Data Structures and Algorithms

Mohammad GANJTABESH

Department of Computer Science,
School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.

mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

min/max - Heap ⟹ Min-Max-Heap.

Double Ended Priority Queue (Deap)

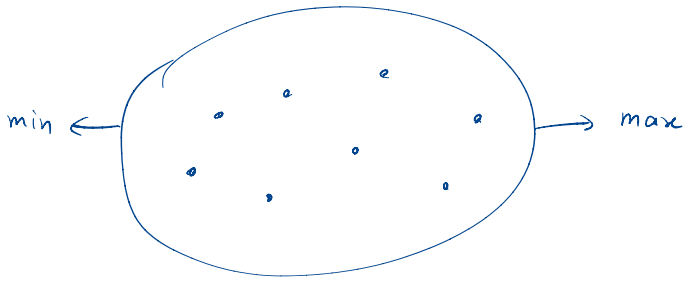
Data Structures and Algorithms
Undergraduate course

Mohammad GANJTABESH
mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Motivations

- Access min and max in constant time
- Adjustment of order $O(\log_2 n)$.



Operations and Naive Implementations

Min, Max, Add, RemoveMin, RemoveMax, and other methods for adjustment.

- Array
- Sorted Array
- Linked List
- Sorted Linked List

Tree Implementation: Properties

Deap:

- has a heap structure (Complete Binary Tree)
- root is always empty. (storing in array is appropriate)
- root's left(right) subtree is a min-heap (max-heap)
- each node in left subtree is smaller than its correspondence in right subtree.
- Access to the min and max is in constant time
- Much more simpler and easier to maintain than the Min-Max-Heap.

Example :

$$Level(i) = \lfloor \log_2 i \rfloor + 1$$

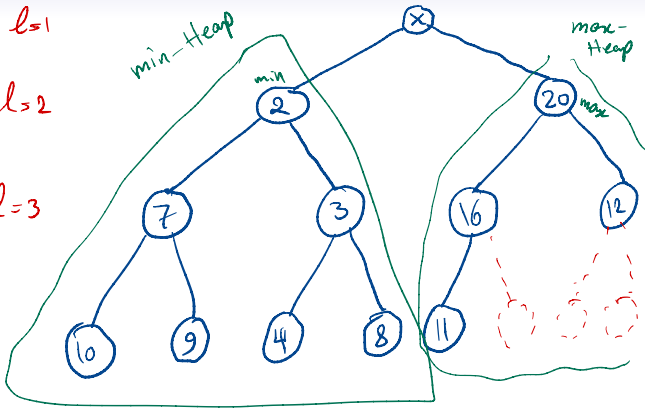
element in level l $= 2^{l-1}$

$$corr(x) = \begin{cases} x + 2^{l-2} & \text{if exist,} \\ \lfloor \frac{x}{2} \rfloor + 2^{l-3} & \text{o.w.} \end{cases}$$

$corr(x)$ ↓ $x \in$ left subtree

$$corr(y) = y - 2^{l-2}$$

$corr(y)$ ↓ $y \in$ right subtree



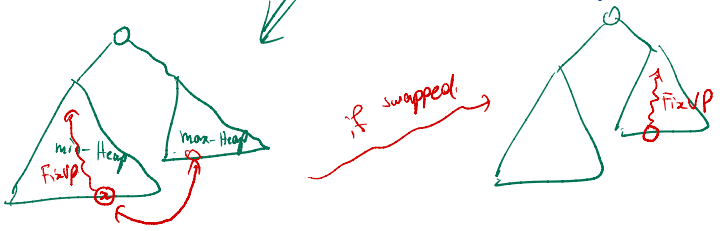
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H	x	2	20	7	3	16	12	10	9	4	8	11			

Tree Implementation

Min ⇝ $H[2]$ is the minimum.

Max ⇝ $H[3]$ is the maximum.

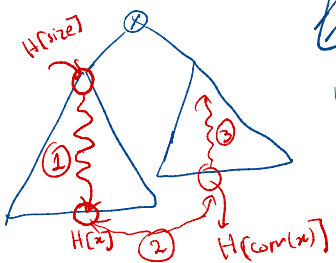
Add(x) ⇝
- Append x to the end of array H.
- swap(x, corr(x)) if necessary.
- FixUp(x) in the corresponding subtree where x belongs to.



Tree Implementation

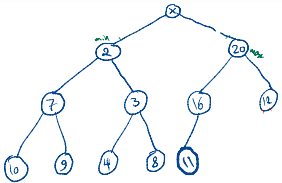
RemoveMin ⟶
- swap(H[2], H[size])
- FixDown(2) in the left subtree
- swap(H[x], H[corr(x)]) if necessary and perform FixUp on the right subtree.

final position of H[size] after FixDown.



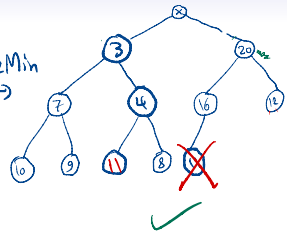
RemoveMax ⟶
- swap(H[3], H[size])
- FixDown(3) in the right subtree
- swap(H[x], H[corr(x)]) if necessary and perform FixUp on the left subtree

Example :



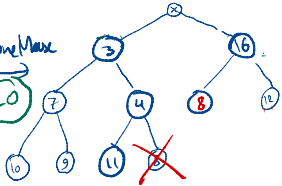
RemoveMin

2



RemoveMax

20



Add (1)