

Algorithmen und Datenstrukturen

Wintersemester 2005/06

7. Übungsblatt

Aufgabe 30:

Demonstrieren Sie das Einfügen der Schlüssel 5, 28, 19, 15, 20, 33, 12, 17, 10 in eine Hash-Tabelle mit 9 Positionen. Mögliche Kollisionen sind durch Verkettung (Chaining) aufzulösen. Die Hash-Funktion sei $h(k) = k \bmod 9$.

Aufgabe 31:

Zur Kollisionsauflösung bei einer Hash-Tabelle werde doppeltes Hashen mit der Hash-Funktion

$$h(k, i) = (h_1(k) + ih_2(k)) \bmod m$$

verwendet. Zeigen Sie: Besitzen für festes k die Zahlen m und $h_2(k)$ den größten gemeinsamen Teiler $d \geq 1$, so wird bei erfolgloser Suche nach Schlüssel k nach der Sondierung von $1/d$ Positionen der Tabelle wieder Position $h_1(k)$ sondiert. Nur im Falle $d = 1$ wird somit die gesamte Tabelle sondiert.

Aufgabe 32:

In einer Hash-Tabelle mit Positionen $0, 1, \dots, m - 1$ werde eine Hash-Funktion $h : G \rightarrow \{0, 1, \dots, m - 1\}$ verwendet und zum Suchen eines Schlüssels k folgendes Verfahren betrachtet:

1. Berechne $i \leftarrow h(k)$ und setze $j \leftarrow 0$.
2. Position i Sondieren; falls diese leer ist oder Schlüssel k enthält, Suche beenden.
3. Setze $j \leftarrow (j + 1) \bmod m$ und $i \leftarrow (i + j) \bmod m$ und fahre bei 2. fort.

Dabei sei m eine Zweierpotenz.

Zeigen Sie, dass dieses Verfahren ein Beispiel von quadratischem Sondieren darstellt und bestimmen Sie die zugehörigen Parameter c_1 und c_2 .

Aufgabe 33:

Sei T ein binärer Suchbaum mit paarweise verschiedenen Schlüsseln.

- (a) Der Knoten $x \in T$ besitze einen leeren rechten Teilbaum und den Nachfolger $y \in T$. Zeigen Sie: y ist derjenige Vorfahre von x mit maximaler Tiefe, dessen linker Sohn ebenfalls Vorfahre von x ist.
- (b) Sei $x \in T$ ein Blatt mit Vater y . Zeigen Sie: es gilt entweder

$$key[y] = \min\{key[z] : z \in T \text{ und } key[z] > key[x]\}$$

oder

$$key[y] = \max\{key[z] : z \in T \text{ und } key[z] < key[x]\}.$$

Aufgabe 34:

Eine Menge von n Zahlen kann sortiert werden, indem man diese zuerst in einen binären Suchbaum einfügt (sukzessive Anwendung von TREE-INSERT) und dann durch INORDER-TREE-WALK in sortierter Reihenfolge wieder ausgeben läßt. Bestimmen Sie die Worst-Case und Best-Case Laufzeiten für diesen Sortieralgorithmus.

Aufgabe 35: (Programmieraufgabe)

Schreiben Sie ein C-Programm, welches eine Hash-Tabelle implementiert. Die Schlüssel seien vom Datentyp `int`. Als Kollisionsauflösung werde offene Adressierung verwendet; implementieren sie hierbei lineares sowie doppeltes Hashen mit geeignet gewählten Parametern. Neben Einfügen, Suchen, und Anzeigen sei auch Löschen in der Hash-Tabelle möglich. Bedienen Sie die Hash-Tabelle mit einem Rahmenprogramm ähnlich dem aus Programmieraufgabe 6.