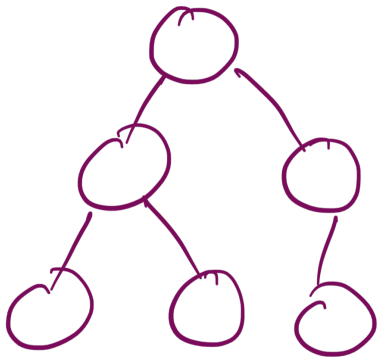
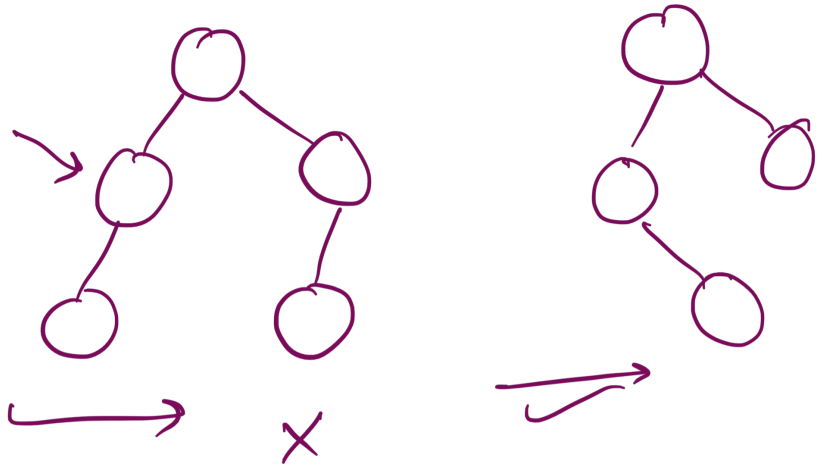
Heaps - I

Basics of Heaps



Complete Binary Tree.

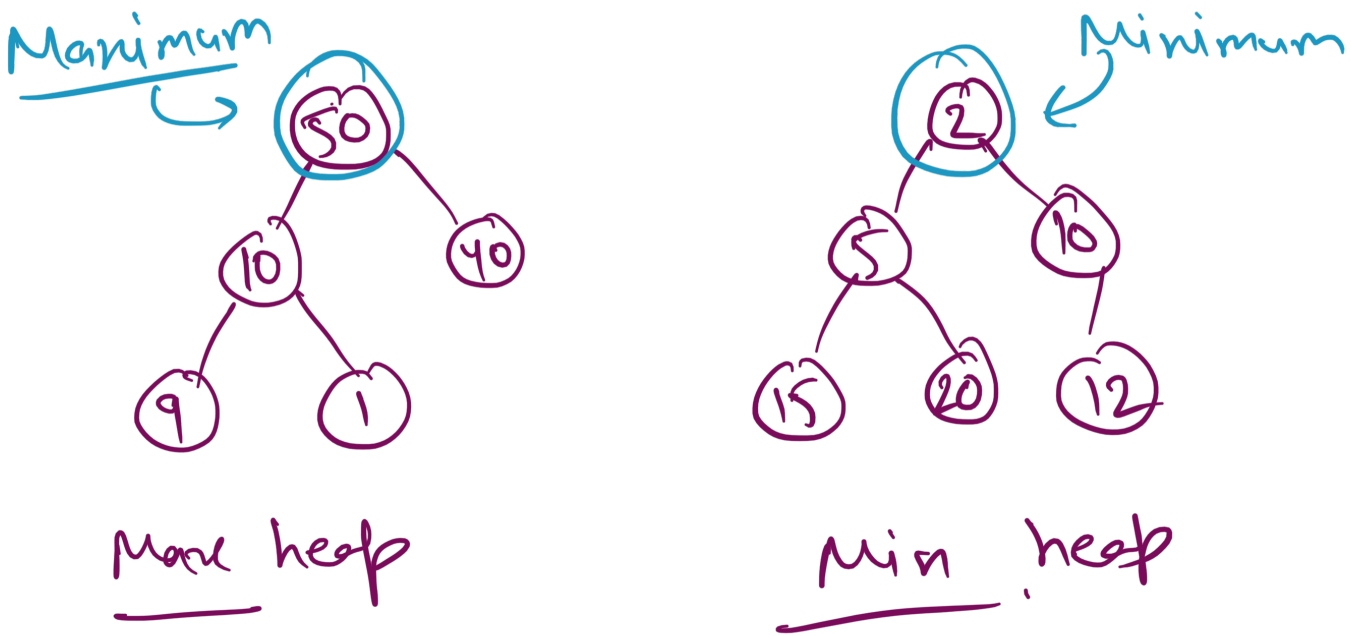


* The level order traversal of a complete binary tree should not have null nodes in b/w.

Definition of a Heap

Node's child are

- Either both smaller (Max heap)
- or both larger (Min heap)



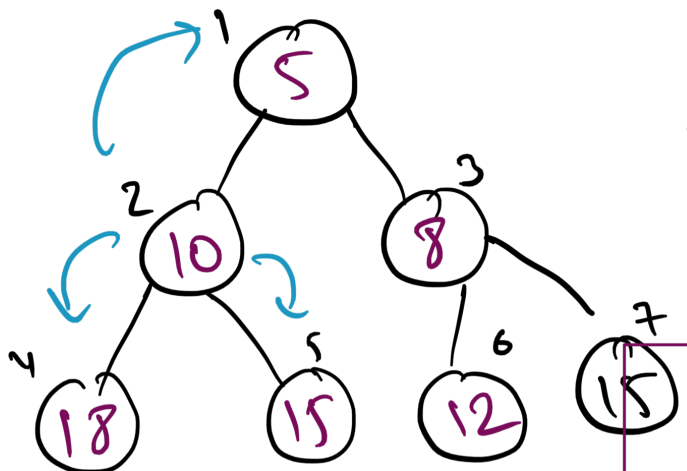
Max heap

Min heap

A heap is a complete binary tree

it should follow minheap or maxheap property.

Representation of a Min heap.



n = 7

	5	10	8	18	15	12	15	
0	1	2	3	4	5	6	7	8

Node (i)

Parent (i) = $\lfloor i/2 \rfloor$ (for $i > 1$)

left (i) = $2 * i$

right (i) = $2 * i + 1$

$2/2 = 1$ parent

$2*2 = 4$ left

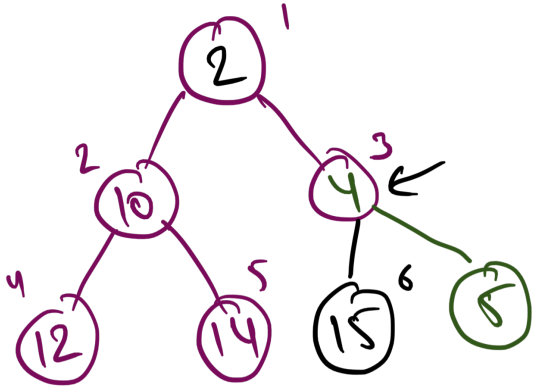
$2*2+1 = 5$ right

$i = 3 \rightarrow 8$

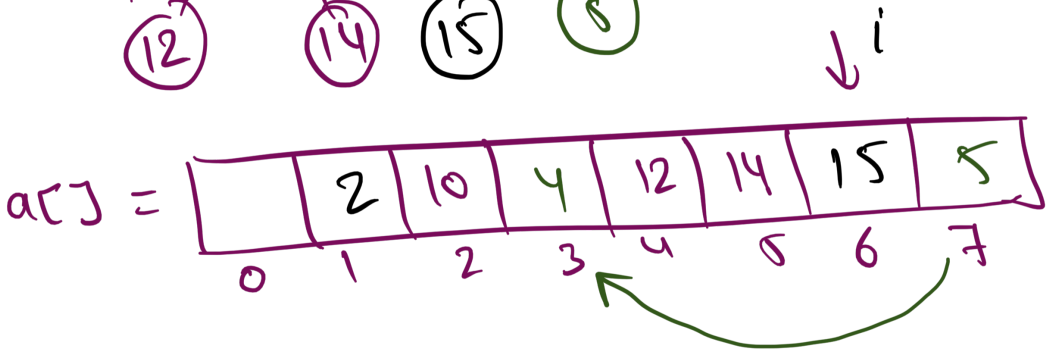
parent(3) = $3/2 = 1$ | left(3) = 2*3 = 6 | right(i) = 2*3+1 = 7

Insertion in Heaps

Min Heap



n = ~~5~~ ~~6~~ 7
Capacity = 8



insert(2)

$h = \log n$

$O(\log n)$

n $\quad 1, 2, 4, 8 \cdots$

$1 + 2 + 4 + 8 \cdots k \text{ terms} = n$

$k = \log_2 n$

leaf

n

$\downarrow$

$\frac{n}{2}$

$\downarrow$

$n/4$

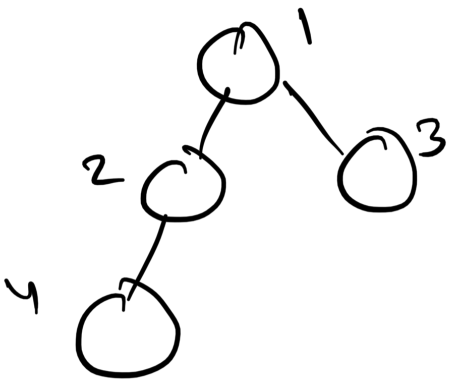
$\vdots$

root $\rightarrow$ $\frac{n}{2^k} = 1$

$\frac{n}{2^k} = 1$

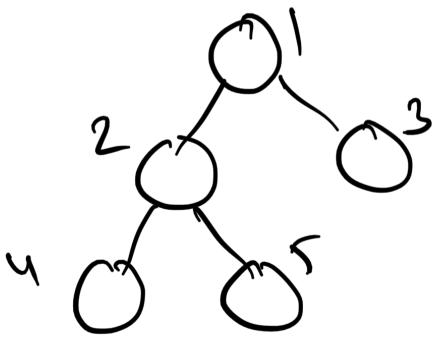
$\boxed{k = \log_2 n}$

Leaf Nodes start from $\frac{n}{2}+1$ to n



$\frac{4}{2}+1$ to 4

3 to 4



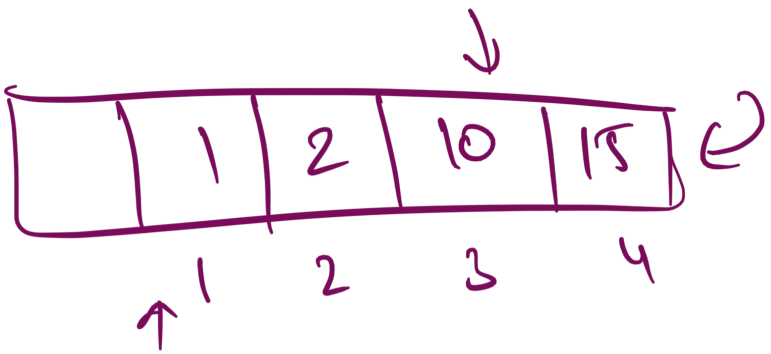
$\frac{5}{2}+1$ to 5 $\Rightarrow$ 3 to 5.

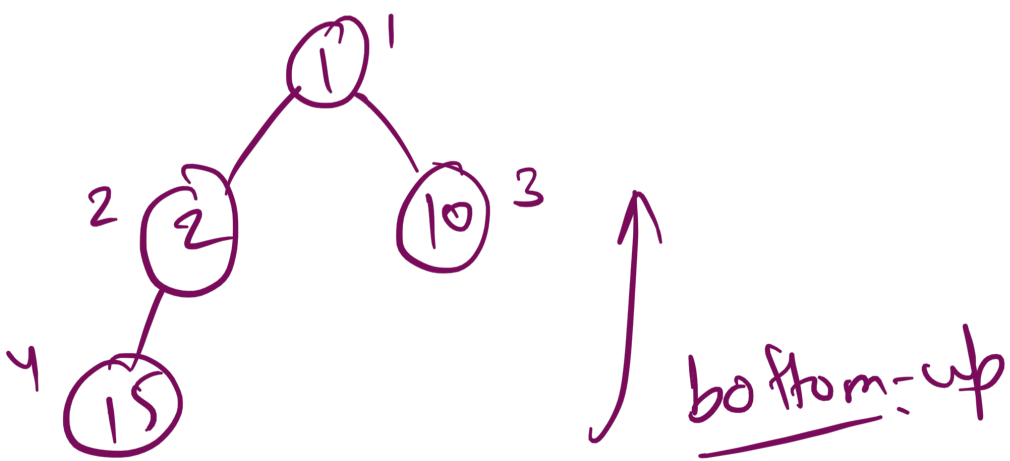
2, 15, 10, 1

$n = 4$ 3

$cp = 4$

$i = 4 \times 1$

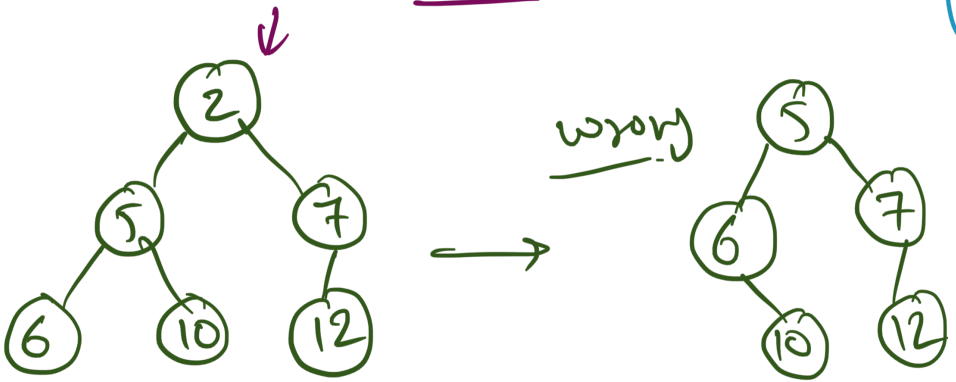




Deletion in Heaps & Heapify

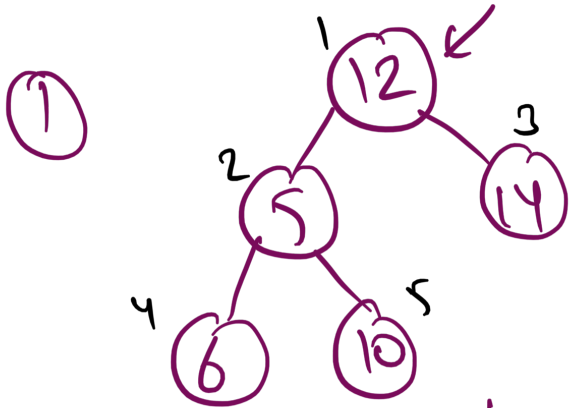
a[1].

extractMin()
↳ delete(1)



(1) replace (delete index) with last element
(2) delete the last element.
(3) heapify (delete index)

O(log n)

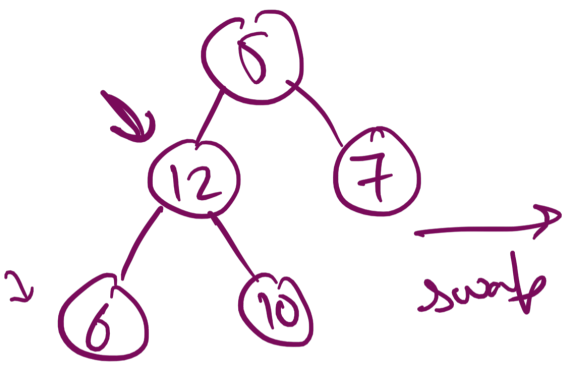


(1)

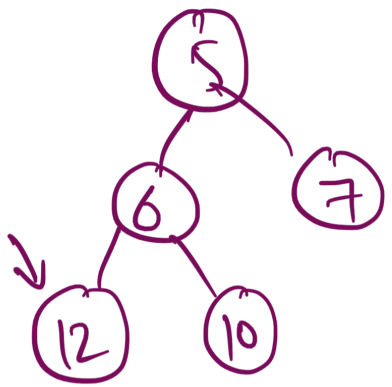
(2) n - -

(3) top down

swap



swap



i = 4

left = 8

right = 9

smallest = 4

Practice Problems

1. Read about the Heap Sort Algorithm
2. Read about the PriorityQueue Class in Java
3. Implement Insert and Delete for MaxHeap.