

^{Max}
Heap is a binary tree w/ invariant:

- * every child has priority less = than parent
- * complete

PQ impl
 Balanced BST

List (sorted)

List (unsorted)

MH

eng	$O(\log n)$
deq	$O(\log n)$
front	$O(\log n)$
front Prio	$O(\log n)$

$O(n)$

$O(1)$

$O(1)$

$O(1)$

$O(1)$

$O(n)$

$O(n)$

$O(n)$

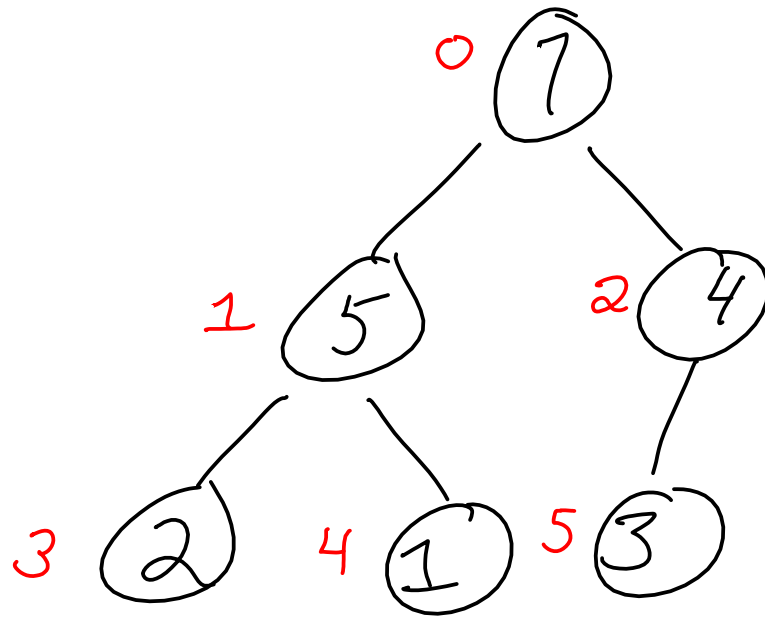
$O(\log n)$

$O(\log n)$

$O(1)$

$O(1)$

```
template <typename P, typename V>  
class MaxHeap:  
    ArrayList < pair<P, V> > nodes;
```



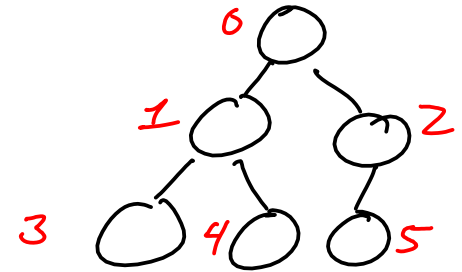
7 5 4 2 1 3

Method enqueue(P priority, V value):
 this → nodes.insertAtTail((priority, value))
 bubbleUp(this → nodes → getSize() - 1)

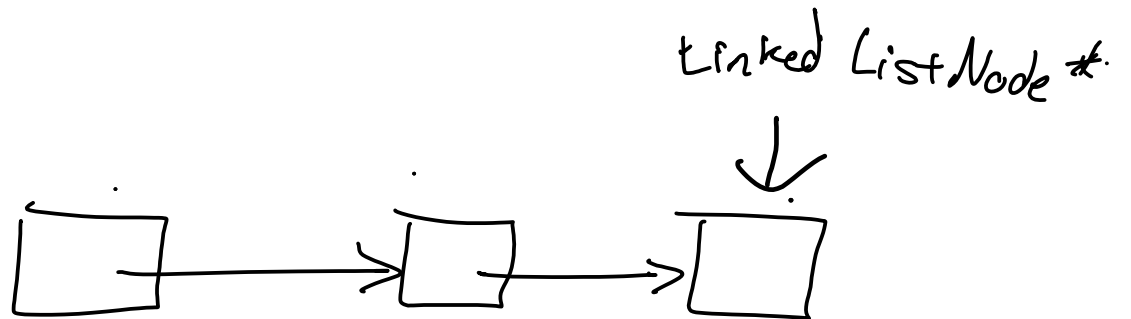
Method bubbleUp(int nodeIdx):

If nodeIdx != 0:

If this → nodes → get(nodeIdx).prio >
 this → nodes → get(parent(nodeIdx)).prio:
 swap(nodeIdx, parent(nodeIdx))
 bubbleUp(parent(nodeIdx))



Function parent(idx):
 $(idx - 1) / 2$



Method dequeue()

swap (this → nodes → getSize() - 1, 0)

x ← this → nodes → removeTail()

bubbleDown(0)

Return x.value

Function left(idx)

$idx * 2 + 1$

Function right(idx)

$idx * 2 + 2$

Method bubbleDown(idx)

If $(idx + 1) * 2 > \text{this} \rightarrow \text{nodes} \rightarrow \text{getSize}()$:

Do Nothing

Else If $\text{right}(idx) \geq \text{this} \rightarrow \text{nodes} \rightarrow \text{getSize}()$

If $\text{nodes}[idx].prio < \text{nodes}[\text{left}(idx)].prio$:

swap(idx, left(idx))

bubbleDown(left(idx))

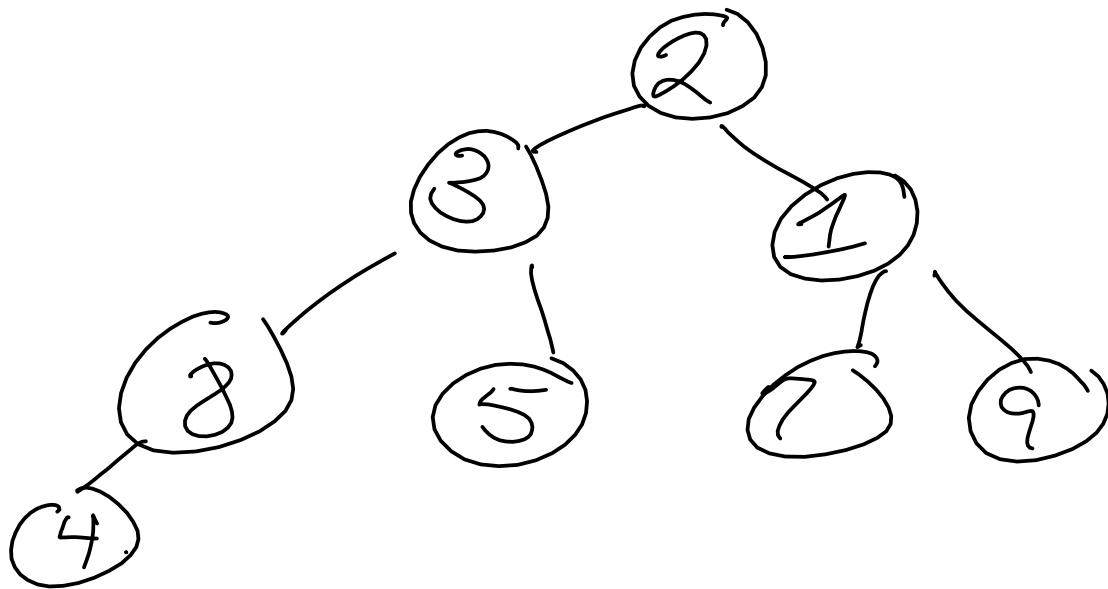
Creating Heap:

make empty heap
for each element in list:
add it to the heap!

$O(n \log n)$

← adding n times
← enqueue ($O(\log n)$)

2, 3, 1, 8, 5, 7, 9, 4

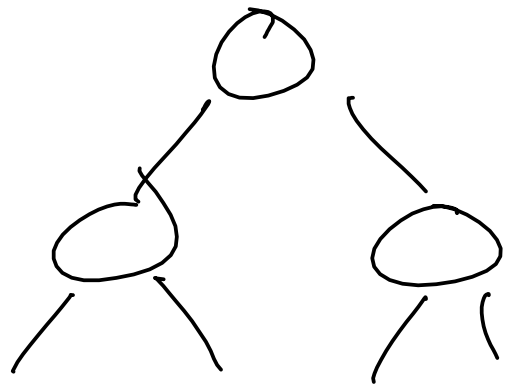


Method heapify () : $O(n)$

{ For $i \leftarrow \text{nodes.getSize}() - 1$ Down To 0 :

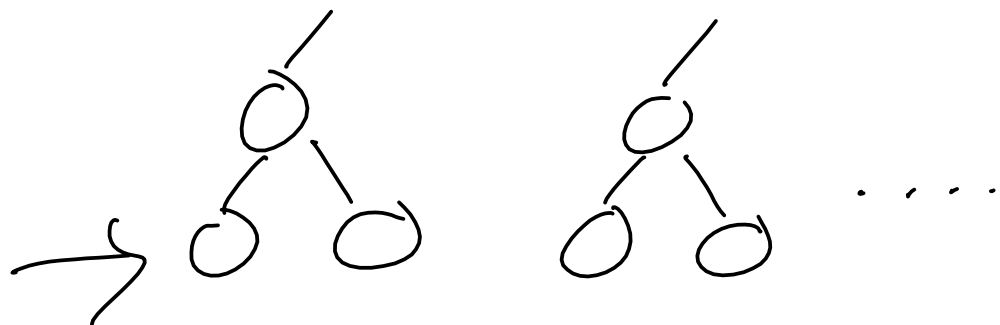
bubble Down (i) $\leftarrow O(\log n)$

$O(n \log n)$



$$\frac{n}{2} \cdot 0 + \frac{n}{4} \cdot 1 + \frac{n}{8} \cdot 2 + \frac{n}{16} \cdot 3 + \dots + \frac{n}{n} \cdot \log(n)$$

.....



$$n : \left(\frac{1}{4} + \frac{2}{8} + \dots + \frac{\log n}{n} \right) \dots < 2$$

Top k elements

heapify $(O(n))$

take k elements $k \cdot O(\log n)$