## Heap

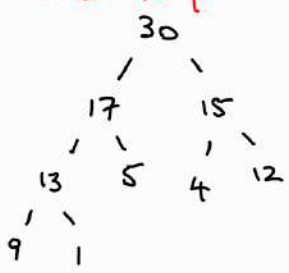
heap property

- Like trees but with additional properties
- Specialized tree-based data structure that satisfy heap property
- Highest / lowest priority element stored at root

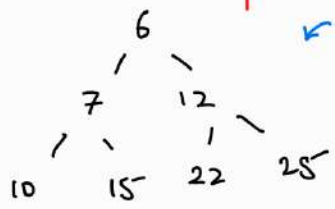
minimum heap:
- parent node always lesser than child nodes
- Root node will always be min. value

Maximum heap:
- Parent node will always be greater than child nodes
- Root node will always be max value

Max. heap




Min. heap




Also a binary heap.
(node have 2 child)
+ complete tree
↳ every level filled except last. & all nodes left justified.

## Heap Analysis

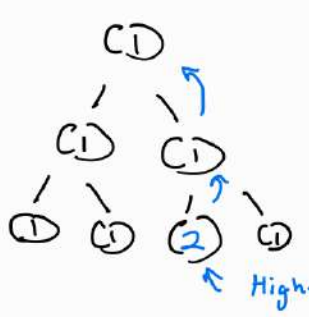
Insert: O(log(n)) - Insert values then compare. (min) (max) if smaller / greater, swap upwards

Check min/max: O(1) - Always at root node.

remove min / max: O(log(n)) - need to shift node upwards

## Heap uses

1. priority Queues
   - Elements assigned weights
     ↳ Higher weight node will swap upwards to be processed.




1 2 3 4
least impt — most impt

Higher priority node will swap upwards

2. Heap sort (not commonly used)
3. Dijkstra's Algorithm. - least move to get between nodes