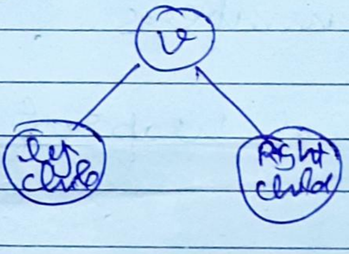
# Heaps1

① Complete Binary Tree

1. Each internal node has atmost 2 children
2. All the levels except the last one are full.
3. ~~Nodes~~ Leaves appear from left to right at last levels.

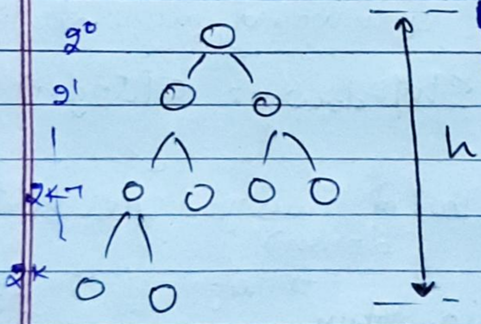
② Heap.

Heap is a complete Binary Tree in which each internal nodes v satisfies.



v < left child and Right child

③



$2^0+2^1+2^3 - \cdot 2^k = n$

Height of heap is O(log n)

n = no. of nodes

Date: / /

Page No.

(4) If an array represents a heap,
for zero based indexing

children of $i = 2i+1, 2i+2$

parent of $i = \begin{cases} \lfloor i/2 \rfloor & i = \text{odd} \\ \lfloor i-1/2 \rfloor & i = \text{even} \end{cases}$

(5) Given a min heap, how can you use it to sort numbers.

(6) Running time of heap sort = O(nlogn)
↑ each time heap sort is implement and shift down is called, the height is decreased by 1, at max you would have to call heap sort on all n.

Running time of shift down = O(log n)
↙
c × (no. of times you call shift down)
↳ height of the tree

(7) Build heap algorithm.
↳ Correctness
↳ Running tm

(8) Given a heap, add or delete.