Assignment 8
Advanced Algorithms & Data Structures PS

Christian Müller 1123410
Daniel Kocher, 0926293

June 1, 2016

Aufgabe 16

Geben Sie für die Operationen über die Datenstruktur aus der Aufgabe 15 eine Potentialfunktion an und zeigen Sie mit Hilfe dieser Potentialfunktion, dass die amortisierten Kosten einer Operation konstant sind.

Op_i	c_i
1	2
2	2 (teuer)
3	3 (teuer)
4	2
5	5 (teuer)
6	2
7	2
8	2
9	9 (teuer)
10	2
11	2
12	2
13	2
14	2
15	2
16	2
17	17 (teuer)
18	2

Table 1: Kosten für $1 \leq i \leq 18$

$$c_i = \begin{cases} i, \text{ falls } i = 2^k + 1 \text{ mit } k \in \mathbb{N} \\ 2, \text{ sonst.} \end{cases}$$

Die Potentialfunktion sei folgendermaßen definiert:

$$\phi(D_0) = 0 \qquad \phi(D_i) = 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 \tag{1}$$

Diese Potentialfunktion $\phi(D_i)$ entspricht genau dem Kredit nach der i-ten Operation (für $i > 0$) wenn man die Bankkonto-Methode anwendet.
Einige Gleichungen, die wir im Zuge der Berechnung der amortisierten Kosten benutzen werden:

$$2^{k+1} = 2(i-1), \text{ wenn } i = 2^{k+1} \tag{2}$$

$$2^k = (i-1), \text{ wenn } i \neq 2^{k+1} \tag{3}$$

$$\lfloor ld(i-1) \rfloor = \lfloor ld(i-2) \rfloor, \text{ wenn } i \neq 2^{k+1} \tag{4}$$

Gleichung 4 gilt aufgrund der Tatsache, dass es für jedes beliebige k zwischen $2^k + 1$ und $2^{k+1} + 1$ immer ein i gibt, sodass $i > 2^k + 1$ gilt.

Für alle Operationen i mit $i = 2^k + 1$ gilt also:

$$\phi(D_i) = 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 = 2i - 2^{k+1} + 1 \tag{5}$$

Mit Gleichung 2 erhält man dann

$$\phi(D_i) = 2i - 2(i-1) + 1 = 2i - 2i + 2 + 1 = 3 \tag{6}$$

Für alle Operationen i mit $i \neq 2^k + 1$ gilt:

$$\phi(D_i) = 2i - (i-1) + 1 = 2i - i + 2 = i + 2 \tag{7}$$

Überlegung: Unser Potential steigt also zwischen den Operationen $i = 2^{j-1} + 1$ und $k = 2^j + 1$ stetig an, um dann nach der Operation $u = 2^j$ seinen Maximalwert zu erreichen, welcher von der Operation $k = 2^j + 1$ aufgebraucht wird.

Für die amortisierten Kosten gilt:

- Für alle Operationen i mit $i = 2^k + 1$:

$$\begin{aligned}\hat{c_i} &= c_i + \phi(D_i) - \phi(D_{i-1}) \\ &= i + 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 - \left(2(i-1) - 2^{\lfloor ld((i-1)-1) \rfloor + 1} + 1\right) \\ &= 3i - 2^{k+1} + 1 - \left(2i - 2 - 2^k + 1\right) \\ &\overset{\text{Gl. 2}}{=} 3i - 2(i-1) + 1 - 2i + 2 + 2^k - 1 \\ &\overset{\text{Gl. 3}}{=} i - 2i + 2 + 1 + 2 - 1 + (i-1) \\ &= 3 \leq 3 \Rightarrow \text{ in } O(1)\end{aligned}$$

- Für alle Operationen i mit $i \neq 2^k + 1$:

$$\begin{aligned}\hat{c_i} &= 2 + 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 - \left(2(i-1) - 2^{\lfloor ld(i-2) \rfloor + 1} + 1\right) \\ &= 2 + 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 - 2i + 2 + 2^{\lfloor ld(i-2) \rfloor + 1} - 1 \\ &\overset{\text{Gl. 4}}{=} 4 \leq 4 \Rightarrow \text{ in } O(1)\end{aligned}$$

$$\Longrightarrow \hat{c_i} = \begin{cases} 3, \text{ wenn } i = 2^k + 1 \\ 4, \text{ sonst} \end{cases} \in O(1) \tag{8}$$

Der Vollständigkeit halber sei noch kurz bewiesen, dass unsere Potentialfunktion $\phi(D_i)$ nicht negativ ist:

$$\phi(D_i) = 2i - 2^{\lfloor ld(i-1) \rfloor + 1} + 1 \geq 2i - 2(i-1) + 1 = 2i - 2i + 2 + 1 = 3 \geq 0 \tag{9}$$

Aufgabe 17

Sei Q eine Binomial Queue, die anfangs genau einen Binomialbaum B_1 mit den Schlüsseln 13 und 21 enthält. Fügen Sie die Schlüssel 3, 7, 15, 18 und 27 in die Queue ein. Beachten Sie dabei, dass beim Einfügen eines Schlüssels k in eine Binomial Queue Q zuerst eine Binomial Queue Q' mit einem Objekt (mit Schlüssel k) erzeugt wird und anschließend Q und Q' vereinigt werden. Geben Sie nach jedem Schritt die resultierende Queue an (verwenden Sie dabei die Child-Sibling-Parent Darstellung).

Im Folgenden werden die einzelnen Schritte dargestellt, wobei für die Child-Sibling-Parent Darstellung die folgenden Pfeile verwendet werden:

$\longrightarrow$ . . . Child-Pointer $\dashrightarrow$ . . . Parent-Pointer $\cdots\!\rightarrow$ . . . Sibling-Pointer

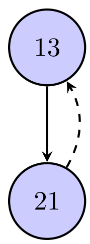

Figure 1: Ausgangs-Queue Q

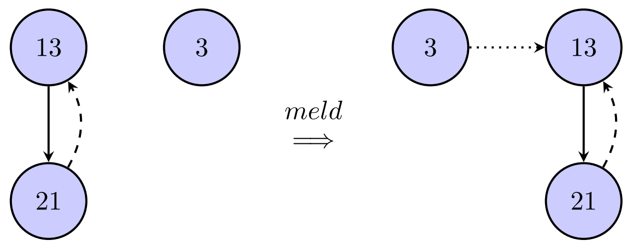

Figure 2: Einfügen von 3 (*meld*: B_0 vor B_1)

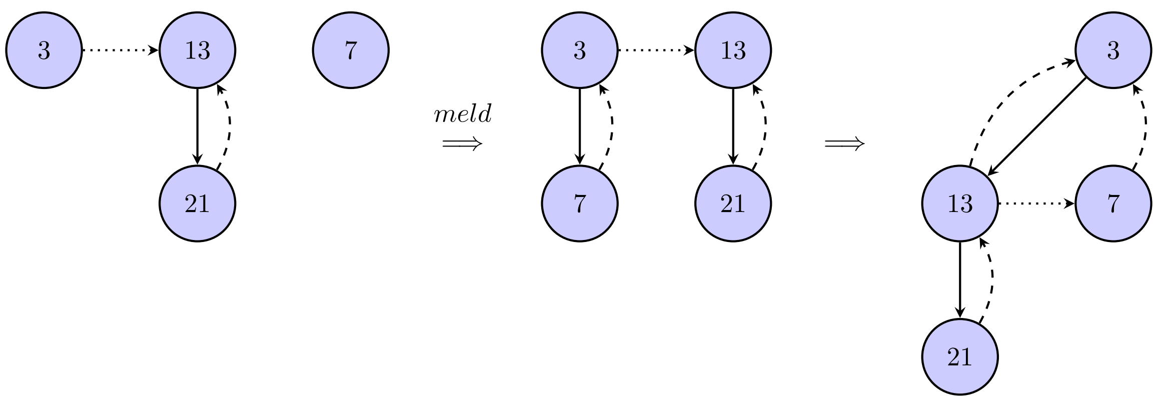

Figure 3: Einfügen von 7 (*meld* vereinigt zweimal: B_0 und B_1)

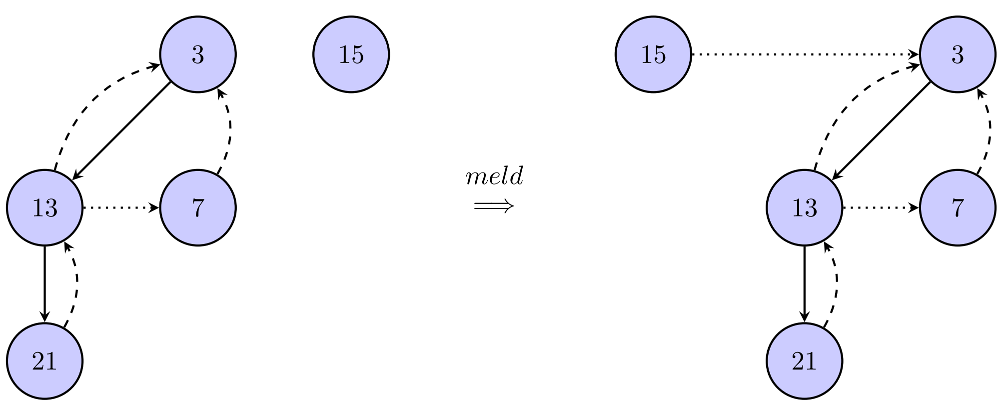

Figure 4: Einfügen von 15 (*meld*: B_0 vor B_2)

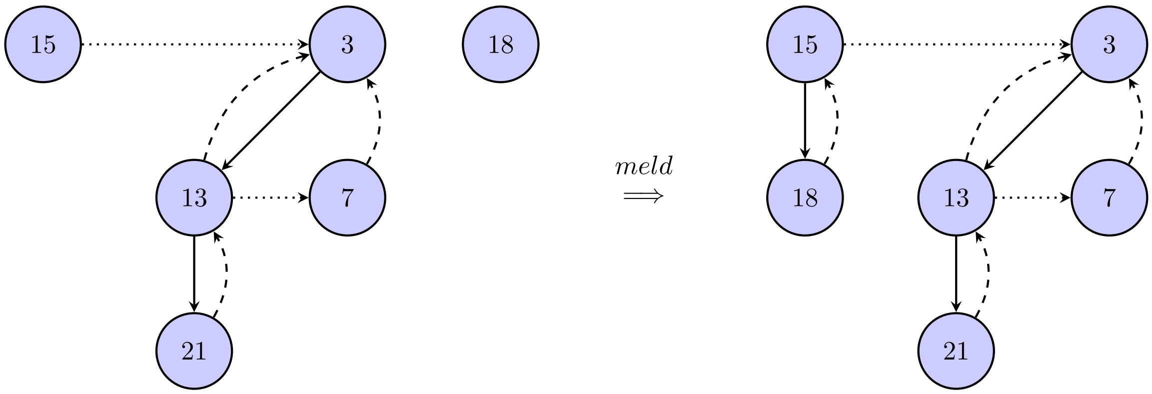

Figure 5: Einfügen von 18 (*meld* vereinigt einmal: B_0)

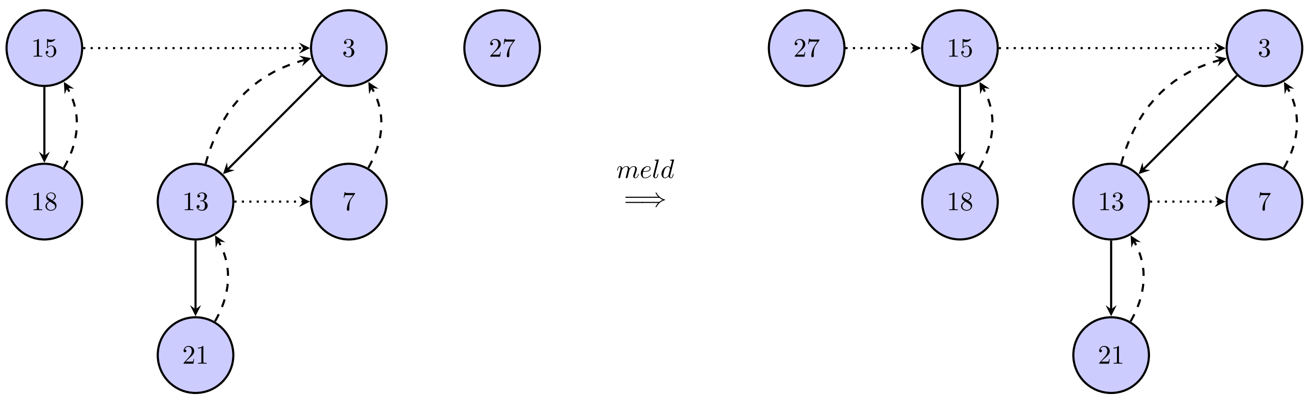

Figure 6: Einfügen von 27 (*meld*: B_0 vor B_1 vor B_2)