

ADT Priority Queue

void enqueue (P prio, V value)
V dequeue ()
V getFront ()
P getFrontPrio ()
int getSize ()
bool isEmpty ()

P — priority
V — value

AVL Tree ^(uniqueness?)

enq

$O(\log n)$

sorted
LL
 $O(n)$

unsorted
LL
 $O(1)$

deq

$O(\log n)$

$O(1)$

$O(n)$

getFr

$O(\log n)$

$O(1)$

$O(n)$

getFrPrio

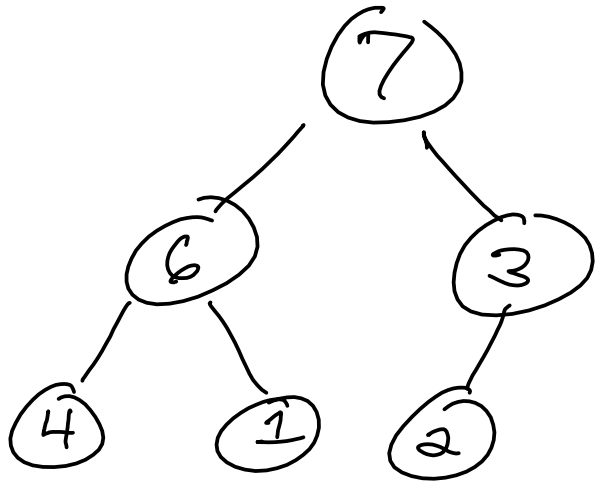
$O(\log n)$

$O(1)$

$O(n)$

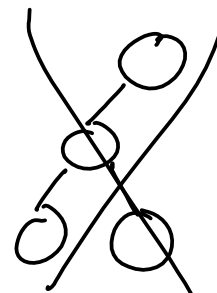
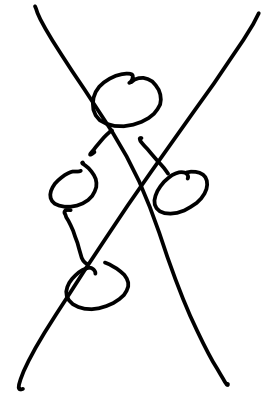
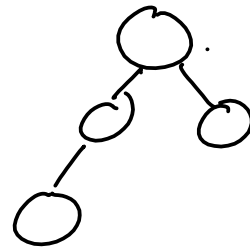
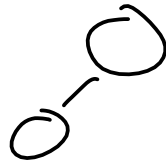
Max Heap

is a complete binary tree w/ invariant: all children have values \leq parents

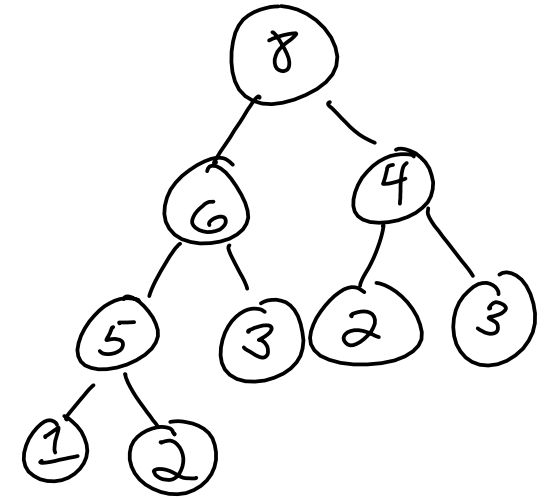
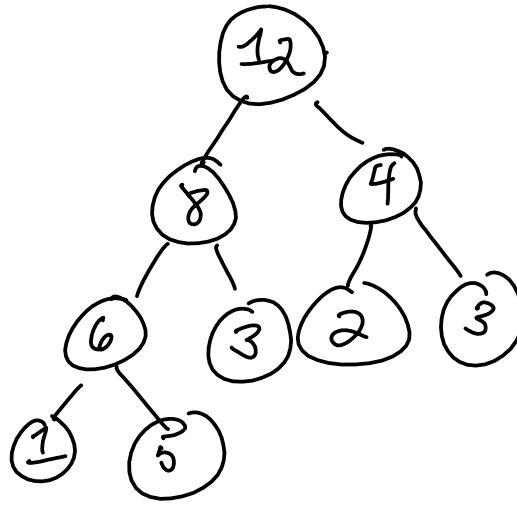
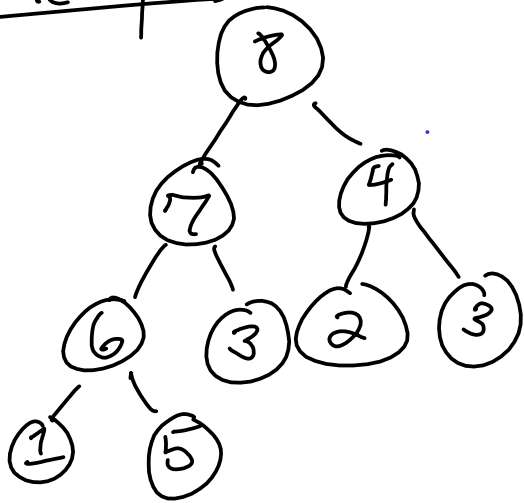


Complete tree:

all levels except last are full AND all nodes in last level are aligned to left



Bubble up

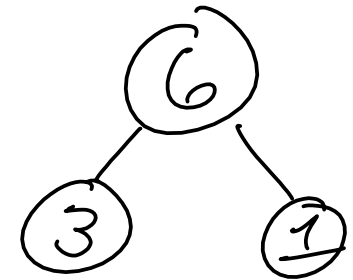


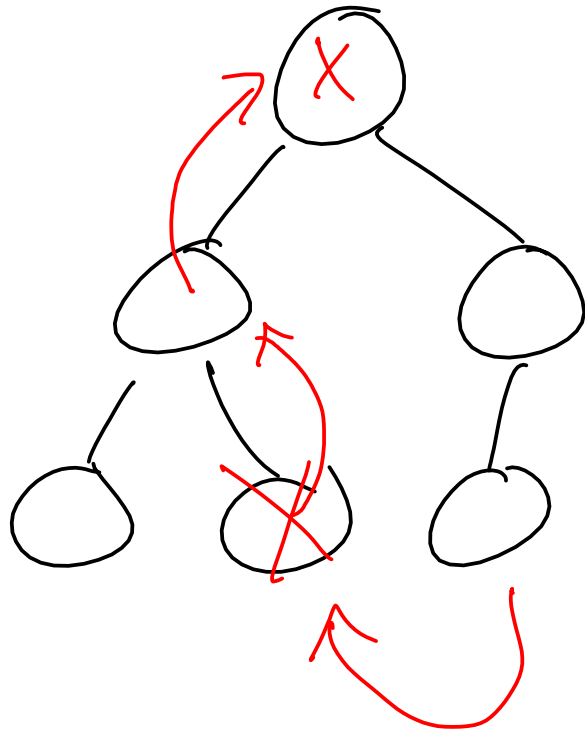
add to heap

Function enqueue(Pprio, V value):

add new node to end of heap

While that node has priority > parent priority
bubble up

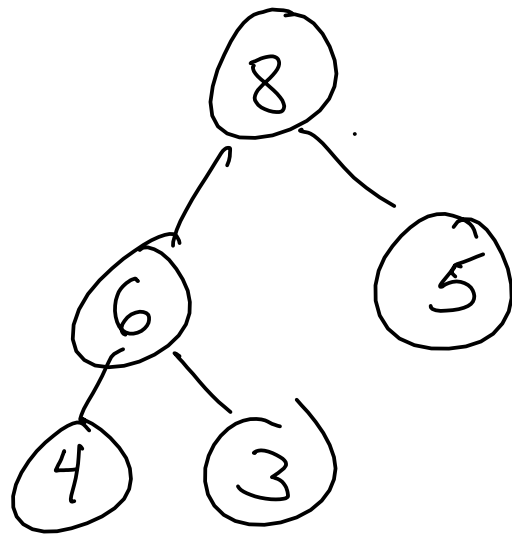




Remove root
Hoist larger elements
Copy last node into open spot
Bubble up that spot

improve

bubble down

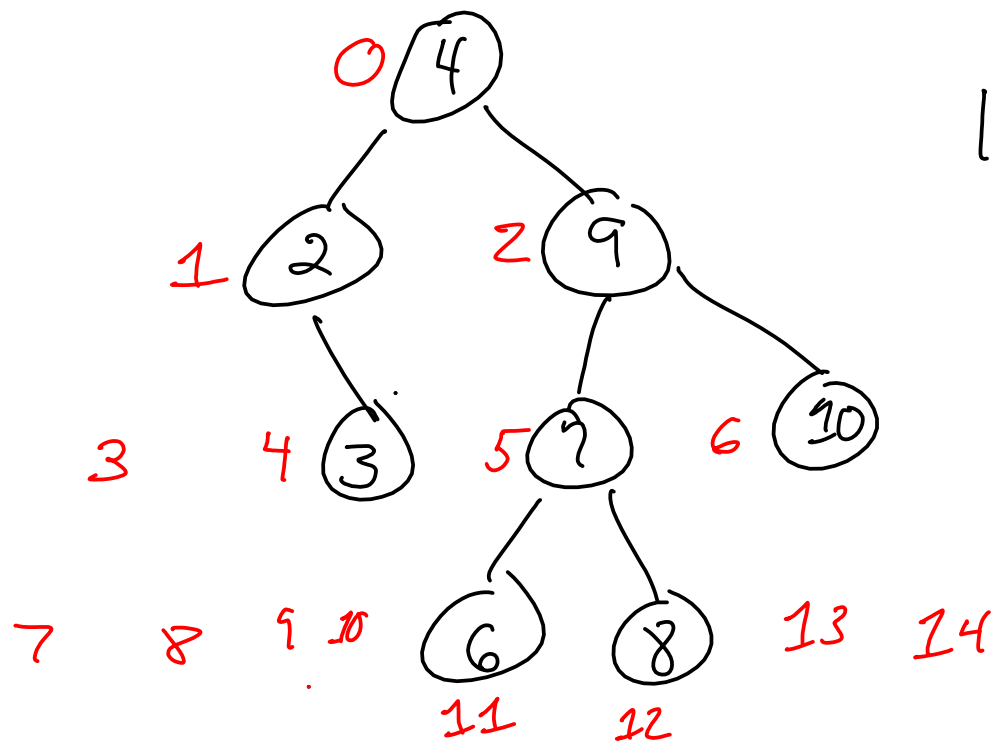


9

Swap last node w/ root

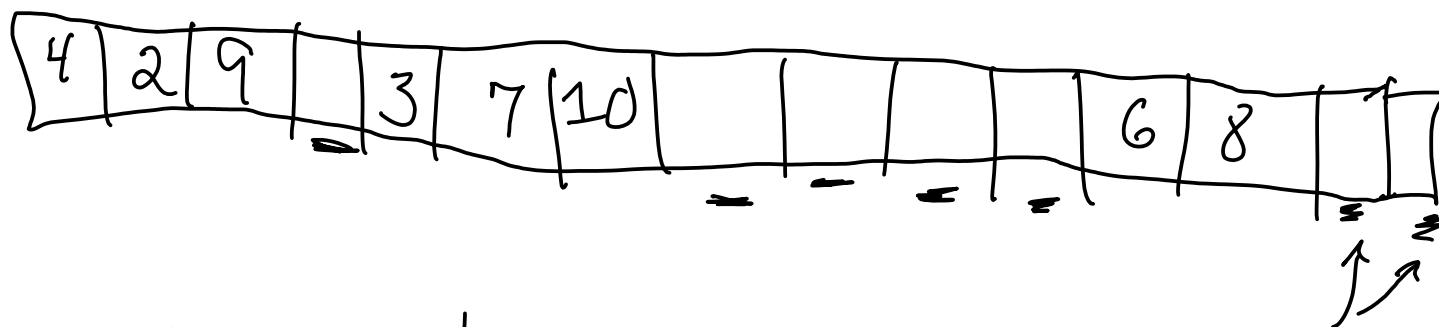
Bubble down root

Array Tree



i
left child of i

$$2i + 1$$



if tree is complete, no need for validity

