

Prof. Dr. T. Härder  
 Fachbereich Informatik  
 AG Datenbanken und Informationssysteme  
 Universität Kaiserslautern

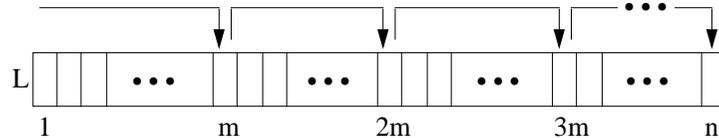
### Übungsblatt 5

für die freiwillige Übung

Unterlagen zur Vorlesung: „[www.dvs.informatik.uni-kl.de/courses/DBSREAL/](http://www.dvs.informatik.uni-kl.de/courses/DBSREAL/)“

#### Aufgabe 1: Suche in einer Liste - Sprungsuche

Gegeben ist eine Liste L mit n Einträgen fester Länge. Die Sprungsuche soll auf Liste L optimiert werden, wobei folgendes Schema angenommen wird:



a) Bei der einfachen Sprungsuche erfolgen konstante Sprünge zu den Positionen m, 2m, 3m usw. Sobald der Abschnitt, in den der gesuchte Schlüssel fällt, lokalisiert ist, wird sequentiell gesucht. Welche mittleren Suchkosten  $C_{avg}(n)$  ergeben sich, wenn ein Sprung a und ein sequentieller Vergleich b Einheiten kostet?

b) Was sind die Kosten  $C_{avg}(n)$  bei optimaler Sprungweite m?

c) Bei der Zwei-Ebenen-Sprungsuche wird statt sequentieller Suche im lokalisierten Abschnitt wiederum eine Quadratwurzel-Sprungsuche angewendet, bevor dann sequentiell gesucht wird. Welches  $C_{avg}(n)$  ergibt sich mit

- a = Kosten eines Sprungs auf der ersten Ebene,
- b = Kosten eines Sprungs auf der zweiten Ebene,
- c = Kosten für einen sequentiellen Vergleich?

d) Welche Verbesserungen ergeben sich durch optimale Abstimmung der Sprungweiten  $m_1$  und  $m_2$  der beiden Ebenen, wenn  $a = b = c$  gesetzt wird?

e) Läßt sich die Effizienz der Suche steigern, wenn das Verfahren zu einem n-Ebenen-Verfahren verallgemeinert wird?

#### Aufgabe 2: Erweiterbares Hashing

Die Sätze mit den Schlüsseln K10, J84, K35, A12, E77, K12, B22, K08 sollen in der angegebenen Reihenfolge mit dem Verfahren des Erweiterbaren Hashing in Buckets abgespeichert werden, die jeweils 2 Sätze aufnehmen können. Den Pseudoschlüssel des jeweiligen Satzes erhält man durch Verknüpfung der in EBCDIC codierten Zeichen mittels XOR. Zeichnen Sie die Belegung der Buckets und das Directory nach jeder Einfügung.

#### Aufgabe 3: Verweislisten und Bitlisten

Gegeben seien folgende Datensätze von Bestellungsinformationen, die die Adressen  $Z_n$  mit  $n \in \{1, 2, \dots, 10\}$  besitzen:

Adressen	Bestellnummer	Bestelldatum	Kundennummer
Z1	711	01.02.1999	100
Z2	600	03.01.1999	102
Z3	801	01.02.1999	141
Z4	505	02.01.1999	100
Z5	475	02.01.1999	102
Z6	730	01.02.1999	102
Z7	621	05.01.1999	141
Z8	699	09.01.1999	102
Z9	670	09.01.1999	100
Z10	515	02.01.1999	180

Invertieren Sie die oben vorgegebenen Datensätze jeweils nach den Attributen Bestelldatum und Kundennummer mit folgenden Verfahren:

- a) Verweislisten
- b) Bitlisten

#### Aufgabe 4: Bitlistenkomprimierung

Gegeben sei eine Bitliste der Länge 200 mit folgenden Eigenschaften:

An den Positionen 1, 2, 3, 73, 76, 90, 119, 135, 136, 161 tritt der Wert '1' und sonst '0' auf.

Führen Sie für die oben angegebene Bitliste eine Komprimierung nach jedem der folgenden Verfahren:

- a) Laufkomprimierung mit  $k = 6$
- b) Nullfolgenkomprimierung mit Codiereinheiten fester Länge mit  $k = 6$
- c) Nullfolgenkomprimierung unter der Anwendung der Golomb-Codierung mit  $m = 4$