Grundzüge DS & Alg (WS14/15)

Lösungsvorschlag zu Aufgabenblatt 8

Aufgabe 1

In dieser Aufgabe beschäftigt man sich mit $pseudo-universellen$ Familien von Hashfunktionen.

Eine Menge $\mathcal{H}$ von Funktionen $h : U \to \{0, 1, \ldots, t-1\}$ heißt $pseudo-universell$, wenn für jedes $x \in U$ und jedes $i \in \{0, 1, \ldots, t-1\}$ gilt

$$\frac{|\ \{h \in \mathcal{H} \mid h(x) = i\}\ |}{|\mathcal{H}|} \leq \frac{1}{t}.$$

Es sei jetzt $S \subset U$ mit $|S| = n$.

a) Zu zeigen ist, daß bei zufälliger Wahl einer Hashfunktion h aus einer pseudo-universellen Familie $\mathcal{H}$ für jedes $i \in \{0, 1, \ldots, t-1\}$ die *erwartete* Anzahl von Elementen von S, die durch h auf i abgebildet werden, höchstens $\frac{n}{t}$ ist.

Dazu definiert man sich zunächst analog zur Vorlesung eine Funktion C_i, die die Stücke von S, die durch h auf i abgebildet werden, zurückliefert:

$$C_i = \{x \in S \mid h(x) = i\}.$$

Der für uns interessante Wert ist also der Erwartungswert von $|C_i|$. Um diesen leichter berechnen zu können, benötigt man noch die passende Indikatorvariable δ_{ix}:

$$\delta_{ix} = \begin{cases} 1, & h(x) = i \\ 0, & \text{sonst} \end{cases}$$

Somit läßt sich der Erwartungswert jetzt analog zur Vorlesung berechnen:

$$E[|C_i|] = \sum_{x \in S} E[\delta_{ix}] = \sum_{x \in S} Pr(\delta_{ix} = 1) = \sum_{x \in S} \underbrace{\frac{|\ \{h \in \mathcal{H} \mid h(x) = i\}\ |}{|\mathcal{H}|}}_{\leq \frac{1}{t}, da\ pseudo-univ.} \leq \frac{n}{t}.$$

b) Gesucht ist ein Beispiel für eine Hashfamilie $\mathcal{H}$, die zwar pseudo-universell ist, aber trotzdem für die Operationen `SEARCH` und `DELETE` wesentlich mehr als $\Omega(\frac{n}{t})$ Operationen benötigt.

Sei $\mathcal{H} = \{h_k \mid h_k(x) = k\ , 0 \leq k < t,\ \forall x \in U\}$. $\mathcal{H}$ ist pseudo-universell, denn es gilt:

- $|\mathcal{H}| = t$, da $0 \leq k < t$.
- $|\{h \in \mathcal{H} | h(x) = i\}| \leq 1$, da die h_k's jeweils konstant sind.

Daraus folgt

$$\frac{|\{h \in \mathcal{H} \mid h(x) = i\}|}{|\mathcal{H}|} \leq \frac{1}{t}.$$

Der Nachteil dieser Familie von Hashfunktionen ist leicht zu erkennen. Da jede Hashfunktion konstant ist, werden die Elemente bei Wahl einer Hashfunktion h_k alle auf k abgebildet. An Stelle k werden alle n Elemente also als Liste verwaltet. Die Anzahl von Operationen, die zum Suchen, bzw. Löschen eines Elementes benötigt werden ist im worst-case also $O(n)$ und somit deutlich höher als $\Omega(\frac{n}{t})$.

Aufgabe 2

Zu zeigen ist, daß die Menge $\mathcal{G} = \{g_a \mid 0 \leq a < t\}$ mit

$$g_a(x) = \left(\sum_{0 \leq i < r} a^i x_i \right) \mod t$$

universell, bzw. falls nein c-universell ist.

Es gilt:

$$\begin{aligned} & \frac{|\{a \mid (\sum_{0 \leq i < r} a^i x_i) \bmod t = (\sum_{0 \leq i < r} a^i y_i) \bmod t\}|}{t} \\ = \; & \frac{|\{a \mid \sum_{0 \leq i < r} a^i (x_i - y_i) = 0 \bmod t\}|}{t} \\ \leq \; & \frac{r-1}{t}. \end{aligned}$$

Da $\mathcal{G}$ nach Voraussetzung nur t Funktionen umfaßt ergibt sich der Nenner zu t. Betrachtet man den Zähler, so stellt man fest, daß es sich bei dem Term

$$\sum_{0 \leq i < r} a^i (x_i - y_i) = 0 \bmod t$$

um ein Polynom vom Grad $r-1$ in der Variablen a mit konstanten Koeffizienten $(x_i - y_i)$ handelt. Das Polynom hat also $r-1$ Nullstellen, sprich maximal $r-1$ verschiedene Lösungen für a, da man sich hier wiederum in einem Körper befindet.

Die Menge ist folglich nicht universell, aber c-universell mit $c = r - 1$.

Grundzüge DS & Alg (WS14/15)

Lösungsvorschlag zu Aufgabenblatt 8

Aufgabe 3

Die Idee des folgenden Verfahrens ist es, dass der Rehash zu einem zufälligen und somit nicht bekannten Zeitpunkt stattfindet. Dazu werden Blöcke von Operationen festgelegt, wobei ein Block die Länge $\frac{n}{2}$ hat und n die Anzahl der in der Table enthalten Elemente ist.

start with n, t fixed ($t \propto n$)
while true **do**
  pick $i \in \{1, ..., \frac{n}{2}\}$
  execute i operations
  rehash into new table with $t = c \cdot n'$
  execute $\frac{n}{2} - i$ operations
  update n to the current value
end while

n' ist dabei die Anzahl der zu diesem Zeitpunkt enthaltenen Elementen und c eine Konstante (z.B. 1)

1. Operationen brauchen konstante Zeit

   Der Rehash dauert: $c'' \cdot c \cdot n' \leq c'' \cdot c \cdot (n + \frac{n}{2})$ mit Wahrscheinlichkeit $\frac{2}{n}$

   $\Rightarrow$ erwartete Laufzeit: $c'' \cdot c \cdot 3$

   c'' ist dabei eine Konstante die angibt, wie lange es dauert um ein Element in die neue Hashtable zu übertragen.

2. $t \in \Theta(n_c)$ mit hoher Wahrscheinlichkeit

   Im folgenden wird davon ausgegangen dass $c = 1$ im Algorithmus verwendet wird, es würde aber auch mit anderen Konstanten funktionieren.

   Direkt nach dem Rehash ist diese Eigenschaft offensichtlich wahr ($t = n$). Deshalb schauen wir nach dem w.c., in dem in einem Block $i = 1$ und im nächsten Block $i = n'$ zufällig gewählt wird. Denn so erhält man die maximale Anzahl von aufeinander folgenden Operationen ohne Rehash. Zudem beschränken wir uns auf die Fälle 'nur Insert-Operations' und 'nur Delete-Operations', da diese das Verhältnis zwischen t und n am meisten verändern.

   In dem Fall 'nur Delete-Operations' sind unmittelbar vor dem nächsten Rehash $n - \frac{n}{2} - \frac{n}{4} = \frac{n}{4}$ Elemente enthalten. ($\Rightarrow t \leq 4n_c$)

   In dem Fall 'nur Insert-Operations' sind unmittelbar vor dem nächsten Rehash $n + \frac{n}{2} + \frac{3n}{4} = \frac{9n}{4}$ Elemente enthalten. ($\Rightarrow t \geq \frac{4}{9}n_c$)

   Somit gilt immer $\frac{4}{9}n_c \leq t \leq 4n_c$ und wir können $t \in \Theta(n_c)$ garantieren.