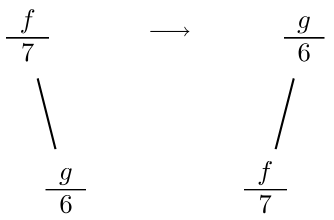
Algorithms Theory, Solution for Assignment 3

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

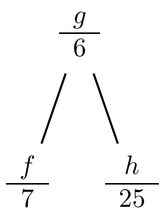
Exercise 3.1 - Treaps

1. Insert f: $\frac{f}{7}$

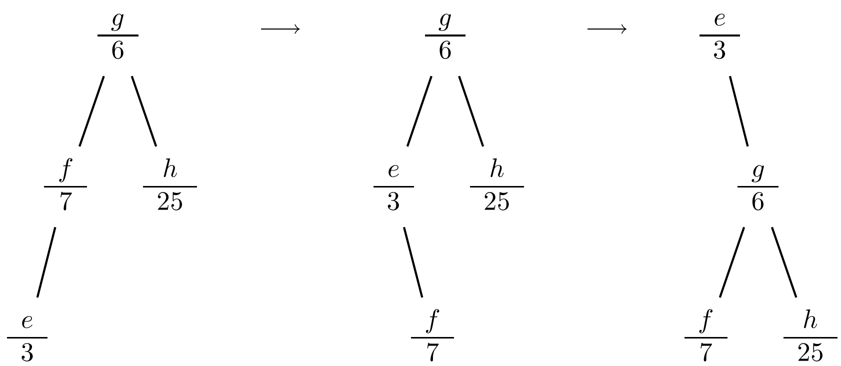
Insert g:



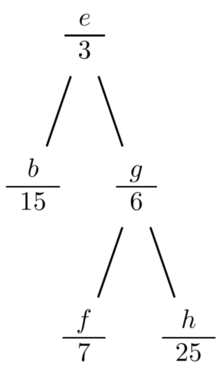
Insert h:



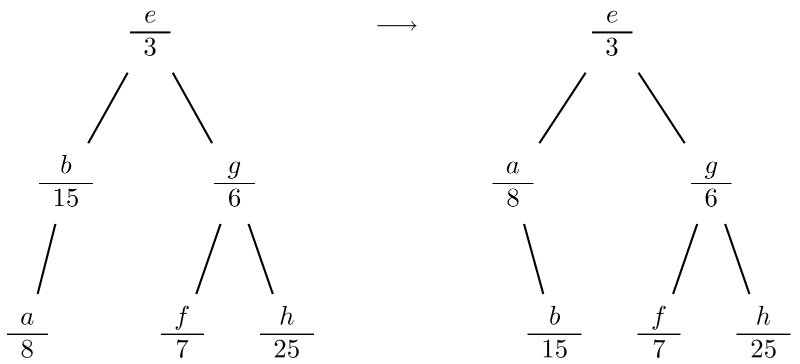
Insert e:



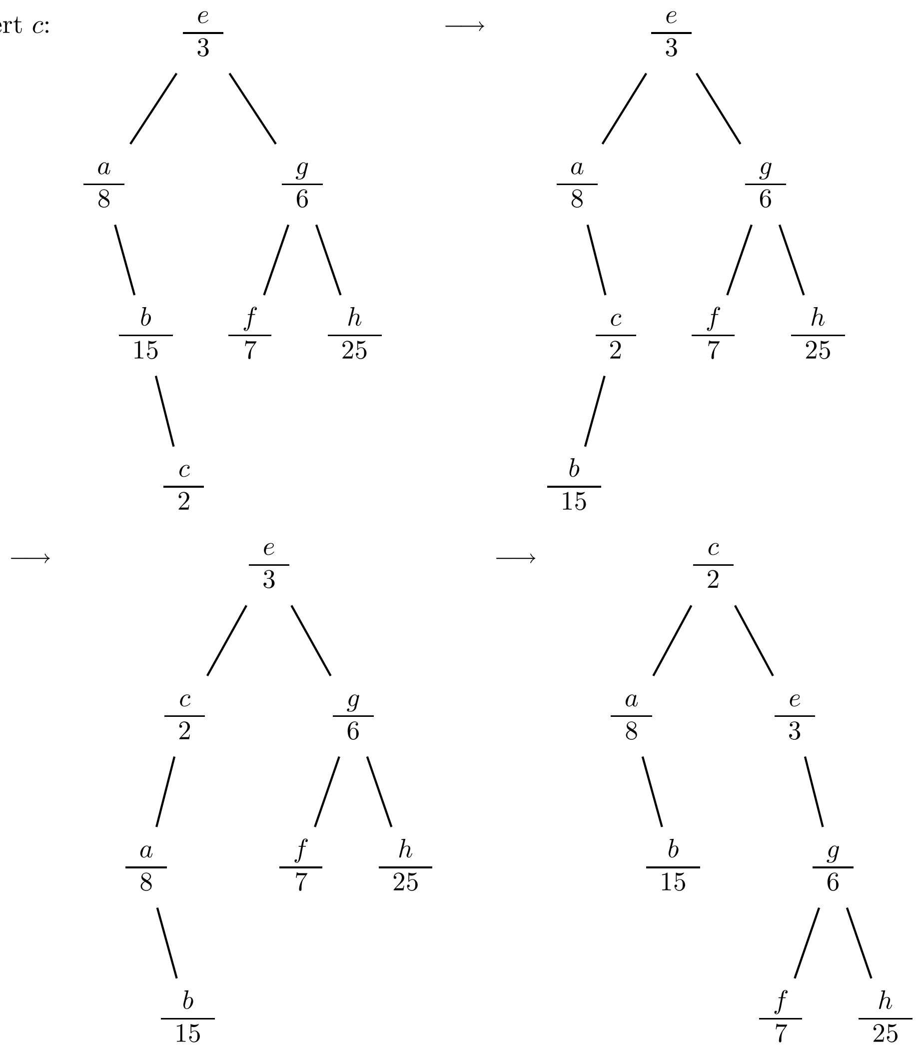
Insert b:



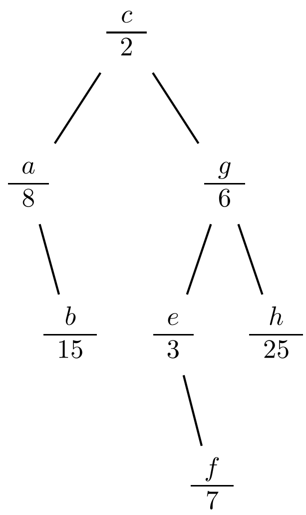
Insert a:



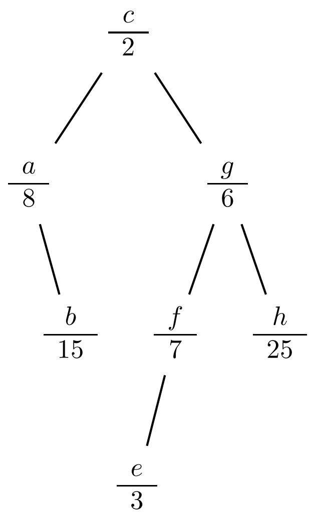
Insert c:



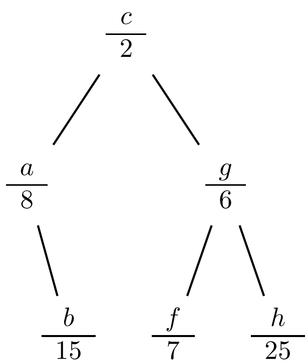
2. Delete e: $\longrightarrow$



$\longrightarrow$

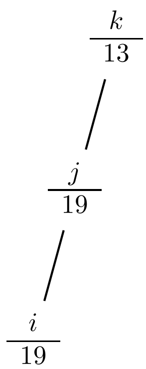


$\longrightarrow$



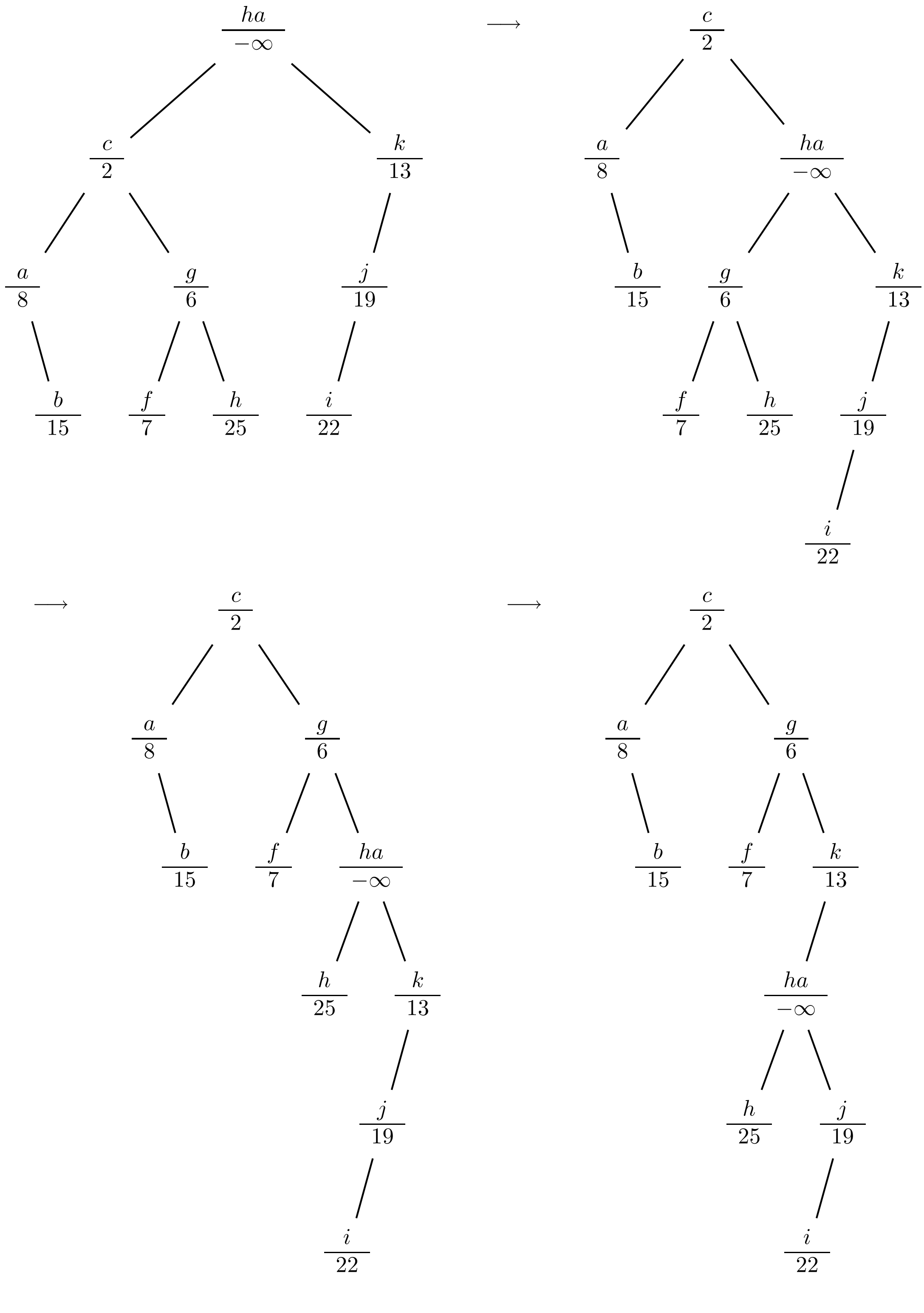
(1)

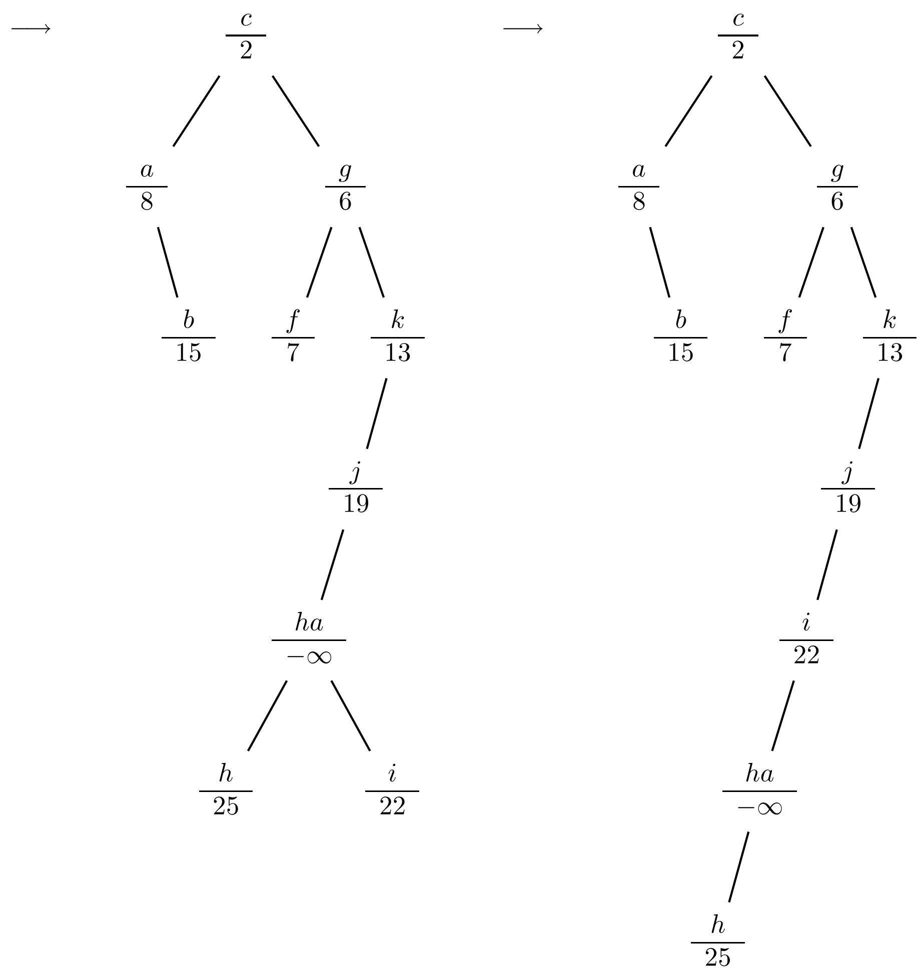
3. The result for insert i, k, j is:

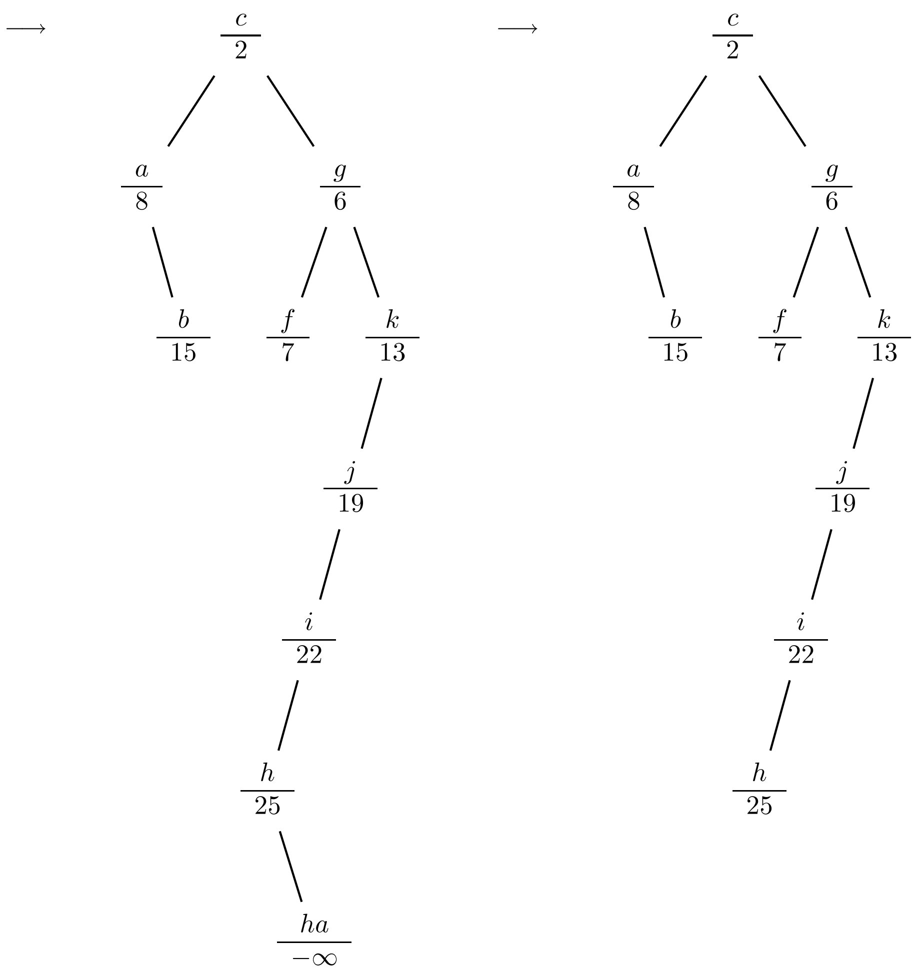


(2)

We choose ha with $prio(ha) = -\infty$ as root, and the two heaps (1) and (2) as children. Afterwards we remove ha from the treap.

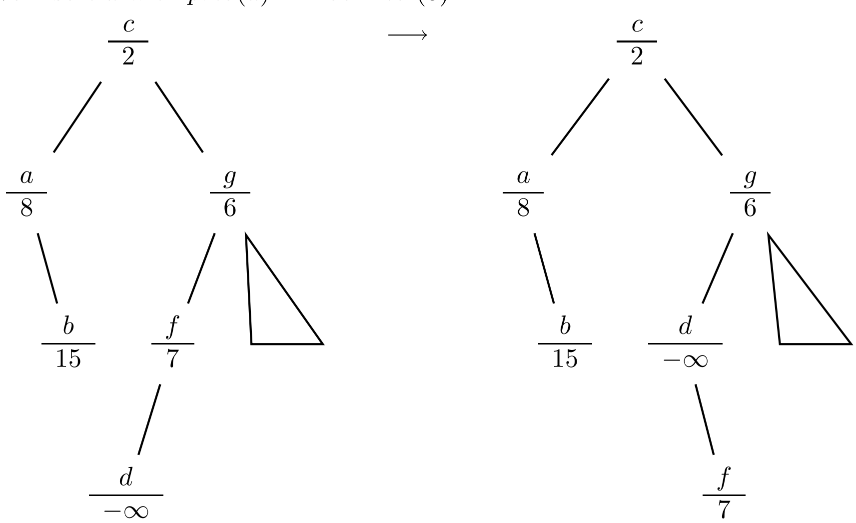

ha −∞
c 2
k 13
a 8
g 6
j 19
b 15
f 7
h 25
i 22
c 2
a 8
ha −∞
b 15
g 6
k 13
f 7
h 25
j 19
i 22
c 2
a 8
g 6
b 15
f 7
ha −∞
h 25
k 13
j 19
i 22
c 2
a 8
g 6
b 15
f 7
k 13
ha −∞
h 25
j 19
i 22

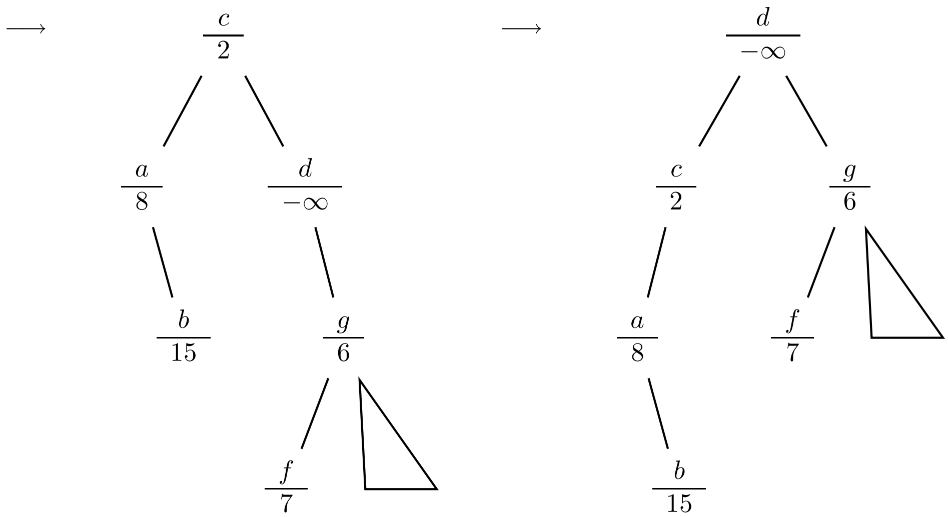

⟶
c 2
a 8
g 6
b 15
f 7
k 13
j 19
ha −∞
h 25
i 22
⟶
c 2
a 8
g 6
b 15
f 7
k 13
j 19
i 22
ha −∞
h 25



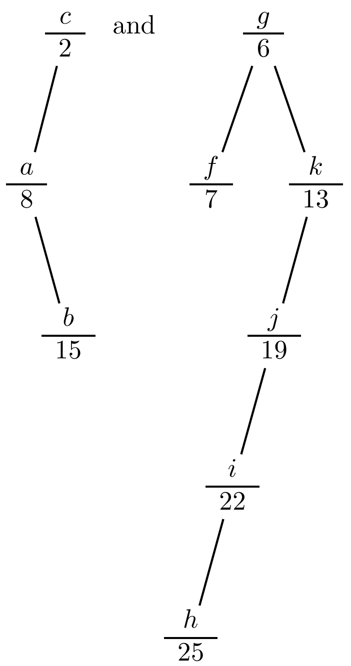
(3)

4. We insert d with $prio(d) = -\infty$ into (3).





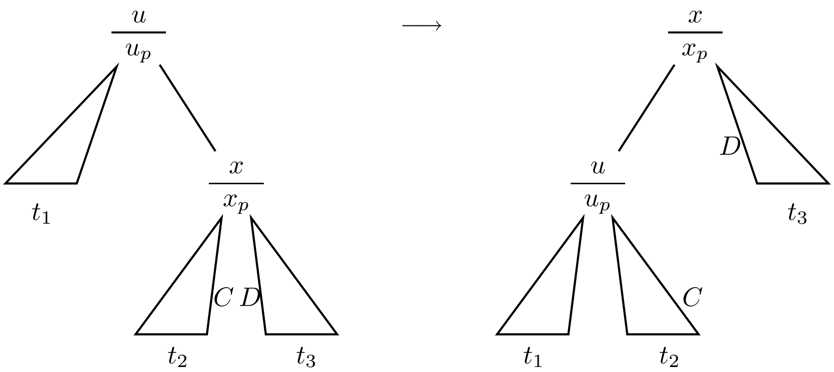
Hence we get the two treaps:



Exercise 3.2 - Treaps

1. Proving that $C + D$ is equal to the number of rotations, we have a look at the treap, when an arbitrary number of rotations is already done. We assume that $C + D$ holds, and we prove that $C + D$ holds for the treap after a rotation, (if necessary).

   Consider the following treap:



The picture shows that the length of the new right spline of the subtree of x is $C' = C + 1$, since it contains all elements the right spine of t_2 and u, while the length of the left spine t_3 remains unchanged.

Hence, after the rotation $C' + D' = C + D + 1$.

The other rotation is symmetric, there it holds $C' = C$ and $D' = D + 1$, hence $C' + D' = C + D + 1$.

Because $C + D$ is initially 0, after all rotations are done, $C + D$ is equal to the number of rotations that was needed to move x to the correct position in the treap.

2. We insert x with priority ∞ into T_S. The new Treap $T_{S'}$ is the treap with elements $S' = S \cup \{x_1, \ldots, x_m, x, x_{m+1}, \ldots, x_n\} = \{x'_1, \ldots, x'_{n+1}\}$.

   The expected number of nodes on the search path of x in $T_{S'}$ is equal to the number of visited nodes for searching x in T_S.

   Define

$$X_{k,i} = \begin{cases} 1 & x'_i \text{ is ancestor of } x'_k \\ 0 & \text{otherwise} \end{cases}$$

   Let X_k be the length of the path from the root to x'_k.

$$X_{m+1} = \sum_{i \in \{1,\ldots,n+1\}} X_{m+1,i} = \sum_{i<m+1} X_{m+1,i} + \sum_{i>m+1} X_{m+1,i}$$
$$\Rightarrow E[X_{m+1}] = \sum_{i<m+1} E[X_{m+1,i}] + \sum_{i>m+1} E[X_{m+1,i}]$$

   Since for all x: $prio(x'_{m+1}) > prio(x)$

$$i < m+1: \qquad E[X_{m+1,i}] = P[x'_i \text{ is ancestor of } x'_{m+1}] = P[P_{min}(\{x'_i, \ldots, x'_{m+1}\}) = x'_i]$$
$$= P[P_{min}(\{x'_i, \ldots, x'_m\}) = x'_i] = \frac{1}{m - i + 1}$$
$$i > m+1: \qquad E[X_{m+1,i}] = P[x'_i \text{ is ancestor of } x'_{m+1}] = P[P_{min}(\{x'_{m+1}, \ldots, x'_i\}) = x'_i]$$
$$= P[P_{min}(\{x'_{m+2}, \ldots, x'_i\}) = x'_i] = \frac{1}{i - (m+2) + 1}$$

$$E[X_{m+1}] = \sum_{i<m+1} \frac{1}{m-i+1} + \sum_{i>m+1} \frac{1}{i-(m+2)+1} = \sum_{i=1}^{m} \frac{1}{i} + \sum_{i=1}^{n-m} \frac{1}{i} = H_m + H_{n-m}$$

Exercise 3.3 - Universal hashing

$$\mathcal{H} = \{h_i(k) = ((a_i * k + b_i) \mod N-1) \mod m\} \text{ for } i \in \{1, \ldots, N(N-1)\}$$

Since $a_i = 40 \cdot i$ and $b_i = 60 \cdot i$ are always even numbers, we can represent each $h_i \in \mathcal{H}$ as $h_i(k) = (2 * (\frac{a_i}{2} \cdot k + \frac{b_i}{2}) \mod N-1) \mod m$. In addition, since N is a prime number, $N-1$ is an even number. Thus, every $h_i \in \mathcal{H}$ yields an even number. Therefore, the total number of cells used by $\mathcal{H}$ is $\frac{m}{2}$.

Now, let $\delta_h(x, y)$ and $x, y \in U$ be defined as in the lecture (slide 17.) as follows:

$$\delta_h(x, y) = \begin{cases} 1 & h(x) = h(y) \text{ and } x \neq y \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{H}$ is universal, if the following condition holds:

$$\frac{|\{h \in \mathcal{H} : h(x) = h(y)\}|}{|\mathcal{H}|} \leq \frac{1}{m}$$

$$\longrightarrow \quad |\{h \in \mathcal{H} : h(x) = h(y)\}| \leq \frac{|\mathcal{H}|}{m}$$

$$\longrightarrow \quad |\{h \in \mathcal{H} : h(x) = h(y)\}| \leq \frac{N(N-1)}{m}$$

$$\longrightarrow \quad \sum_{h \in \mathcal{H}} \delta_h(x, y) \leq \frac{N(N-1)}{m}$$

$$\longrightarrow \quad \frac{N(N-1)}{\frac{m}{2}} \leq \frac{N(N-1)}{m}$$

$$\longrightarrow \quad \frac{2 \cdot N(N-1)}{m} \not\leq \frac{N(N-1)}{m}$$

Which implies that $\mathcal{H}$ is not universal.

In order to ensure $\mathcal{H}$ to be universal, we modify each $h_i \in \mathcal{H}$ with the help of the theorem from slide 18, as follows. $a_i = (i-1) \mod (N-1) + 1$, $b_i = \lfloor \frac{i-1}{N-1} \rfloor$ and $h_i(k) = ((a_i * k + b_i) \mod N) \mod m$. Now, $\mathcal{H}$ is a Universal class of hash functions, since it holds:

$$\{(a_i, b_i) \mid i \in \{1, \ldots, N(N-1)\}\} = \{(a, b) \mid a \in \{1, \ldots, N-1\}, b \in \{0, \ldots, N-1\}\}$$

Exercise 3.4 - Perfect hashing

$U = \{0, \ldots, 30\}$, $S = \{2, 4, 7, 11, 12, 18, 19, 21, 26, 28\}$ and $k = 3, N = 31, n = |S| = 10$.

1. First hash function $h_3(x) = ((3x) \mod 31) \mod 10$

i	$W_i = \{x \in S : h_3(x) = i\}$	b_i	$m_i = 2b_i(b_i - 1) + 1$	s_i
0		0	1	0
1	7,21	2	5	1
2	4,11,28	3	13	6
3	18	1	1	19
4		0	1	20
5	12	1	1	21
6	2,19,26	3	13	22
7		0	1	35
8		0	1	36
9		0	1	37

Now compute k_i, such that $h_{k_i}(x) = ((k_i x) \mod 31) \mod m_i$ restricted to W_i is injective.

- W_1

x	$h_1(x)$
7	2
21	1

For $k_1 = 1$, h_{k_1} restricted to W_1 is injective.

- W_2

x	$h_{k_2}(x)$
4	4
11	11
28	2

For $k_2 = 1$, h_{k_2} restricted to W_2 is injective.

- W_6

x	$h_{k_2}(x)$
2	2
19	6
26	0

For $k_6 = 1$, h_{k_6} restricted to W_6 is injective.

Then we obtain the following hash functions for the second level.

$$\begin{aligned} h_{k_1} &= (x \mod 31) \mod 5 \\ h_{k_2} &= (x \mod 31) \mod 13 \\ h_{k_6} &= (x \mod 31) \mod 13 \\ h_{k_j} &= (x \mod 31) \mod 1 \quad \text{for } j \in \{0, \ldots, n-1\} \setminus \{1, 2, 6\} \end{aligned}$$

2. Positions of the elements $x \in S$ in the hash table: $pos(x) = s_i + j$ where $i = h_k(x), j = h_{k_i}(x)$

x	$pos(x)$
2	24
4	10
7	3
11	17
12	21
18	19
19	28
21	2
26	22
28	8