Algorithms Theory, Solution for Assignment 4

http://lak.informatik.uni-freiburg.de/lak_teaching/ws09_10/algo0910.php

Exercise 4.1 - Amortized Analysis

1.

n	Costs	Cost Sum	Average Costs
1	1	1	1
2	2	3	1.5
3	1	4	1.33
4	4	8	2
5	1	9	1.8
6	1	10	1.67
7	1	11	1.57
8	8	19	2.38
9	1	20	2.22
10	1	21	2.1
11	1	22	2
12	1	23	1.91
13	1	24	1.85
14	1	25	1.79
15	1	26	1.73
16	16	42	2.62
⋮	⋮	⋮	⋮

It holds $A(n) = \sum_{i=1}^{n} c_i \overset{!}{\leq} 3n$. Without loss of generality we assume that $n \in \{2^x \mid x \in \mathbb{N}\}$ then,

$$\begin{aligned}
A(n) &= \sum_{i=1}^{n} c_i \\
&= \underbrace{\sum_{\substack{i=1 \\ i \in \{2^x \mid x \in \mathbb{N}\}}}^{n} i}_{1+2+4+8+16+\ldots+2^{\log n}} + \sum_{\substack{i=1 \\ i \notin \{2^x \mid x \in \mathbb{N}\}}}^{n} 1 \\
&= \sum_{i=1}^{\log n} 2^i + \sum_{\substack{i=1 \\ i \notin \{2^x \mid x \in \mathbb{N}\}}}^{n} 1 \\
&= 2^{\log n + 1} - 1 + \sum_{\substack{i=1 \\ i \notin \{2^x \mid x \in \mathbb{N}\}}}^{n} 1 \\
&\leq 2n - 1 + n \\
&\leq 3n
\end{aligned}$$

2. Define $\Phi'_i := \Phi_i - \Phi_0$ for all i. Then holds that $a_i = a'_i$. Thus,

$$\begin{aligned}
a'_i &= t_i + \Phi'_i - \Phi'_{i-1} \\
&= t_i + \Phi_i - \Phi_0 - \Phi_{i-1} + \Phi_0 \\
&= t_i + \Phi_i - \Phi_{i-1} \\
&= a_i
\end{aligned}$$

Also $\Phi'_0 = 0$ and $\Phi'_i = \Phi_i - \Phi_0 \geq 0$ since $\Phi_i \geq \Phi_0$.

Exercise 4.2 - Ternary counter

1. Let $c = \frac{3}{2}$.
   By means of the potential method we show that for $c = \frac{3}{2}$, it holds that $A(n) \le cn$ for all $n \in \mathbb{N}$.
   Let $T_i \in \{0,1,2\}^*$ be the counter's state at point i and $|T_i|_k$ be the number of digits k in T_i.
   Then, $\Phi_i = \frac{1}{2}|T_i|_1 + |T_i|_2$
   Let b_i denote the number of 2 digits an the end from T_{i-1}.

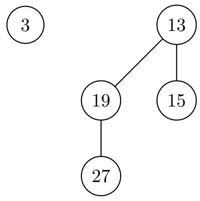
   - **Case 1:** There is a 0 before the last sequence of digits in b_i. (i.e. $T_{i-1} = 0/1/2\ldots0/1/2\,0\,\underbrace{2\ldots2}_{b_i}$).
     Then after 1 increment we have $T_i = 0/1/2\ldots0/1/2\;1\;\underbrace{0\ldots0}_{b_i\;0-en}$. Therefore, $\Phi_i = \Phi_{i-1} - b_i + \frac{1}{2}$.
   - **Case 2:** There is a 1 before the last sequence of digits in b_i. (i.e. $T_{i-1} = 0/1/2\ldots0/1/2\,1\,\underbrace{2\ldots2}_{b_i}$).
     Then after 1 increment we have $T_i = 0/1/2\ldots0/1/2\;2\;\underbrace{0\ldots0}_{b_i\;0-en}$. Therefore, $\Phi_i = \Phi_{i-1} - b_i + 1 - \frac{1}{2} = \Phi_{i-1} - b_i + \frac{1}{2}$.

   In both cases the potential is $\Phi_i = \Phi_{i-1} - b_i + \frac{1}{2}$.
   The actual costs for both cases are: $t_i = b_i + 1$.
   Then, we have the amortized costs for operation $a_i = t_i + \Phi_i - \Phi_{i-1} = b_i + 1 + \phi_{i-1} - b_i + \frac{1}{2} - \Phi_{i-1} = \frac{3}{2}$.
   Since $a_i = t_i + \Phi_i - \Phi_{i-1}$, we have $\sum_{l=0}^{n} a_l = \sum_{l=0}^{n} t_l + \sum_{l=0}^{n}(\Phi_l - \Phi_{l-1}) = \sum_{l=0}^{n} t_l + \underbrace{\Phi_l}_{\ge 0} - \underbrace{\Phi_0}_{=0}$
   and therefore, $cn = \sum_{l=0}^{n} a_l \ge \sum_{l=0}^{n} t_l = A(n)$

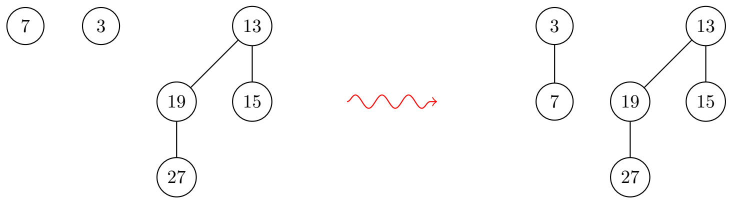
2. We show that for each $c = \frac{3}{2} - \epsilon(\epsilon > 0)$ there exists an n such that $A(n) > (\frac{3}{2} - \epsilon)n$.
   Let $\delta = \lceil -\log_3 \epsilon \rceil \Rightarrow \delta \ge -\log_3 \epsilon \Rightarrow -\delta \le \log_3 \epsilon \Rightarrow 3^{-\delta} \le \epsilon$
   Choose $n = 3^\delta$
   $\Rightarrow T_n = 1\underbrace{0\ldots0}_{\delta 0-en}$
   While counting until n the first position will be incremented exactly $1 = 3^0$ times, the second position $3 = 3^1$ times, ..., the last position will be incremented 3^δ times.
   Then $A(n) = \sum_{k=0}^{\delta} 3^k = 3^{\delta+1} - 1$.
   In addition, since $3^{-\delta} \le \epsilon$, we have that $(c - \epsilon)n \le (c - 3^{-\delta})n$.
   Therefore; it follows that $(c - \epsilon)n \le (\frac{3}{2} - 3^{-\delta})3^\delta = \frac{3^{\delta+1}}{2} - 1 < 3^{\delta+1} - 1 = A(n)$

Exercise 4.3 - Binomial queues

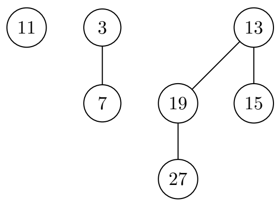
1. - Q.insert(3)

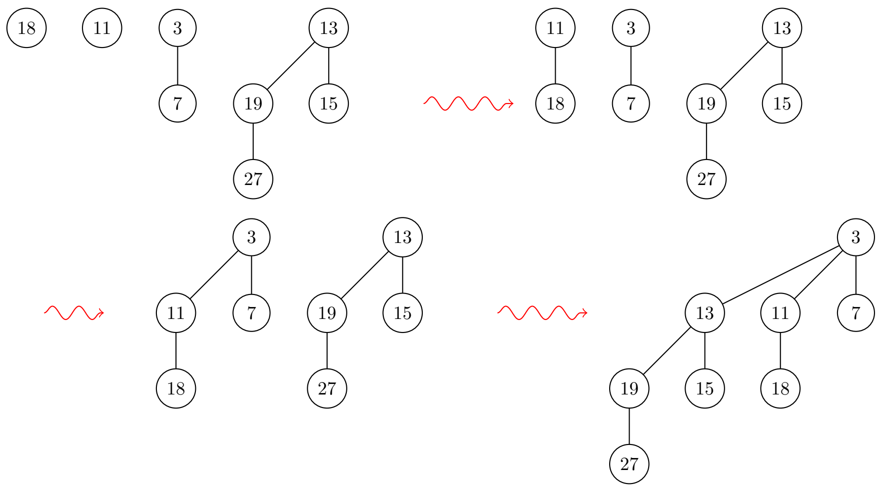
   - Q.insert(7)



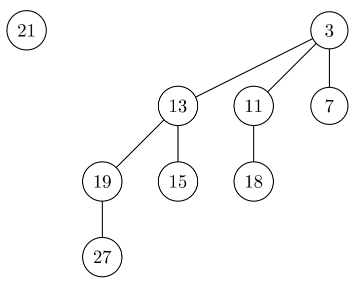
- Q.insert(11)



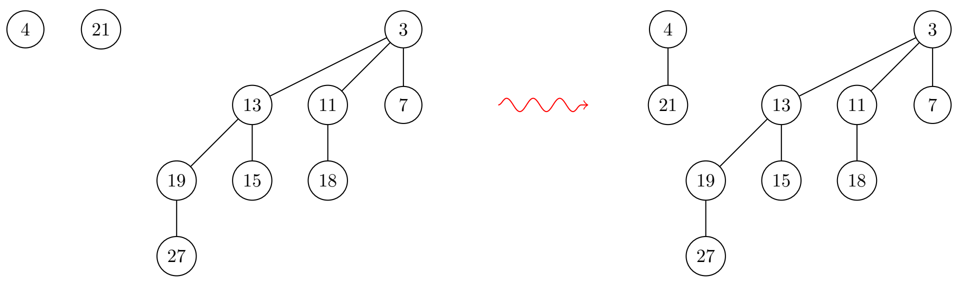
- Q.insert(18)



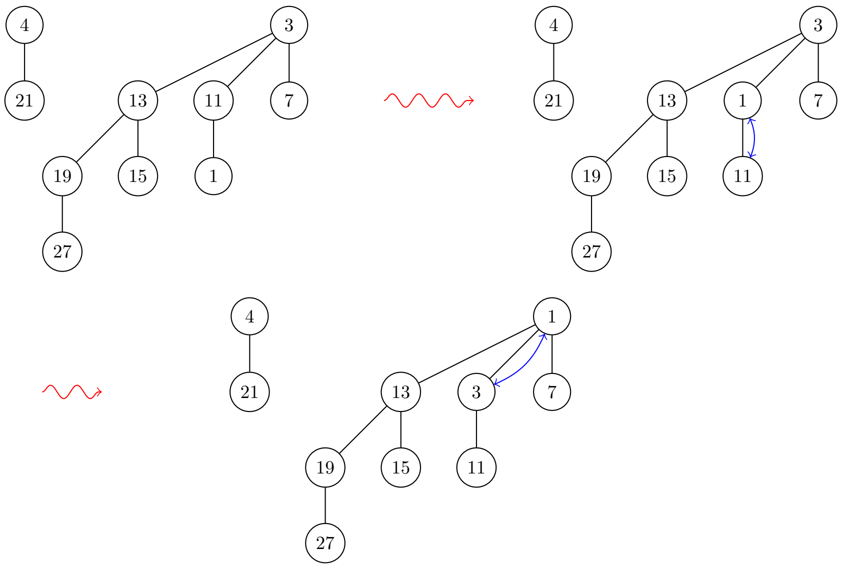
- Q.insert(21)



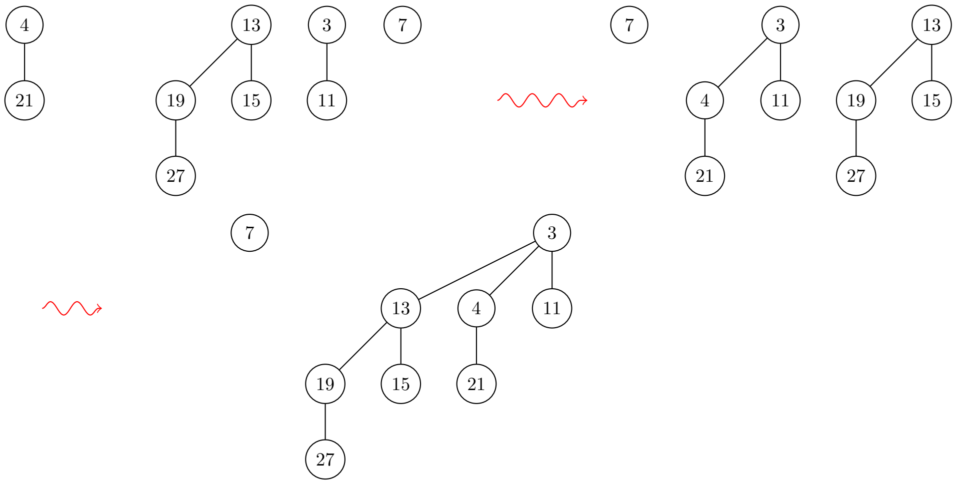
- Q.insert(4)



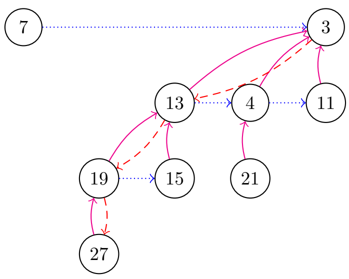
- Q.decreaseKey(18,1)



- Q.deleteMin()



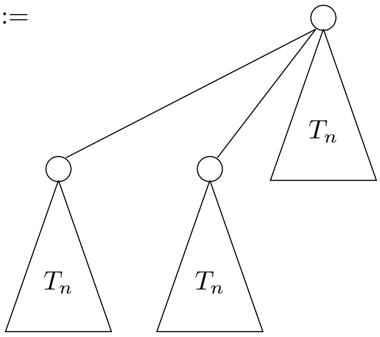
2. Child-Sibling-Representation.



Exercise 4.4 - Trinomial Trees

Trinomial trees are defined recursively by

$T_0 :=$ ○ $\quad T_{n+1} :=$



.

We prove by induction over the structure of the tree that the following properties hold

1. Number of Nodes in T_i is 3^i. Let $|T_i|$ denote the number of nodes in T_i.

   For $i = 0$ it holds: $|T_0| = 1 = 3^0$ by definition of T_0.

   The induction step from i to $i + 1$: The tree T_{i+1} contains three trees T_i. Hence T_{i+1} have $3 \cdot |T_i| = 3 \cdot 3^i = 3^{i+1}$ nodes.

2. The degree of the root is 2^i. Let $\deg(T_i)$ denote the degree of the root of T_i.

   For $i = 0$ it holds: The degree of the root is $\deg(T_0) = 0 = 2i$.

   The induction step $i \to i + 1$: $\deg(T_{i+1}) = \deg(T_i) + 2 = 2 \cdot i + 2 = 2(i + 1)$.

3. The height of T_i is i. Let $\text{height}(T_i)$ denote the height of T_i.

   For $i = 0$ is holds $\text{height}(T_0) = 0 = i$. For $i \to i + 1$: $\text{height}(T_{i+1}) = \text{height}(T_i) + 1 = i + 1$.

4. The number of nodes with depth k is $F_i(k) := 2^k \binom{i}{k}$.

   For $i = 0$ it holds that $F_0(k) = 1$ for $k = 0$ and $F_0(k) = 0$ otherwise. Since T_0 has only one node with depth 0 the property holds and the induction base follows.

   Induction step, $i \to i + 1$:

   For a fixed k, count the number of nodes in T_{i+1}. By induction hypothesis it holds that for T_i there are $F_i(k)$ nodes with depth k. Hence in T_{i+1} we have two times the number of nodes for depth $k - 1$ ($F_i(k - 1)$) and once the number of nodes of depth k ($F_i(k)$).

$$
\begin{aligned}
F_{i+1}(k) &= 2 \cdot F_i(k-1) + F_i(k) \\
&= 2 \cdot 2^{k-1} \binom{i}{k-1} + 2^k \binom{i}{k} \\
&= 2^k \left(\binom{i}{k-1} + \binom{i}{k} \right) \\
&\overset{(1)}{=} 2^k \binom{i+1}{k}
\end{aligned}
$$

   (1) holds by the definition of $\binom{n}{k}$ (Pascal triangle):

$$\binom{n}{k} := \binom{n-1}{k-1} + \binom{n-1}{k}, \binom{n}{0} = \binom{n}{n} := 1$$

In order to save $n = (b_{\lceil \log_3 n \rceil} \cdots b_1 b_0)_3$ keys in a trinomial queue, where $b_i \in \{0, 1, 2\}$ we first notice that we allow two trees T_i instead of one to be part of our queue. We define the structure of the queue by:

- $b_i = 0$ iff T_i is not part of the queue
- $b_i = 1$ iff T_i is part of the queue once
- $b_i = 2$, iff T_i is part of the queue twice

Since every trinomial tree stores 3^i keys, we can store n keys in such a way, because

$$n = \sum_{i=1}^{n} b_i 3^i \qquad .$$

The meld operation will merge the trees if we have more than two T_i in our queue.