NAME : Jagadeesh R

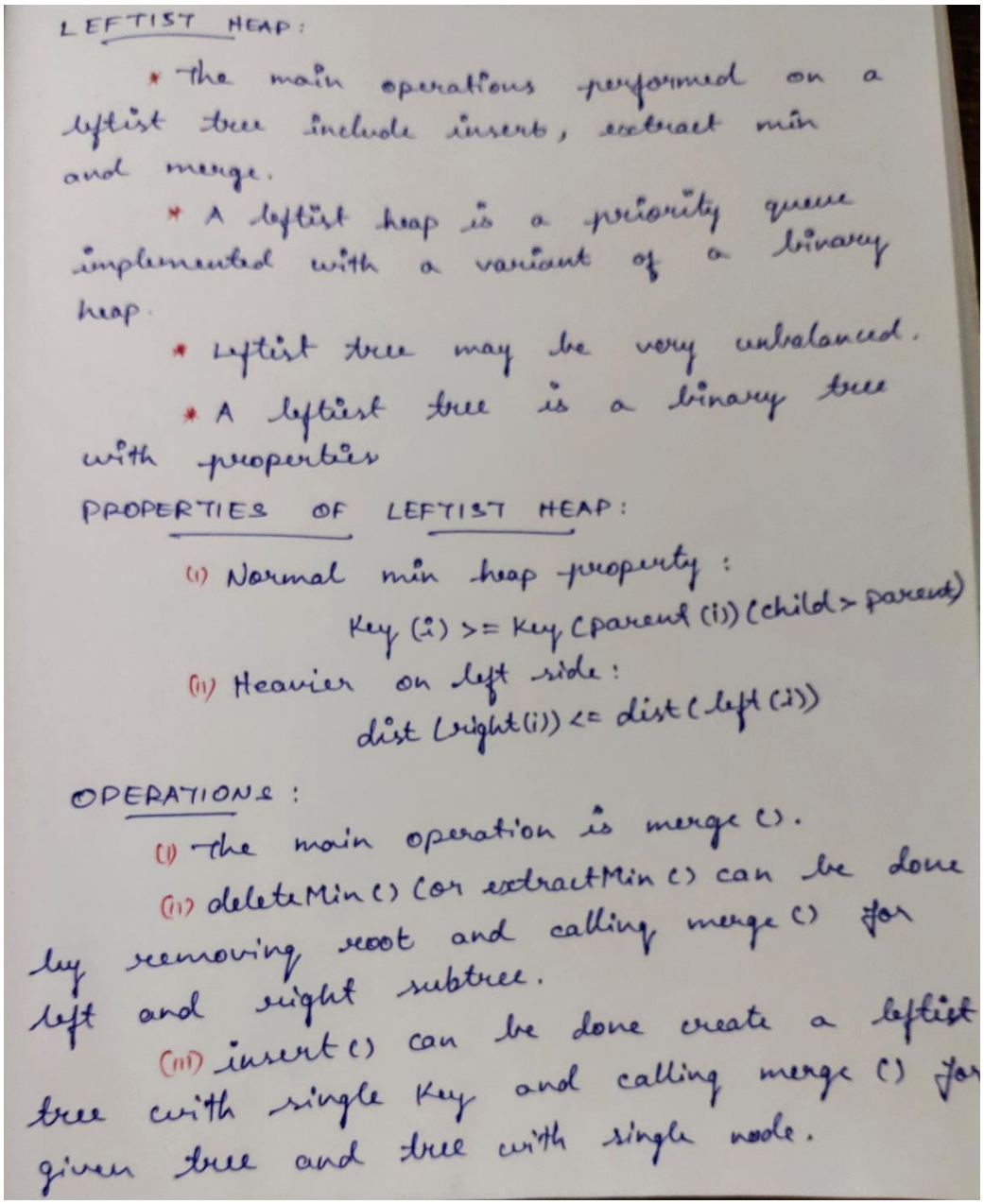
REGNO : 2021506314

OOPS Lab 9 : Study on

(i) Leftist heap

(ii) Binomial heap

LEFTIST HEAP :



LEFTIST HEAP:

* the main operations performed on a leftist tree include insert, extract min and merge.
* A leftist heap is a priority queue implemented with a variant of a binary heap.
* Leftist tree may be very unbalanced.
* A leftist tree is a binary tree with properties

PROPERTIES OF LEFTIST HEAP:

(i) Normal min heap property:

Key (i) >= Key (parent (i)) (child > parent)

(ii) Heavier on left side:

dist (right(i)) <= dist (left (i))

OPERATIONS:

(i) the main operation is merge ().

(ii) deleteMin () (or extractMin () can be done by removing root and calling merge () for left and right subtree.

(iii) insert () can be done create a leftist tree with single key and calling merge () for given tree and tree with single node.

INSERTION:

(i) newHeap = createLeftistHeap (val)

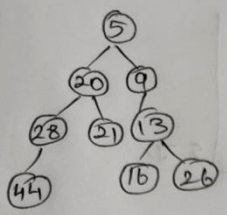
(ii) merge (leftistHeap, newHeap)

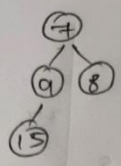
DELETION:

(i) val = root, val

(ii) root = merge (root, right, root, left)

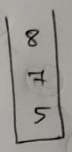
(iii) return val

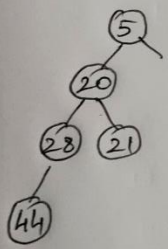




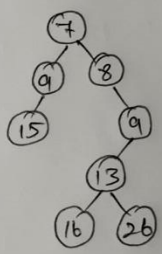
MERGE:

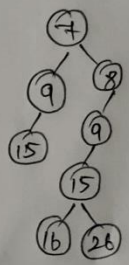
compare (5,7)
Push 5
compare (9,7)
Push 7
compare (9,8)
Push 9
compare (9,null)
NULL

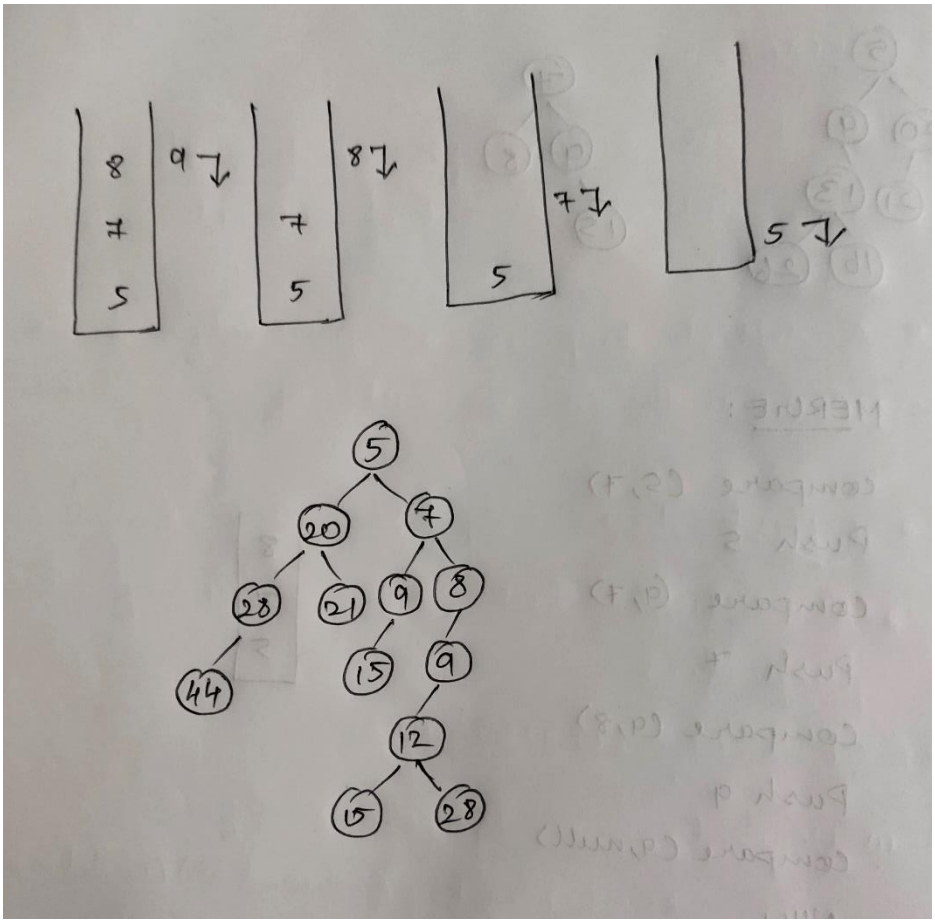




Refers to node 9






8
7
5
9
7
5
8
5
7
5

BINOMIAL HEAP :

BINOMIAL HEAP:

* It is similar to binary heap
* It supports quick merging of two heaps
* Implemented as set of binomial tree
* Binomial trees of order 0 is single node.

PROPERTIES OF BINOMIAL HEAP:

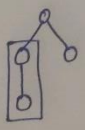
* Each binomial tree is a heap obey minimum heap properties.
* (i.e) Heap of the node is greater than or equal to the heap of its parent
* There can be either one or zero (1 or 0) binomial tree for each order including zero order.

ORDER OF BINOMIAL HEAP:

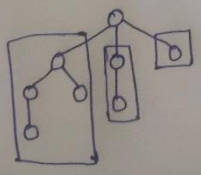
⇒ Order 0 : 0

⇒ Order 1 :

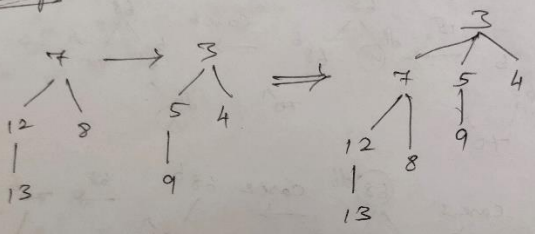
⇒ Order 2 :



⇒ Order 3 :



Merging :

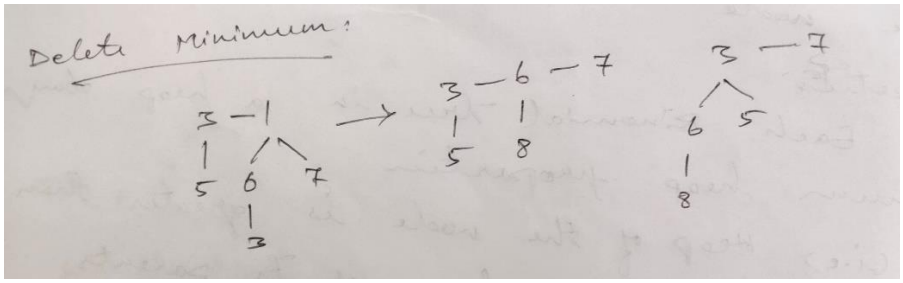


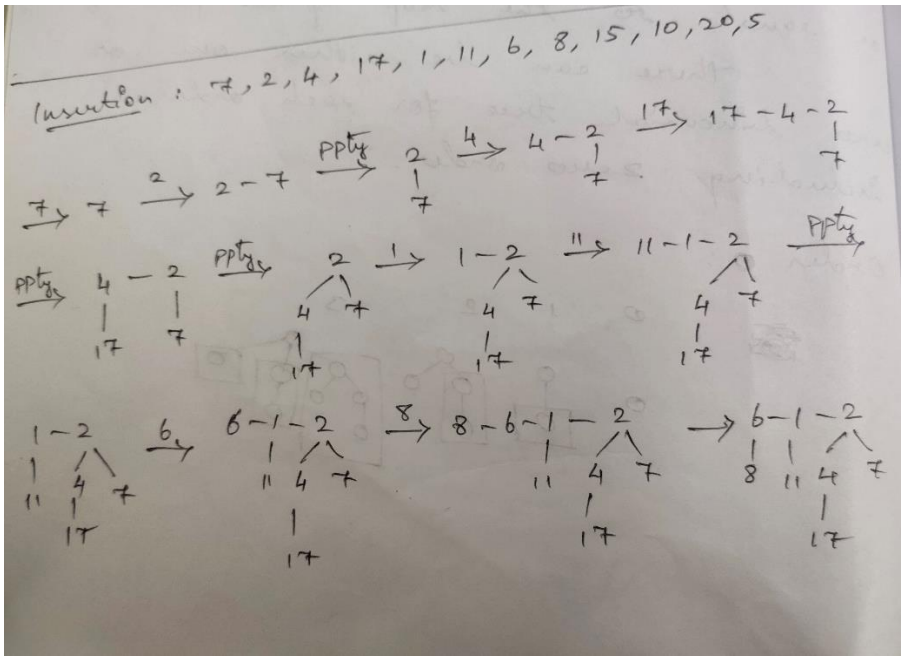
INSERTION:

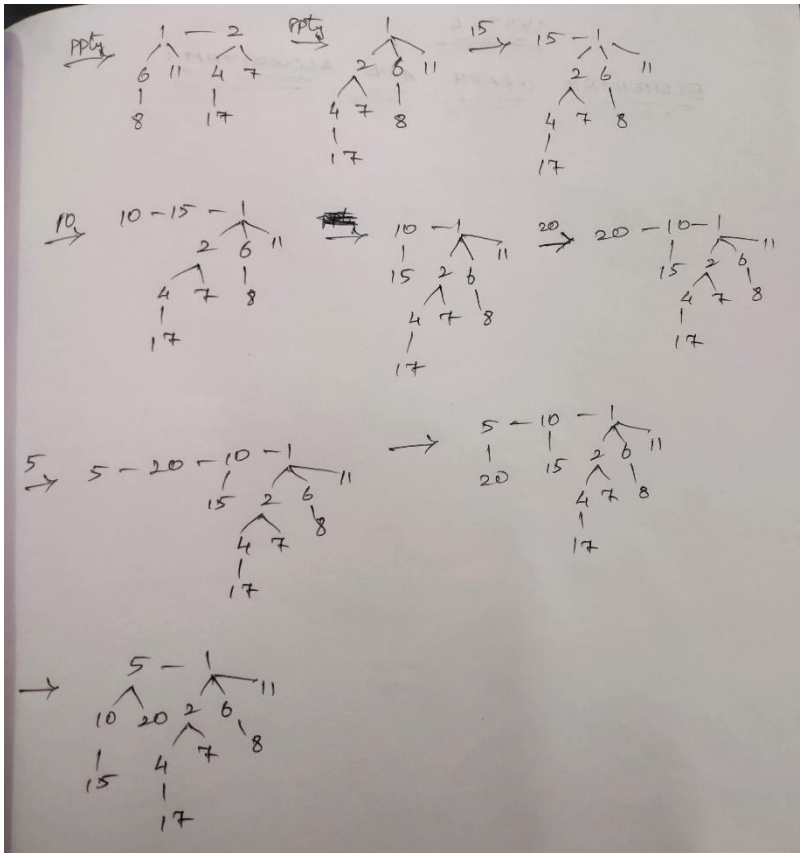
* Create new heap elements and merge it with original heap.

⇒ FIND MINIMUM:

* No need to travel entier tree just check its root node.






ppty
ppty
15
10
20
5