zur Vermeidung der Asymmetrie wird im Skript die Verwendung des sog. *Shakersort* vorgeschlagen, bei dem das Datenfeld abwechselnd von oben und von unten durchlaufen wird. Geben Sie eine effiziente Implementierung des *Shakersort* an und wenden Sie diese auf Ihr Beispiel aus Teil a an.

Aufgabe 5: (Bubblesort-Variante) **schriftlich**

Bubblesort braucht wie jedes Verfahren, das nur benachbarte Elemente vergleicht und ggf. austauscht, im worst case stets $\Theta(n^2)$ Schritte. Dies kann man leicht am "Fehlstand" $\phi(a_1, ..., a_n) = |\{(a_i, a_j) \mid i < j, a_i > a_j\}|$ (die Anzahl der Elementpaare, bei denen das weiter links stehende Element größer ist) erkennen. Jedes Vertauschen zweier Nachbarn erniedrigt den Fehlstand ϕ höchstens um 1.

a. (1 Punkt) Finden Sie ein Beispiel, bei dem der Fehlstand ϕ in $\Theta(n^2)$ liegt (worst case beträgt $n(n-1)/2$).

b. (4 Punkte) Betrachten Sie folgendes Preprocessing für den Bubblesort Algorithmus:

```
for j in 2..n loop
   wähle zufällig ein i in 1..(j-1);
   if a_i > a_j then vertausche a_i und a_j end if;
end loop;
```

Nach diesem Vertauschen von maximal $n-1$ Elementpaaren, soll der Ihnen aus der Vorlesung bekannte Bubblesort Algorithmus durchgeführt werden.

(i) Welchen Fehlstand erhalten Sie nach diesem Preprocessing für Ihr Beispiel aus Teil a im günstigsten Fall? Welchen im schlechtesten Fall?

(ii) Programmieren Sie den Bubblesort Algorithmus in Ada 95.

(iii) Führen Sie jeweils 50 Experimente mit Feldgrößen $n = 10000, 20000, ..., 100000$ durch. Zählen Sie dabei jeweils die Vergleich und Vertauschungen und geben Sie diese im Mittel aus (beim modifizierten Bubblesort Algorithmus getrennt nach Preprocessing und Bubblesort Phase). Welchen Effekt hat das Preprocessing auf das Verhalten des Bubblesort?