

WOP 3502 11/3/25

1) PS is posted. (WC for due date)

2) ~~Hash Tables~~ Heaps!

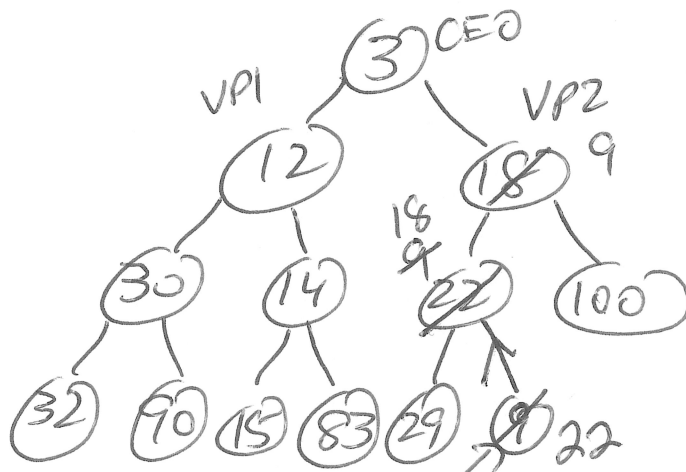
→ ~~Set~~ Abstract Data ~~Struct~~ Type

Operations

$O(\lg n)$ time $\left\{ \begin{array}{l} \text{Insert} \\ \text{Delete } \underline{\text{Minimum}} \text{ (some implementations max)} \end{array} \right.$

Always do purposes of our
this for class or Foundation Exam!

Heap - complete binary tree that satisfies the
heap node property = all values in a subtree of
a node store values greater than it.



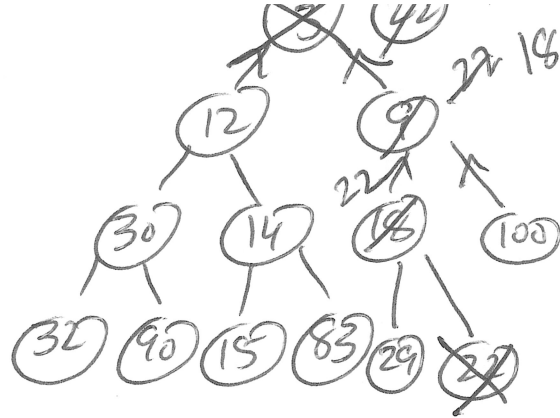
TASKS

Insert ✓
Delete Min.

Insert: "new hire" — initially placed here!

Issue - heap node property might be messed up

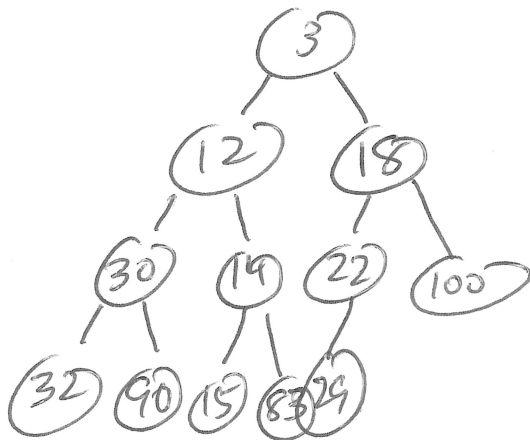
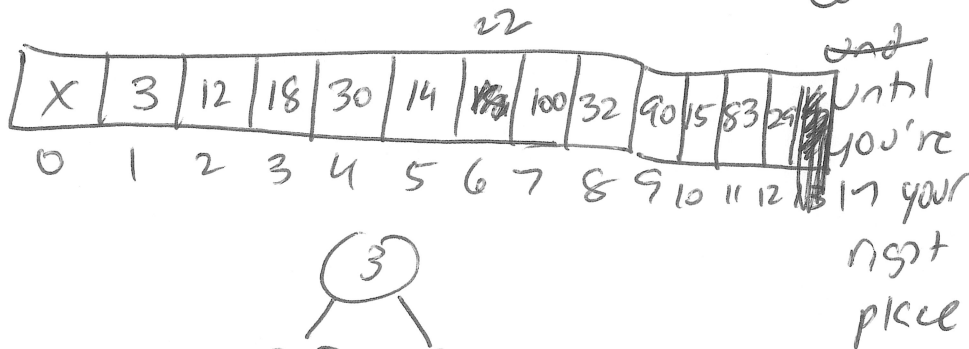
Fix - Percolate Up (while your value < parent value)
Swap w/ parent, stop @ root.



Delete min

- 1) vacate value at root.
- 2) move last item into root
- 3) Percolate "mailbox" down

Implement in an array



$$\text{par}(i) = \text{node } i/2$$

$$\text{leftchild}(i) = \text{node } 2*i$$

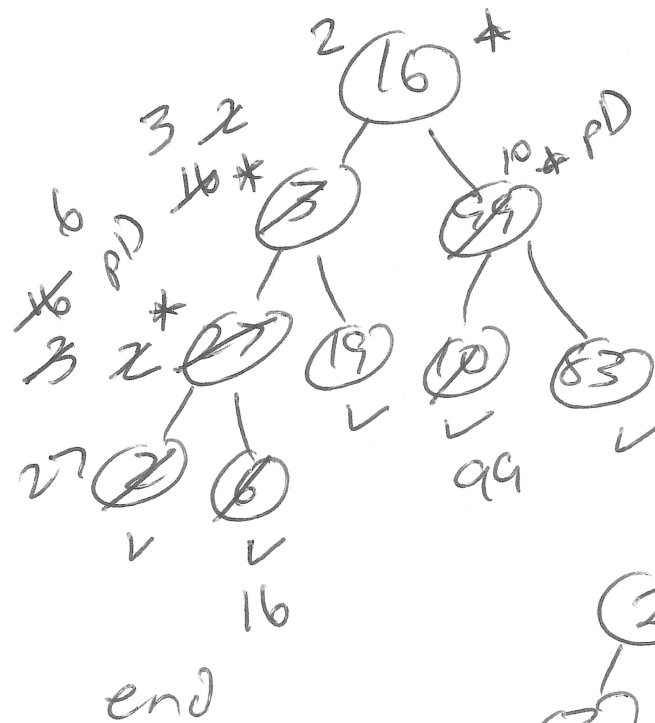
$$\text{rightchild}(i) = \text{node } 2*i+1$$

- find min value of 2 children (if there are)
- if this is lower then you swap

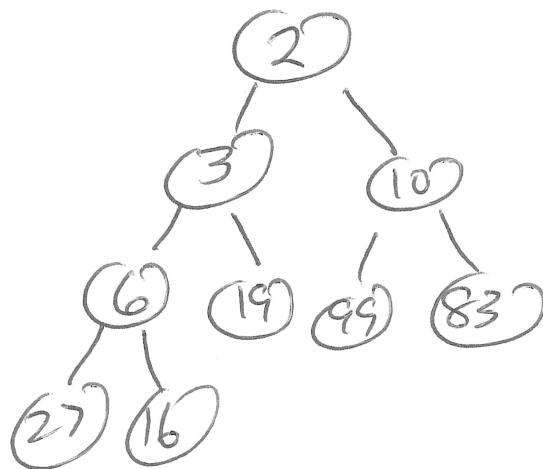
What if you get all n values at once and want to make a heap

- 1) Call Insert n times - run time $O(n \lg n)$
- 2) Immediately place in array and call heapify - $O(n)$

16, 3, 99, 27, 19, 10, 83, 2, 6



Start at highest
index non-leaf
node go backwards
at each node
call percolateDown



* Next time will do proof of $O(n)$
run-time.