Technische Universität Braunschweig

Algorithmen und Datenstrukturen II

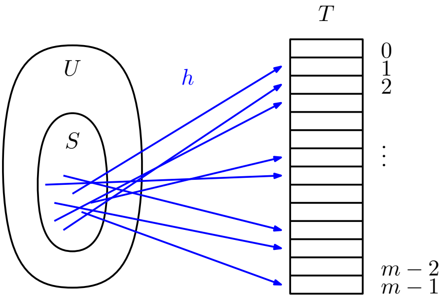
11. Vorlesung

Linda Kleist, 10.07.2019

Hashverfahren – Wiederholung

Universum $U = \{0, 1, \dots, N-1\}$
Schlüsselmenge $S \subseteq U$, $n := |S|$

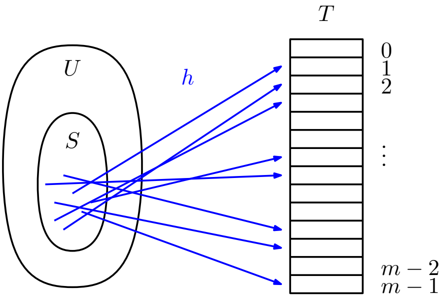
- Eine Hashtabelle der Größe m ist ein Array T mit den Zellen $T[0]$ bis $T[m-1]$ zur Speicherung der Datensätze.
- Eine Hashfunktion h liefert für jeden Schlüssel $x \in U$ eine Adresse in der Hashtabelle, d.h. $h : U \to \{0, \dots, m-1\}$.



Hashverfahren – Wiederholung

Universum $U = \{0, 1, \ldots, N-1\}$
Schlüsselmenge $S \subseteq U$, $n := |S|$

- Der Belegungsfaktor einer Hashtabelle der Größe m ist $\beta := \frac{n}{m}$.
- Bei einer Kollision erhalten verschiede Schlüssel x_1 und x_2 den selben Hashwert $h(x_1) = h(x_2)$ (unvermeidbar wenn $|U| > m$).



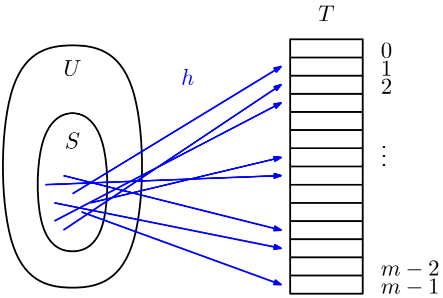
Hashverfahren – Wiederholung

Universum $U = \{0, 1, \ldots, N-1\}$
Schlüsselmenge $S \subseteq U$, $n := |S|$

- Ein Hashverfahren ist durch
  - eine Hashtabelle,
  - eine Hashfunktion, und
  - eine Strategie zur Auflösung von Kollisionen

  gegeben.

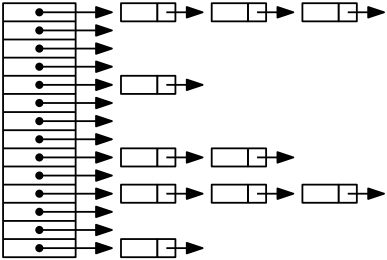
- Herausforderungen
  - $|U| \gg |S|$
  - S ist a priori unbekannt



Kollisionsbehandlung – Wiederholung

Möglichkeiten zur Kollisionsbehandlung:

- Verkettete Listen: Jede Zelle der Hashtabelle enthält einen Zeiger auf eine Überlaufliste.
- Offene Adressierung: Bei Adresskollision nach fester Regel alternativen freien Platz suchen (Sondierungsfolge).



Offene Adressierung (OA)

Im Kollisionsfall wird nach einer festen Regel (Sondierungsfolge) ein freier Platz in der Hashtabelle gesucht.

Annahme: $n < m$, d.h. $\beta < 1$!

Eine Sondierungsfolge ist gegeben durch eine Abbildung

$t(x, i) : U \times \{0, \ldots, m-1\} \to \{0, \ldots, m-1\}$, wobei
$t(x, i)$ die Position des i-ten Versuchs zum Einfügen von x beschreibt.

Anforderungen an die Abbildung $t(x, \cdot)$

- berechenbar in $O(1)$,
- $t(x, 0) = h(x)$,
- $(t(x, 0), \ldots, t(x, m-1))$ ist eine Permutation von $(0, 1, \ldots, m-1)$ (Permutationsbedingung)

Offene Adressierung (OA)

Operationen

- search(x): Suche x an den Positionen $t(x,0),\ldots,t(x,m-1)$. Brich ab, falls x oder eine freie Stelle gefunden ist.
- insert(x): Nach erfolgloser Suche, finde freien Platz und füge x dort ein.

Bemerkungen:

- delete(x) kann nicht einfach ausgeführt werden, da entstehende Lücken bei einer search-Operation zu Fehlern führen würden.
- Möglichkeit: Markierung mit 'besetzt', 'noch nie besetzt', 'wieder frei'
- Problem: Mit der Zeit gibt es keine *noch nie besetzten* Positionen mehr $\implies$ Hashing wird ineffizient

Wir betrachten OA nur mit den Operationen search und insert!

Offene Adressierung

Varianten

- Lineares Sondieren
- Quadratisches Sondieren
- Doppeltes Hashing

Hilfsmittel zur Analyse: Uniformes Hashing

OA I: Lineares Sondieren

$$t(x, i) = (h(x) + i) \bmod m$$

- Permutationsbedingung immer erfüllt

Beispiel

Für $m = 19$ und $h(x) = 7$, erhalten wir die Sondierungsfolge:
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 0, 1, 2, 3, 4, 5, 6.

- Bildung von zusammenhängenden belegten Abschnitten (primäres Clustering) $\implies$ erhöhte Suchzeiten

OA I: Lineares Sondieren

$$t(x,i) = (h(x)+i) \bmod m$$

- Permutationsbedingung immer erfüllt
- Bildung von zusammenhängenden belegten Abschnitten (primäres Clustering) $\implies$ erhöhte Suchzeiten

Beispiel

Hashtabelle mit $m = 19$, wobei Positionen 2, 5, 6, 9, 10, 11, 12, 17 belegt

Pos i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$h(x) = i$ landet bei:	0	1	3	3	4	7	7	7	8	13	13	13	13	13	14	15	16	18	18
Belegungs-W'keit:	$\frac{1}{19}$	$\frac{1}{19}$	0	$\frac{2}{19}$	$\frac{1}{19}$	0	0	$\frac{3}{19}$	$\frac{1}{19}$	0	0	0	0	$\frac{5}{19}$	$\frac{1}{19}$	$\frac{1}{19}$	$\frac{1}{19}$	0	$\frac{2}{19}$

- Bei belegtem Abschnitt der Länge q wird neues x mit zufälligem und gleichverteiltem $h(x)$ mit Wahrscheinlichkeit $\frac{q+1}{m}$ in darauffolgender Position abgelegt $\implies$ weit weg von ideal

OA II: Quadratisches Sondieren

Idee: additiver Term wächst quadratisch

$$t(x, i) = \left(h(x) + c_1 i + c_2 i^2\right) \bmod m \text{ mit } c_1, c_2 \in \mathbb{R}$$

Beispiel

Für $m = 16$, $h(x) = 0$, $c_1 = c_2 = \frac{1}{2}$ ergibt sich die Sondierungsfolge:
0, 1, 3, 6, 10, 15, 5, 12, 4, 13, 7, 2, 14, 11, 9, 8

- Wenn $h(x_1) = h(x_2)$, dann haben x_1 und x_2 die selbe Sondierungsfolge (sekundäres Clustering)
- Permutationsbedingung hängt von m, c_1, c_2 ab

Satz 16

Wenn m Zweierpotenz und $c_1 = c_2 = \frac{1}{2}$, dann erfüllt $t(x, \cdot)$ die Permutationsbedingung.

OA III: Doppeltes Hashing

Idee: Verknüpfe zwei Hashfunktionen $h_1(x)$ und $h_2(x)$ additiv.

$$h(x, i) := \big(h_1(x) + i \cdot h_2(x)\big) \bmod m$$

Achtung: Wenn $h_2(x) = 0$ ist die Sondierungsfolge konstant.

Beispiel

$m = 19$, $x = 23$, $h_1(x) = x \bmod m$, $h_2(x) = x \bmod (m - 2) + 1$ ergibt
$h_1(23) = 23 \bmod 19 = 4$,
$h_2(23) = 23 \bmod 17 + 1 = 7$ und die Sondierungfolge:
4, 11, 18, 6, 13, 1, 8, 15, 3, 10, 17, 5, 12, 0, 7, 14, 2, 9, 16.

OA III: Doppeltes Hashing

Idee: Verknüpfe zwei Hashfunktionen $h_1(x)$ und $h_2(x)$ additiv.

$$h(x,i) := \big(h_1(x) + i \cdot h_2(x)\big) \bmod m$$

Achtung: Wenn $h_2(x) = 0$ ist die Sondierungsfolge konstant.

Satz 17

$h(x,i) := \big(h_1(x) + i \cdot h_2(x)\big) \bmod m$ erfüllt die Permutationsbedingung wenn für alle $x \in U$ gilt: $h_2(x) \neq 0$ und $\mathrm{ggT}(m, h_2(x)) = 1$.

Bemerkung: $\mathrm{ggT}(m, h_2(x)) = 1$ ist (z.B.) erfüllt wenn m Primzahl, oder m Zweierpotenz und $h_2(x)$ ungerade.

Beobachtungen: Wenn Permutationsbedingung erfüllt ist, dann ist

- $(0, 1 \cdot h_2(x), \ldots, (m-1) \cdot h_2(x))$ eine Permutation von $(0, \ldots, m-1)$
- und der Summand $h_1(x)$ verschiebt den Anfang zufällig.

$\Longrightarrow$ Doppeltes Hashing kommt dem idealen Hashing am nächsten.

Offene Adressierung – Analyse

Uniform Hashing Annahme

Jedem Schlüssel x wird eine der $m!$ Permutationen von $(0, 1, \dots, m-1)$ mit Wahrscheinlichkeit $\frac{1}{m!}$ als Sondierungssequenz zugewiesen.

Satz 18

Unter der Uniform Hashing Annahme gilt für jede Hashtabelle mit Belegungsfaktor $\beta = \frac{n}{m} < 1$ gilt: Die durchschnittliche Anzahl der Sondierungsversuche

- bei einer erfolglosen Suche ist höchstens $\frac{1}{1-\beta}$ und
- bei einer erfolgreichen Suche ist höchstens $\frac{1}{\beta} \ln(\frac{1}{1-\beta})$.

Beweis.

Tafel... $\square$

Offene Adressierung – Analyse

Uniform Hashing Annahme

Jedem Schlüssel x wird eine der $m!$ Permutationen von $(0, 1, \ldots, m-1)$ mit Wahrscheinlichkeit $\frac{1}{m!}$ als Sondierungssequenz zugewiesen.

Satz 18

Unter der Uniform Hashing Annahme gilt für jede Hashtabelle mit Belegungsfaktor $\beta = \frac{n}{m} < 1$ gilt: Die durchschnittliche Anzahl der Sondierungsversuche

- bei einer erfolglosen Suche ist höchstens $\frac{1}{1-\beta}$ und
- bei einer erfolgreichen Suche ist höchstens $\frac{1}{\beta} \ln(\frac{1}{1-\beta})$.

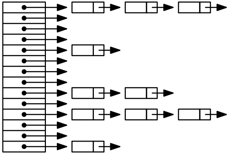
Beispiel

Für $\beta = \frac{1}{2}$ gilt: $\frac{1}{1-\beta} = 2$ (erfolglos) und $\frac{1}{\beta} \ln(\frac{1}{1-\beta}) \approx 1,4$ (erfolgreich).
Für $\beta = \frac{9}{10}$ gilt: $\frac{1}{1-\beta} = 10$ (erfolglos) und $\frac{1}{\beta} \ln(\frac{1}{1-\beta}) \approx 2,6$ (erfolgreich).

Zusammenfassung

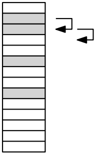
verkette Listen

- Idee: Überlaufliste
- Operationen:
  search, insert, delete
- $n > m$ möglich
- Laufzeiten
  - im average case: $\Theta(1)$
    erfolglose Suche: $1 + \beta$
    erfolgreiche Suche: $1 + \frac{\beta}{2}$
  - im worst case: $\Theta(n)$



offene Adressierung

- Idee: neuen Platz suchen
- Operationen:
  search, insert
- $n < m$, d.h. $\beta < 1$
- Laufzeiten
  - im average case: $\Theta(1)$
    erfolglose Suche: $\frac{1}{1-\beta}$
    erfolgreiche Suche: $\frac{1}{\beta}\ln(\frac{1}{1-\beta})$
  - im worst case: $\Theta(n)$



Universelles Hashing

- worst case: für jede Schlüsselmenge gibt es eine schlechte Hashfunktion
- Idee: wähle Hashfunktion zufällig

Eine Familie von Hashfunktionen $\{h \mid h\colon U \to \{0, \dots, m-1\}\}$ heißt universell wenn für alle $x_1, x_2 \in U$ mit $x_1 \neq x_2$ gilt:

$$\text{Prob}[h(x_1) = h(x_2)] \leq \frac{1}{m}.$$

Beispiel für universelle Familie

Sei $p \geq m$ Primzahl

$H := \{h_{a,b}(x) = ((ax+b) \bmod p) \bmod m \mid a, b \in \{0, \dots p-1\}, a \neq 0\}$

- Die erwartete Anzahl an Kollisions-Paaren ist höchtens $\frac{n^2}{2m}$.
- Wenn $m = n^2$ gibt es zu $\geq 50\%$ keine Kollision.