Heap Sort Notes 2/21/19.

Q. What is a heap?

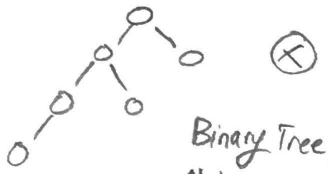
A. A Binary Tree w/ 2 properties :

1. All nodes in specific order : root must be >= all children (max-heap)
or <= all children (min-heap)
2. Shape must be complete.

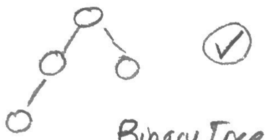
Q What does a complete tree look like?

A. Every Single level of the tree must be filled.

* Exception last level does not have to be complete but filled left to right.

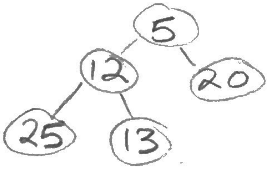


Binary Tree,
Not a heap

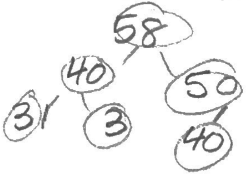


Binary Tree
+
Heap

Min heap example



Max Heap Example



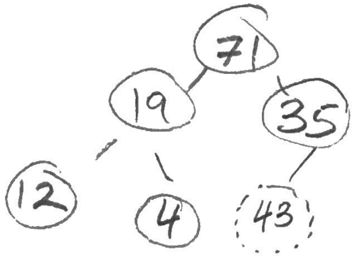
* Heaps can have duplicate values
* Does not necessarily follow rules of BST

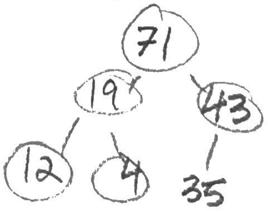
Q. What is the heap order property?

A. Ordering of the parent nodes compared to children. (min heap vs max heap).

Q. Where do we add a node to a heap?

A. left & Bottom most available spot in tree. How do we update the heap?
Maintain Heap Order Property by swapping.



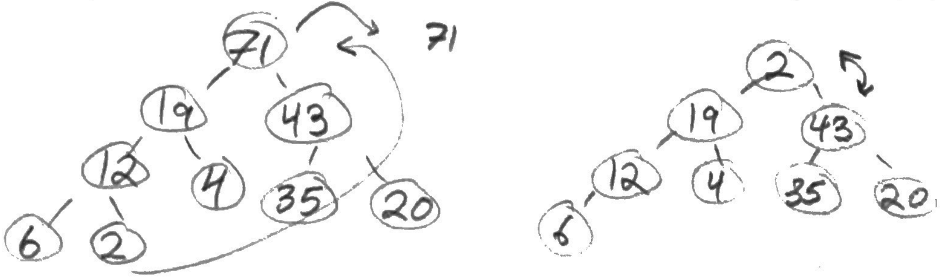


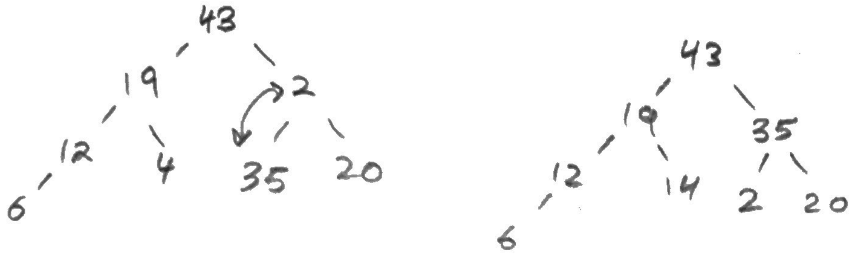
Q How do we remove from the heap?

A. 1. Remove the root node

2. Place bottom right most node in root.

3. Swap children until the heap maintains Heap Order Property



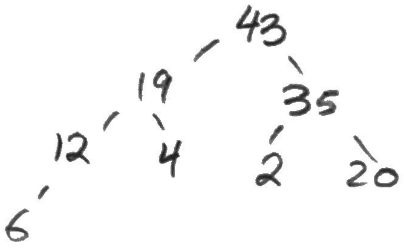


"Bubble Down" to maintain Heap Order Property

Q. What is the most important characteristic of a heap?

A. Maximum / Minimum always the root node.

☆ Represented as an array, it is the zero-th element.

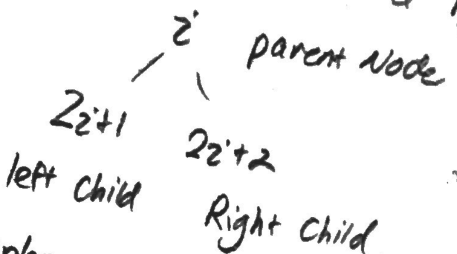


[43, 19, 35, 12, 4, 2, 20, 6]

Q. Given index of node in heap, find its left & right child.

A. Left: $2i+1$

Right: $2i+2$



Q. Why are heaps often implemented as arrays?

A. Very efficient way of representing a priority Queue.

Q. What is Run time of removing min/max element from the heap?

A. O(1)

min & max are inherently at the zero-th index.

Q. What is the Run time of Insertion & Deletion?

o(log n) Because of inherent tree structure.