INSTITUT FÜR INFORMATIK FREIBURG

Vorlesung Informatik 2
Algorithmen und Datenstrukturen

(27 - Selbstanordnende lineare Listen)

Prof. Dr. Susanne Albers

Problemstellung

Gegeben sei eine Menge von Objekten (Schlüsseln), auf die mit

- zeitlich veränderlichen
- unbekannten

Häufigkeiten zugegriffen wird.

Problem: Finde eine Speicherungsform, die die Zugriffskosten minimiert!

Lösungsidee: Speichere Schlüssel in eine sich selbst anordnenden linearen Liste!
(Sequentiell gespeichert)

Strategien für Selbstanordnung

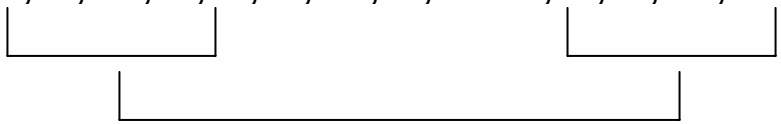
- MF (Move to front): Mache aktuelles Element zum ersten Listenelement
- T (Transpose): Vertausche aktuelles Element mit seinem Vorgänger
- FC (Frequency count): Führe eine Zugriffsstatistik mit und sortiere nach jedem Zugriff neu!

Beispiel für Anwendung von MF (1)

Gegeben Liste L = < 1, 2, 3, 4, 5, 6, 7>

Zugriffskosten für Element an Position i seien i.

Zugriffsfolge s: 1, 2, ..., 7, 1, 2, ..., 7, ..., 1, 2, ..., 7



10 mal

Kosten der Zugriffsfolge s:

Beispiel für Anwendung von MF (2)

Gegeben Liste L = < 1, 2, 3, 4, 5, 6, 7>

Zugriffskosten für Element an Position i seien i.

Zugriffsfolge s‘: 1, ..., 1, 2, ..., 2, ..., 7, ..., 7

10 mal 10 mal 10 mal

Kosten der Zugriffsfolge s‘:

Beispiel für Anwendung von MF (3)

Gegeben Liste L = < 1, 2, 3, 4, 5, 6, 7>

Zugriffskosten für Element an Position i seien i.

10 maliger Zugriff auf jedes der Elemente 1, ..., 7 (in beliebiger Reihenfolge)

Kosten bei statischer Anordnung:

Ergebnis: MF kann besser sein, als eine (jede) statische Anordnung, besonders, wenn die Zugriffe gehäuft auftreten

Beobachtungen

- MF-Regel ist „radikal“, aber (vielleicht) nicht schlecht!
- T-Regel ist „vorsichtiger“, aber:

  Für L = <1, 2, ..., N-1, N> liefert die Zugriffsfolge N, N-1, N, N-1, ...

  die durchschnittlichen Zugriffskosten N.
- FC-Regel benötigt zusätzlichen, prinzipiell nicht beschränkten Speicherplatz

Experimentelle Resultate zeigen:

T schlechter als FC,

MF und FC ungefähr gleich gut

MF hat Vorteile

Analyse: Mit Hilfe der amortisierten worst-case-Analyse

Analyse der MF-Regel

MF (Move to front): Mache aktuelles Element zum ersten Listenelement.

Ziel: Vergleich von MF mit beliebiger Strategie A zur Selbstanordnung.

Für eine Folge $s = s_1, s_2, ..., s_m$ von m Zugriffsoperationen und für eine Strategie A zur Selbstanordnung bezeichne:

$C_A(s)$ = Gesamtkosten (gesamte Zugriffskosten) zur Durchführung aller m Operationen von s gemäß Strategie A

Satz: Für jede Strategie A zur Selbstanordnung und jede Folge s vom m Zugriffs-Operationen gilt:

$$C_{MF}(s) <= 2\ C_A(s)$$

Analyse der Zugriffsregeln

Kostenmodell: Zugriff auf Element an Position i verursacht Kosten i

Amortisierte Worst-Case-Analyse

Unter den amortisierten worst-case Kosten einer Folge $o_1, ..., o_n$ von Operationen, die auf einer gegebenen Struktur ausgeführt werden, versteht man die durchschnittlichen Kosten pro Operation für eine schlechtest mögliche Wahl der Folge $o_1, ..., o_n$.

Genauer als simple, oft zu pessimistische worst-case-Analyse

Mögliche Vorgehensweisen bei der amortisierten worst-case-Analyse:

- Gesamtkosten der Operationsfolge berechnen und Durchschnitt bilden (Aggregat Methode)
- Bankkonto Paradigma: Bezahle für die „billigen“ Operationen (freiwillig) etwas mehr und bezahle die teueren von den Ersparnissen
- Potentialmethode

Potentialmethode

Sei Φ: D → IR, D Menge der Datenstrukturzustände

Sei Φ_i das „Potential“ der Datenstruktur nach der i-ten Zugriffs-Operation.

t_i = wirkliche Kosten der i-ten Operation

Definiere die amortisierten Kosten a_i der i-ten Zugriffs-Operation durch:

a_i =

Analyse mit Potentialfunktion

Betrachte die Wirkungen der Strategien MF und A bei Ausführung von s_1, s_2, ...

für eine Ausgangsfolge L,

und ordne jedem nach Ausführung der ersten i Operationen erreichten Zustand ein Potential Φ_i zu.

Inversionszahl

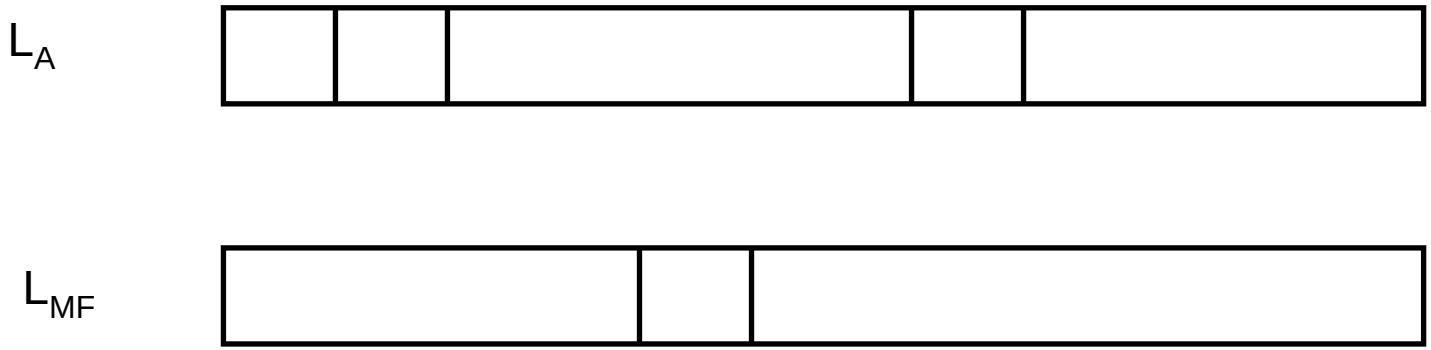
Definiere die Inversionszahl $inv(L_1, L_2)$ zweier Listen L_1 und L_2, die dieselben Elemente in ggfs. unterschiedlicher Reihenfolge enthalten, als die Anzahl der Inversionen von Elementen in L_1 bzgl. L_2.

$L_1 = <3, 2, 1, 4, 6, 7, 5>$ $L_2 = <2, 1, 3, 6, 4, 7, 5>$

Definition der Potentialfunktion

Das einem Paar von Listen, die nach der i-ten Operation mit Strategie MF und A entstehen, zugeordnete Potential Φ_i ist:

Wirkung einer Zugriffsoperation, MF



Wirkung einer Zugriffsoperation, A