Algorithmen und Datenstrukturen
Tutorium VII

Michael R. Jung

01. - 04. 06. 2015

Betrachten wir folgendes Szenario: Es wird für eine Kontoänderung, eben dieser verändert sowie diese Veränderung gespeichert, alle k Schritte wird außerdem eine Sicherheitskopie der letzten k Veränderungen sowie der neue Kontostand gespeichert. Zusätzlich wird, falls insgesamt 2^j (für jedes j) Veränderungen vorgenommen wurden, eine Konsistenzüberprüfung der kompletten Sicherung vorgenommen. Das heißt also

$$c_i = \begin{cases} 2, & k \nmid i \land i \neq 2^j \ \forall j \\ 2 + (k+1), & k | i \land i \neq 2^j \ \forall j \\ 2 + i, & k \nmid i \land \ \exists j : i = 2^j \\ 2 + (k+1) + i & k | i \land \ \exists j : i = 2^j . \end{cases}$$

Aufgabe 1

Untersuchen Sie dieses Verfahren mit der Account-Methode!

Es gilt:

$$\sum_{i=0}^{n} c_i = \underbrace{2n}_{\substack{\text{jedes Mal}\\ \text{Änderung/}\\ \text{Speicherung}}} + \underbrace{\left\lfloor \frac{n}{k} \right\rfloor (k+1)}_{\substack{\text{alle } k \text{ Schritte werden}\\ k \text{ Operationen und ein}\\ \text{Kontostand gespeichert}}} + \underbrace{\sum_{j=0}^{c} 2^j}_{\substack{c \text{ Konsis-}\\ \text{tenzchecks}}}$$

mit:
$$\begin{aligned} & 2^c \leq n < 2^{c+1} \\ \Leftrightarrow \quad & c \leq \log n < c+1 \\ \Leftrightarrow \quad & c = \lfloor \log n \rfloor \end{aligned}$$

$$\leq 2n + \frac{n}{k} 2k + 2^{c+1} - 1$$

$$\leq 4n + 2 \cdot 2^{\log n}$$

$$= 6n$$

Für geometrische Reihen gilt: $\sum_{i=0}^{n} q^i = \frac{q^{n+1}-1}{q-1}$.

Mit $d_i = 6$ für alle i gilt also für alle n: $\sum_{i=0}^{n} d_i \geq \sum_{i=0}^{n} c_i$.

Betrachten wir nun das ein dynamisches Array T welches seine Größe verdoppelt, wenn bei vollem Array etwas eingefügt wird, und seine Größe halbiert, wenn weniger als ein Viertel seiner Größe genutzt wird. Die Kosten einer Operation belaufen sich auf

- 1 für das normale Löschen oder Einfügen und
- $\mathrm{num}(T)$ für das Kopieren der Elemente von T in ein neues Array.

Aufgabe 2

Zeigen Sie mit Hilfe der folgenden Funktion Φ, dass die amortisierten Kosten von Einfüge- und Löschoperationen konstant sind!

$$\Phi(T) := \begin{cases} 2\,\mathrm{num}(T) - |T|, & \text{falls } \frac{\mathrm{num}(T)}{|T|} \geq \frac{1}{2} \\ \frac{|T|}{2} - \mathrm{num}(T), & \text{sonst.} \end{cases}$$

Wir beobachten zunächst, dass $\Phi \geq 0$ (d.h. $\forall T : \Phi(T) \geq 0$) gilt.

Betrachten wir zunächst die Einfügeoperationen. Es gibt folgende Fälle zu unterscheiden:

1. $\text{num}(T_{i-1}) = |T_{i-1}|$,
2. $\text{num}(T_{i-1}) < |T_{i-1}| \wedge \frac{\text{num}(T_{i-1})}{|T_{i-1}|} \geq \frac{1}{2}$,
3. $\frac{\text{num}(T_{i-1})}{|T_{i-1}|} < \frac{1}{2} \wedge \frac{\text{num}(T_i)}{|T_i|} \geq \frac{1}{2}$ und
4. $\frac{\text{num}(T_i)}{|T_i|} < \frac{1}{2}$.

Ähnliches gilt für die Löschoperationen, hier gibt es folgende Fälle zu unterscheiden:

1. $\frac{\mathrm{num}(T_i)}{|T_i|} \geq \frac{1}{2}$,
2. $\frac{\mathrm{num}(T_i)}{|T_i|} < \frac{1}{2} \wedge \frac{\mathrm{num}(T_{i-1})}{|T_{i-1}|} \geq \frac{1}{2}$,
3. $\frac{1}{4} \leq \frac{\mathrm{num}(T_i)}{|T_{i-1}|} < \frac{\mathrm{num}(T_{i-1})}{|T_{i-1}|} < \frac{1}{2}$ und
4. $\mathrm{num}(T_i) < \frac{1}{4}|T_{i-1}|$.

Einfügen bei $\text{num}(T_{i-1}) = |T_{i-1}|$:

Hier muss nun die Arraygröße verdoppelt werden. Das neue Array ist somit wieder mehr als halb gefüllt.

$$
\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= \underbrace{\text{num}(T_i)}_{\substack{=\text{num}(T_{i-1})+1 \\ =|T_{i-1}|+1}} + 2\,\text{num}(T_i) - \underbrace{|T_i|}_{=2|T_{i-1}|} - 2\,\text{num}(T_{i-1}) + |T_{i-1}| \\
&= (|T_{i-1}| + 1) + 2(|T_{i-1}| + 1) - 2|T_{i-1}| - 2|T_{i-1}| + |T_{i-1}| \\
&= 3
\end{aligned}
$$

Einfügen bei $\mathrm{num}(T_{i-1}) < |T_{i-1}| \land \frac{\mathrm{num}(T_{i-1})}{|T_{i-1}|} \geq \frac{1}{2}$:
Die Arraygröße muss nicht verändert werden ($|T_i| = |T_{i-1}|$).

$$\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= 1 + 2\,\mathrm{num}(T_i) - |T_i| - 2\,\mathrm{num}(T_{i-1}) + |T_{i-1}| \\
&= 1 + 2(\mathrm{num}(T_{i-1}) + 1) - 2\,\mathrm{num}(T_{i-1}) \\
&= 3
\end{aligned}$$

Einfügen bei $\frac{\mathrm{num}(T_{i-1})}{|T_{i-1}|} < \frac{1}{2} \wedge \frac{\mathrm{num}(T_i)}{|T_i|} \geq \frac{1}{2}$:

$$\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= 1 + 2\,\mathrm{num}(T_i) - |T_i| - \frac{|T_{i-1}|}{2} + \mathrm{num}(T_{i-1}) \\
&= 1 + 2(\mathrm{num}(T_{i-1}) + 1) - \frac{3|T_{i-1}|}{2} + \mathrm{num}(T_{i-1}) \\
&= 3 + 3\left(\mathrm{num}(T_{i-1}) - \frac{|T_{i-1}|}{2}\right) \\
&= 3 + 3\left(\frac{\mathrm{num}(T_{i-1})}{|T_{i-1}|}|T_{i-1}| - \frac{|T_{i-1}|}{2}\right) \\
&< 3 + 3\left(\frac{1}{2}|T_{i-1}| - \frac{|T_{i-1}|}{2}\right) \\
&= 3
\end{aligned}$$

Einfügen bei $\frac{\mathrm{num}(T_i)}{|T_i|} < \frac{1}{2}$:

$$\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= 1 + \frac{|T_i|}{2} - \mathrm{num}(T_i) - \frac{|T_{i-1}|}{2} + \mathrm{num}(T_{i-1}) \\
&= 1 - (\mathrm{num}(T_{i-1}) + 1) + \mathrm{num}(T_{i-1}) \\
&= 0
\end{aligned}$$

Löschen bei $\frac{\mathrm{num}(T_i)}{|T_i|} \geq \frac{1}{2}$:

$$\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= 1 + 2\,\mathrm{num}(T_i) - |T_i| - 2\,\mathrm{num}(T_{i-1}) + |T_{i-1}| \\
&= 1 + 2(\mathrm{num}(T_{i-1}) - 1) - 2\,\mathrm{num}(T_{i-1}) \\
&= -1
\end{aligned}$$

Löschen bei $\frac{\text{num}(T_i)}{|T_i|} < \frac{1}{2} \wedge \frac{\text{num}(T_{i-1})}{|T_{i-1}|} \geq \frac{1}{2}$:

$$
\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= 1 + \frac{|T_i|}{2} - \text{num}(T_i) - 2\,\text{num}(T_{i-1}) + |T_{i-1}| \\
&= 1 + \frac{3|T_{i-1}|}{2} - (\text{num}(T_{i-1} - 1) - 2\,\text{num}(T_{i-1}) \\
&= 2 + 3\left(\frac{|T_{i-1}|}{2} - \text{num}(T_{i-1})\right) \\
&= 2 + 3\left(\frac{|T_{i-1}|}{2} - \frac{\text{num}(T_{i-1})}{|T_{i-1}|}|T_{i-1}|\right) \\
&\leq 2 + 3\left(\frac{|T_{i-1}|}{2} - \frac{1}{2}|T_{i-1}|\right) \\
&= 2
\end{aligned}
$$

Löschen bei $\frac{1}{4} \leq \frac{\text{num}(T_i)}{|T_{i-1}|} < \frac{\text{num}(T_{i-1})}{|T_{i-1}|} < \frac{1}{2}$:

$$\begin{aligned} d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\ &= 1 + \frac{|T_i|}{2} - \text{num}(T_i) - \frac{|T_{i-1}|}{2} + \text{num}(T_{i-1}) \\ &= 1 - (\text{num}(T_{i-1}) + -1) + \text{num}(T_{i-1}) \\ &= 2 \end{aligned}$$

Löschen bei $\text{num}(T_i) < \frac{1}{4}|T_{i-1}|$:
Hier muss nun das Array halbiert werden ($|T_i| = \frac{1}{2}|T_{i-1}|$), es ist dann wieder weniger als halb gefüllt.

$$\begin{aligned}
d_i &= c_i + \Phi(T_i) - \Phi(T_{i-1}) \\
&= (\text{num}(T_i) + 1) + \frac{|T_i|}{2} - \text{num}(T_i) - \frac{|T_{i-1}|}{2} + \text{num}(T_{i-1}) \\
&= (\text{num}(T_i) + 1) + \frac{|T_{i-1}|}{4} - (\text{num}(T_{i-1}) - 1) - \frac{|T_{i-1}|}{2} + \text{num}(T_{i-1}) \\
&= \text{num}(T_i) - \underbrace{\frac{|T_{i-1}|}{4}}_{>\text{num}(T_i)} + 2 \\
&< 2
\end{aligned}$$

Es gilt also für alle $i : d_i \leq 3$ und folglich für beliebige Abfolgen von n Eingabe- und Löschoperationen:

$$\sum_{i=0}^{n} c_i \leq \sum_{i=0}^{n} d_i \leq 3n.$$