IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

Vorlesung Informatik 2
Algorithmen und Datenstrukturen

(16 – Dynamische Tabellen)

Prof. Dr. Susanne Albers

Dynamische Tabellen

Problem: Verwaltung einer Tabelle unter den Operationen Einfügen und Entfernen, so dass

- die Tabellengröße der Anzahl der Elemente angepasst werden kann
- immer ein konstanter Anteil der Tabelle mit Elementen belegt ist
- die Kosten für n Einfüge- oder Entferne-Operationen $O(n)$ sind.

Organisation der Tabelle: Hashtabelle, Heap, Stack, etc.

Belegungsfaktor α_T : Anteil der Tabellenplätze von T, die belegt sind.

Kostenmodell:

Einfügen oder Entfernen eines Elementes verursacht Kosten 1, wenn Tabelle noch nicht voll. Wird Tabellengröße geändert, müssen zunächst alle Elemente kopiert werden.

Initialisierung

```
class dynamicTable {

    private int [] table;

    private int size;
    private int num;

    dynamicTable () {
        table = new int [1];//Initialisierung leere Tabelle
        size = 1;
        num = 0;
    }
```

Vergrößerungsstrategie: Einfügen

Verdoppele Tabellengröße, sobald versucht wird, in bereits volle Tabelle einzufügen!

```
public void insert (int x) {
       if (num == size) {
  int[] newTable = new int[2*size];
    for (int i=0; i < size; i++)
     fuege table[i] in newTable ein;
          table = newTable;
           size = 2*size;
                }
      fuege x in table ein;
         num = num + 1;
              }
```

Einfüge-Operationen in eine anfangs leere Tabelle

t_i = Kosten der *i*-ten Einfüge-Operation

Worst case:

t_i = 1, falls die Tabelle vor der Operation i nicht voll ist

t_i = (i – 1) + 1, falls die Tabelle vor der Operation i voll ist.

Also verursachen *n* Einfüge-Operationen höchstens

Gesamtkosten von $\sum_{i=1}^{n}(i) = O(n^2)$

Amortisierter Worst-Case:

Aggregat -, Bankkonto -, Potential-Methode

Potential-Methode

T Tabelle mit

- $k = T.num$ Elementen und
- $s = T.size$ Größe

Potentialfunktion

$$\phi(T) = 2k - s$$

Eigenschaften der Potentialfunktion

Eigenschaften

- $\phi_0 = \phi(T_0) = \phi$ (leere Tabelle) = -1

- Für alle $i \geq 1$: $\phi_i = \phi(T_i) \geq 0$

  Weil $\phi_n - \phi_0 \geq 0$ gilt, ist $\Sigma\, a_i$ eine obere Schranke für $\Sigma\, t_i$

- Unmittelbar vor einer Tabellenexpansion ist $k = s$,

  also $\phi(T) = k = s$.

- Unmittelbar nach einer Tabellenexpansion ist $k = s/2$,

  also $\phi(T) = 2k - s = 0$.

Amortisierte Kosten des Einfügens (1)

k_i = # Elemente in T nach der i-ten Operation

s_i = Tabellengröße von T nach der i-ten Operation

Fall 1: [*i*-te Operation löst keine Expansion aus]

Amortisierte Kosten des Einfügens (2)

Fall 2: [*i*-te Operation löst Expansion aus]

Einfügen und Entfernen von Elementen

Jetzt: Kontrahiere Tabelle, wenn Belegung zu gering!

Zíele:

(1) Belegungsfaktor bleibt durch eine Konstante

nach unten beschränkt

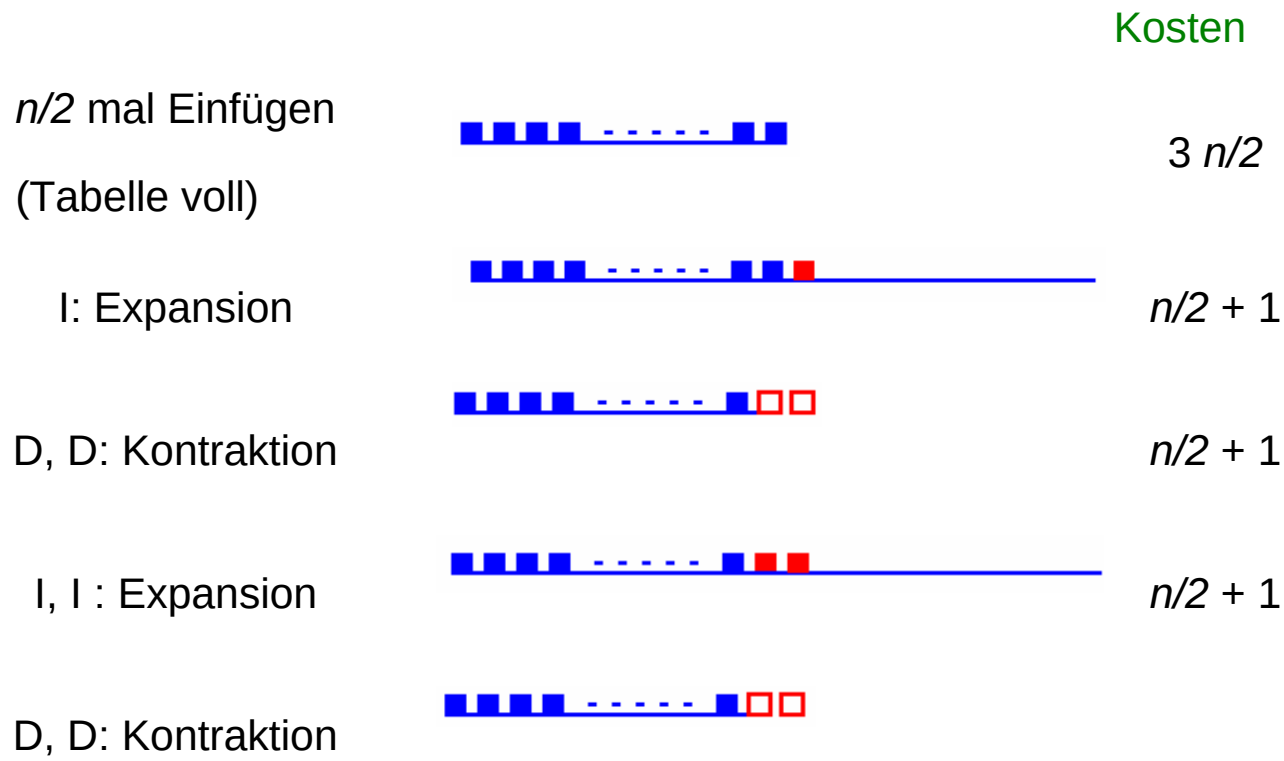
(2) amortisierte Kosten einer einzelnen Einfüge- oder Entferne- Operation sind konstant.

1. Versuch

- Expansion: wie vorher
- Kontraktion: Halbiere Tabellengröße, sobald Tabelle

weniger als ½ voll ist (nach Entfernen)!

„Schlechte“ Folge von Einfüge- und Entfernenoperationen



Gesamtkosten der Operationsfolge:
$I_{n/2}$, *I,D,D,I,I,D,D*,... der Länge *n* sind

2. Versuch

Expansion: Verdoppele die Tabellengröße, wenn in die volle Tabelle eingefügt wird.

Kontraktion: Sobald der Belegungsfaktor unter ¼ sinkt ,

halbiere die Tabellengröße.

Folgerung:

Die Tabelle ist stets wenigstens zu ¼ voll, d.h.

$$¼ \leq \alpha(T) \leq 1$$

Kosten einer Folge von Einfüge- und

Entferne-Operationen?

Analyse Einfügen und Enfernen

k = T.num, s = T.size, α *= k/s*

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Analyse Einfügen und Entfernen

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Unmittelbar nach einer Expansion oder Kontraktion der Tabelle:

$s = 2k$, also $\phi(T) = 0$

Einfügen

i-te Operation: $k_i = k_{i-1} + 1$

Fall 1: $\alpha_{i-1} \geq ½$

Fall 2: $\alpha_{i-1} < ½$

Fall 2.1: $\alpha_i < ½$

Fall 2.2: $\alpha_i \geq ½$

Einfügen

Fall 2.1: $\alpha_{i-1} < ½$, $\alpha_i < ½$ (keine Expansion)

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Einfügen

Fall 2.2: $\alpha_{i-1} < ½$, $\alpha_i \geq ½$ (keine Expansion)

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Entfernen

$k_i = k_{i-1} - 1$

Fall 1: $\alpha_{i-1} < ½$

Fall1.1: Entfernen verursacht keine Kontraktion

$s_i = s_{i-1}$

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Entfernen

$k_i = k_{i-1} - 1$

Fall 1: $\alpha_{i-1} < ½$

Fall 1.2: $\alpha_{i-1} < ½$ Entfernen verursacht Kontraktion

$2s_i = s_{i-1}$ $k_{i-1} = s_{i-1}/4$

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Entfernen

Fall 2: $\alpha_{i-1} \geq ½$ keine Kontraktion

$$s_i = s_{i-1} \quad k_i = k_{i-1} - 1$$

Fall2.1: $\alpha_{i-1} \geq ½$

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$

Entfernen

Fall 2: $\alpha_{i-1} \geq ½$ keine Kontraktion

$$s_i = s_{i-1} \quad k_i = k_{i-1} - 1$$

Fall2.2: $\alpha_i < ½$

Potentialfunktion ϕ

$$\phi(T) = \begin{cases} 2k - s, \text{ falls } \alpha \geq 1/2 \\ s/2 - k, \text{ falls } \alpha < 1/2 \end{cases}$$