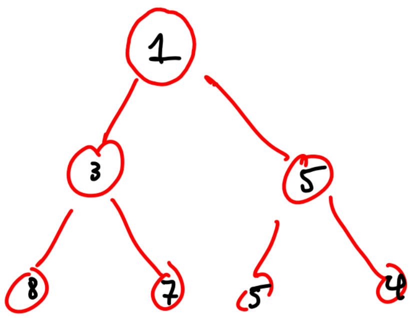
G54 AAD

L05 Leftist Heaps

This lecture: PFDS Ch 3.1

Next lecture: PFDS Ch 3.2

Def: A (min) <u>heap</u> is a tree such that the value stored at a node is no larger than values stored in child nodes (heap property).



Applications

- Dijkstra's algorithm
- Heap Sort
- ...

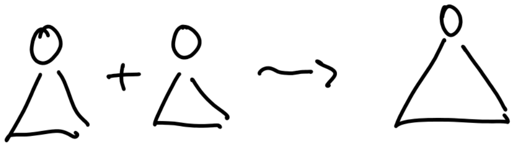
This lecture: A special type of heap called Leftist Heaps

Operations on Heaps

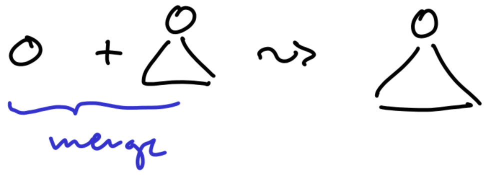
findMin - find smallest element

$O(1)$

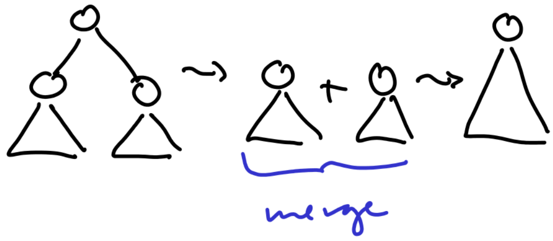
merge - merge two heaps



insert - insert new element



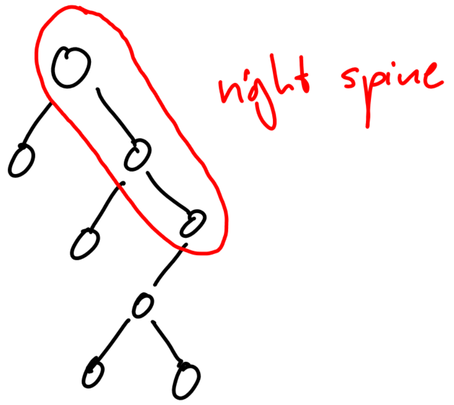
delMin - remove smallest element

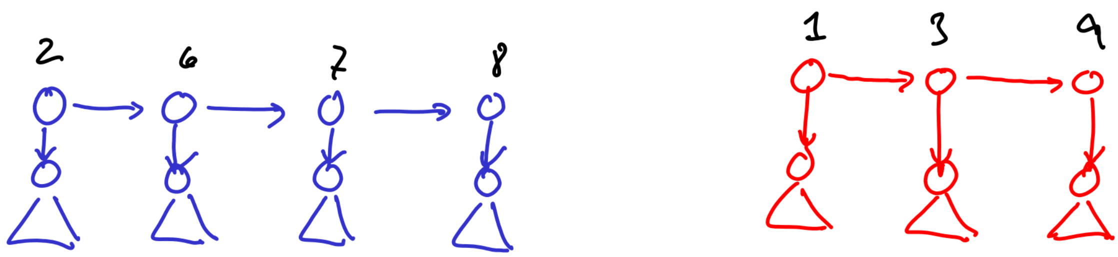


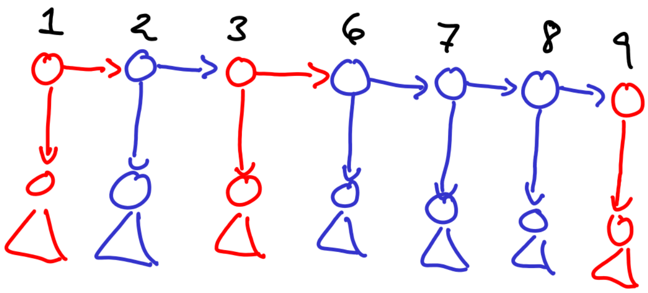
$\Rightarrow$ insert and deleteMin implemented by merge

How to merge two Heaps?

Idea: merge along some fixed path, eg right spine







$\Rightarrow$ Sorting maintains heap property

How to avoid long right spines?

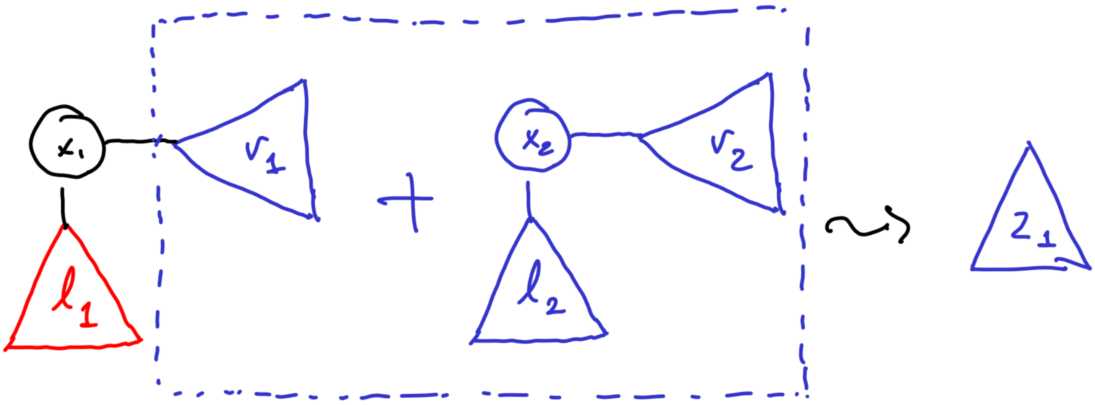
$\Rightarrow$ Leftist Heap Property:

For any node with left node l and right node r

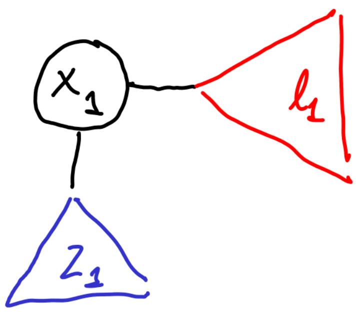
$$\mathrm{rank}(l) \geq \mathrm{rank}(r)$$

where $\mathrm{rank}(x)$ is the size of the right spine of x.

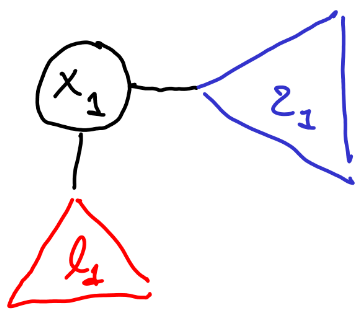
To ensure leftist property (assume $x_1 \leq x_2$)

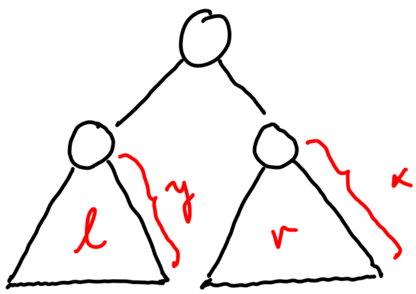


Case a rank $l_1 \leq$ rank z_1



Case b rank $z_1 \leq$ rank l_1





Exercise 3.1

The right spine of a leftist heap of size n contains at most $\lfloor \log(n+1) \rfloor$ elements.

Proof by induction on n, where $n=1$ is trivial.

By the induction hypothesis, we have

$$x \leq \lfloor \log(r+1) \rfloor \tag{1}$$

$$y \leq \lfloor \log(l+1) \rfloor \tag{2}$$

By the leftist property we have

$$x \leq y \tag{3}$$

Hence, we have

$$\begin{aligned}
x &\leq \lfloor \log(\min\{l, r\} + 1) \rfloor \\
&\leq \left\lfloor \log\left(\frac{n-1}{2} + 1\right) \right\rfloor \\
&= \left\lfloor \log\left(\frac{n+1}{2}\right) \right\rfloor \\
&= \lfloor \log(n+1) - \log 2 \rfloor \\
&= \lfloor \log(n+1) \rfloor - 1
\end{aligned}$$

and the theorem is proved.