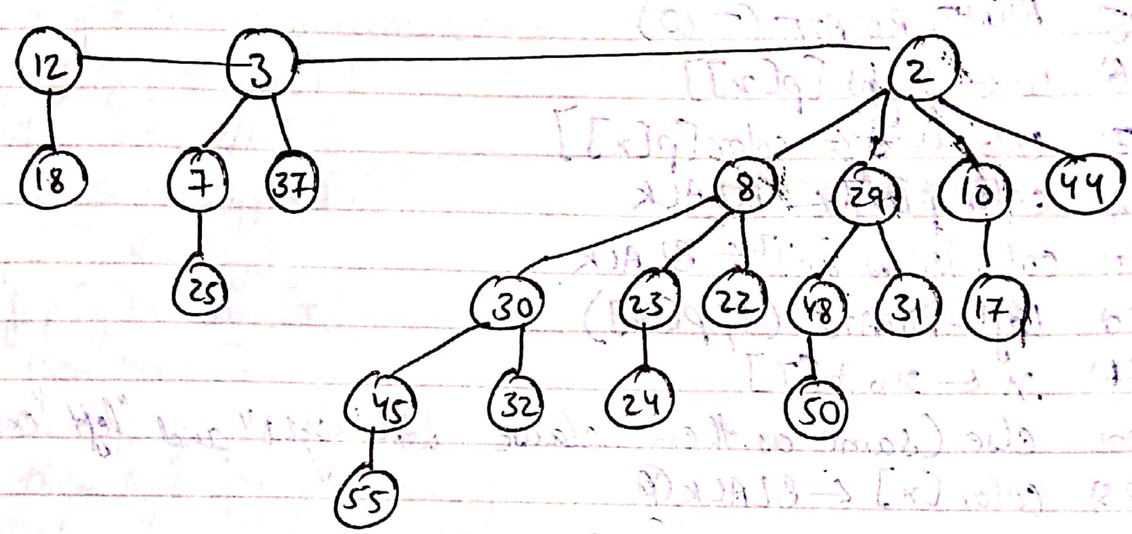
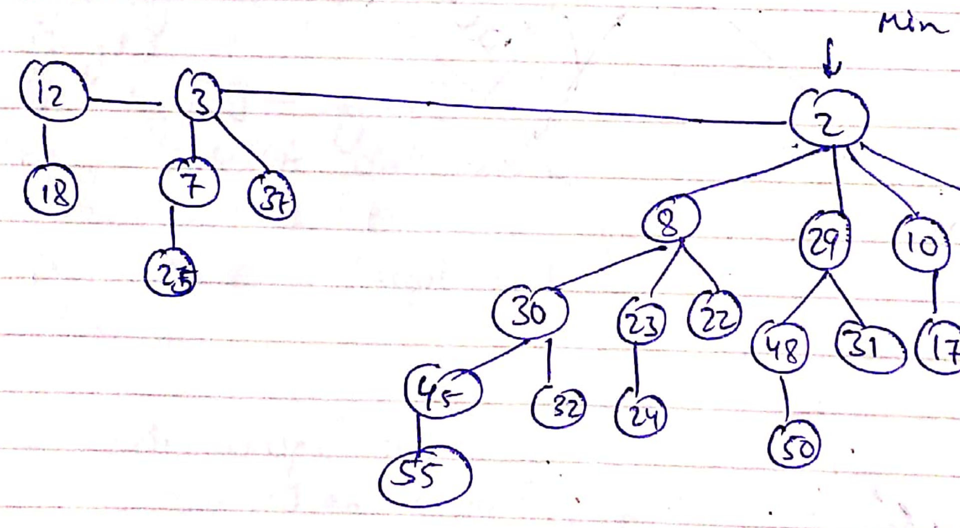
Assignment

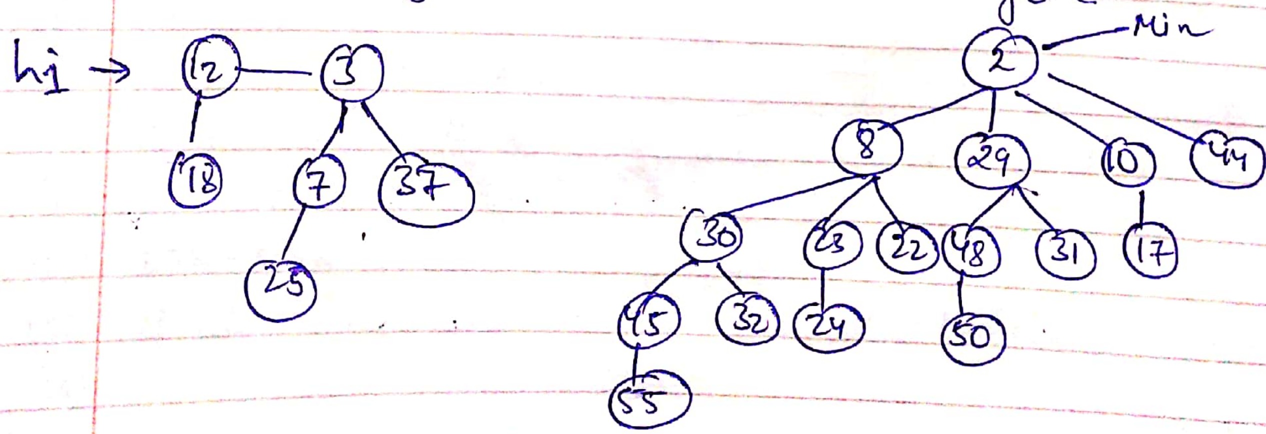
① Run the binomial heap extract min procedure for the following binomial heap and give the resultant binomial heap.



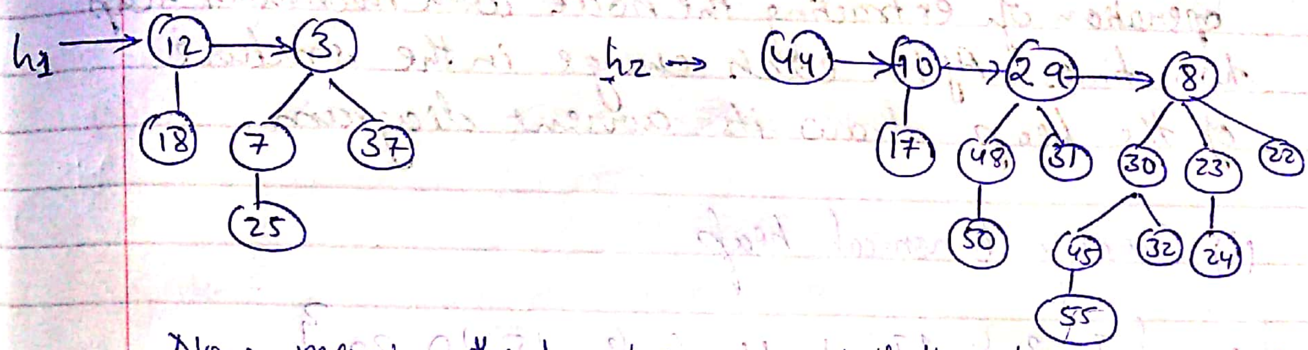
Sol



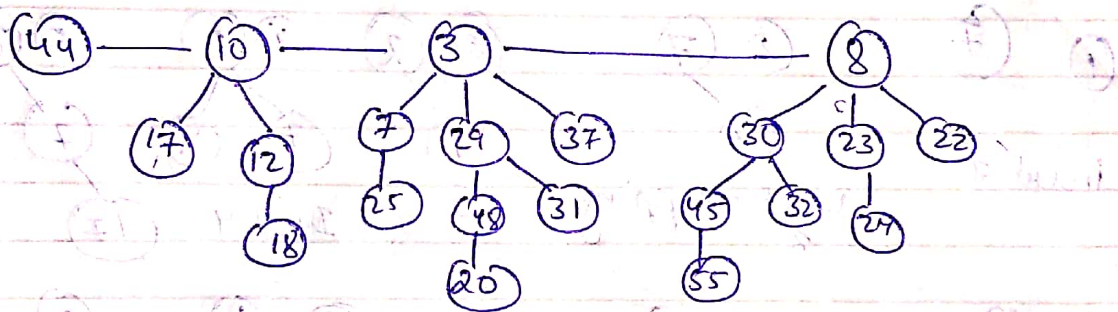
Now removing the connection ~~btw~~. Therefore



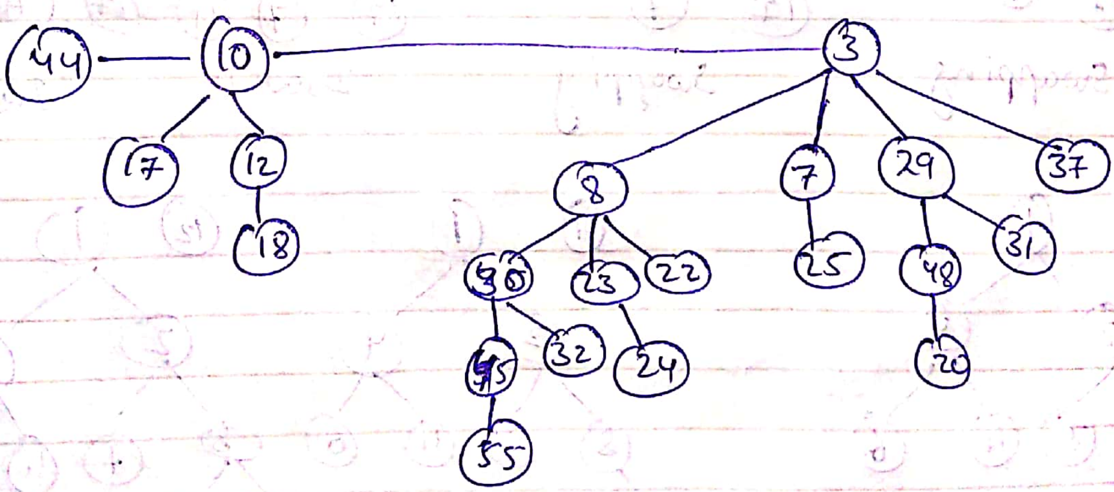
Removing the min.



Now merging the two heaps. and if they have same order then they will get merge.



Now merging 3 & 8 heap as they have same order



(2) Let a = {7, 2, 4, 17, 1, 11, 6, 8, 15, 10, 20}.

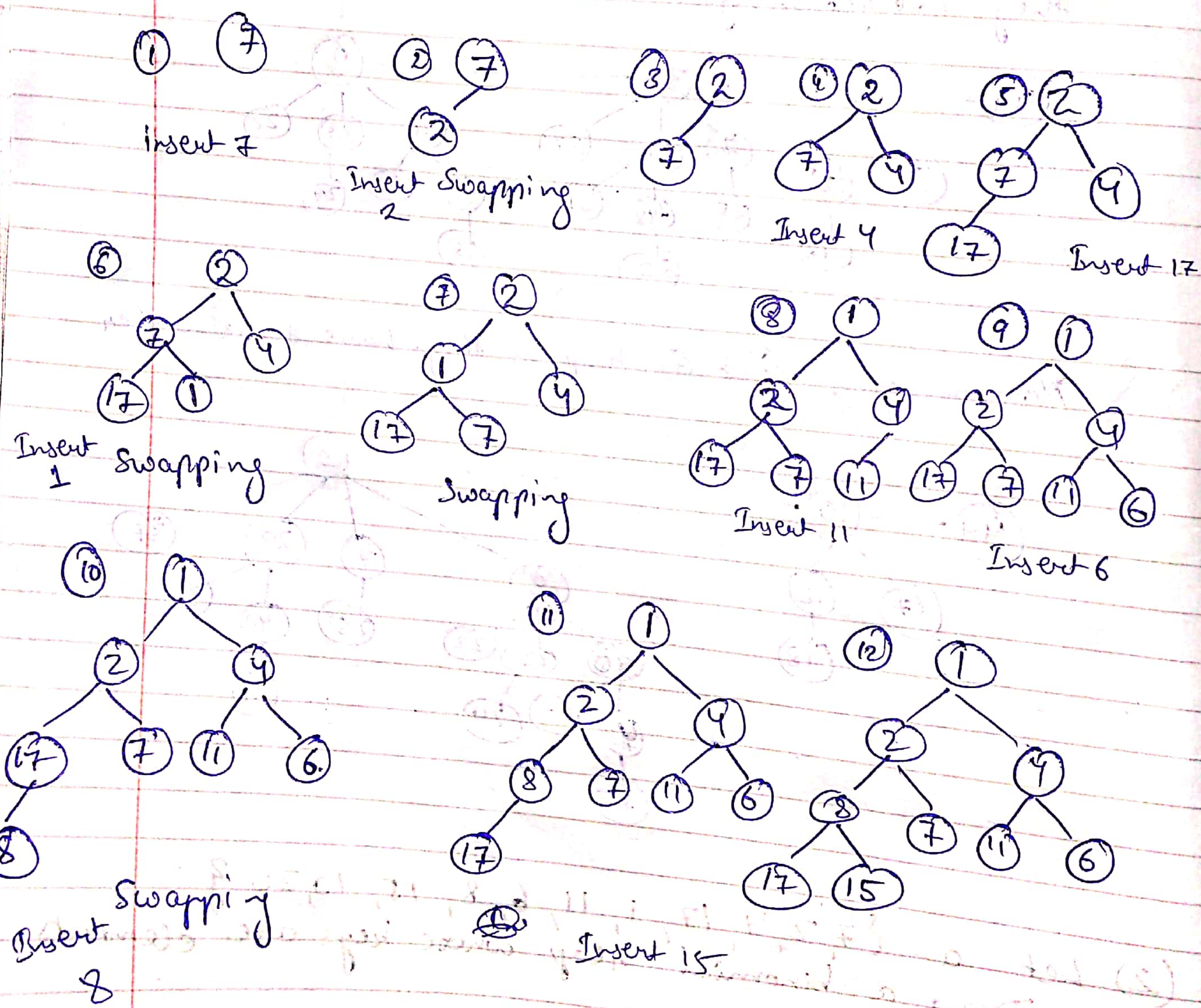
(i) Draw a binomial heap whose keys are elements of "a".

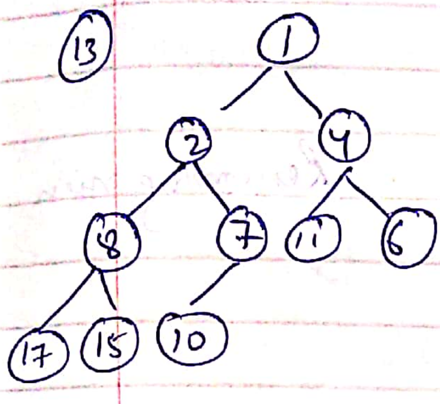
(ii) Insert a new element with the key 5 into the heap.

(vii) To a binomial heap obtain this way apply the operation of extracting the node with minimum heap two times. After each change in the structure of the heap draw its current diagram.

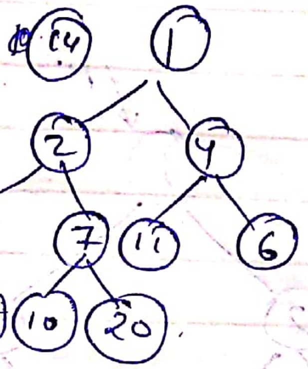
Sol (i) Creating binomial heap

$a = \{7, 2, 4, 17, 1, 11, 6, 8, 15, 10, 20\}$



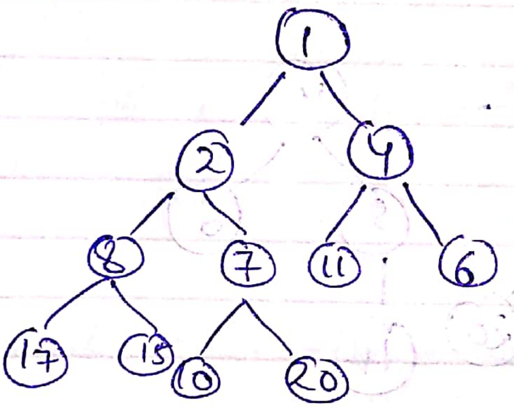


Inser 10

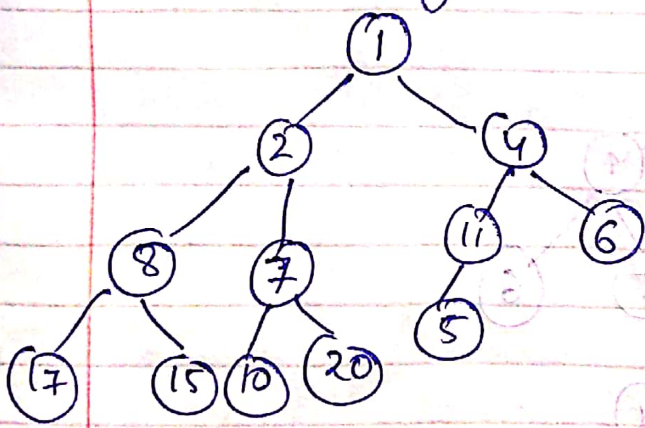


Insert 20.

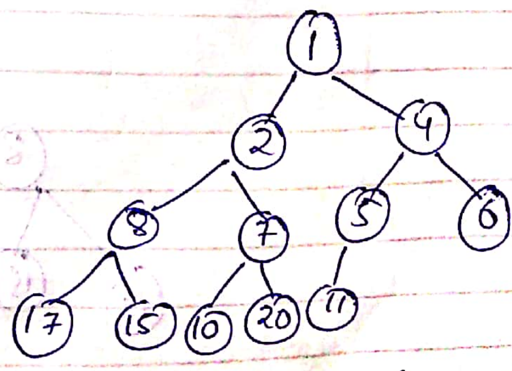
Final Binomial Heap



(ii) ~~10~~ Inserting 5.



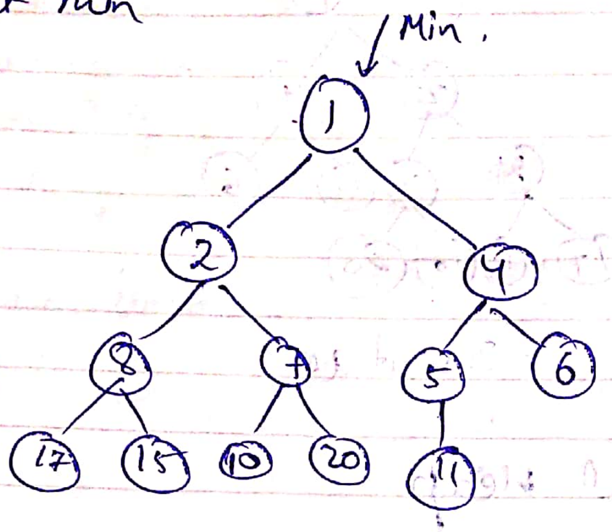
Swapping 5 with 11



Final binomial heap.

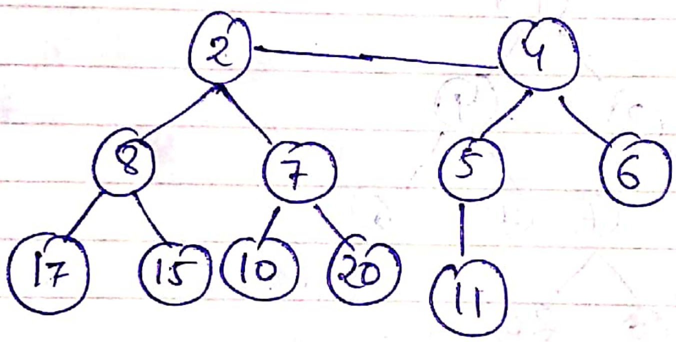
(iii) Extract min

1

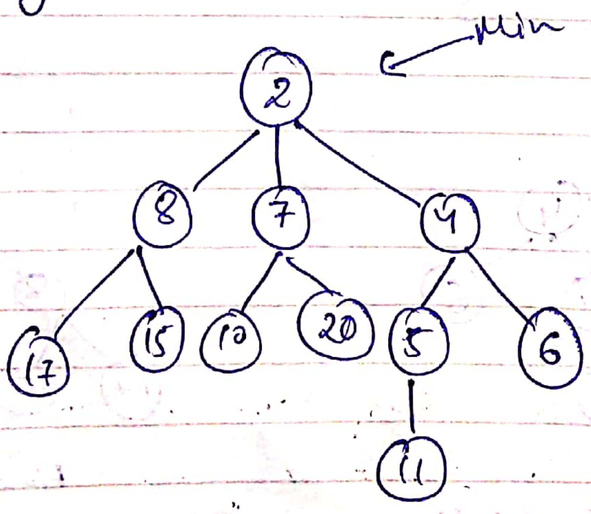


Removing min

2

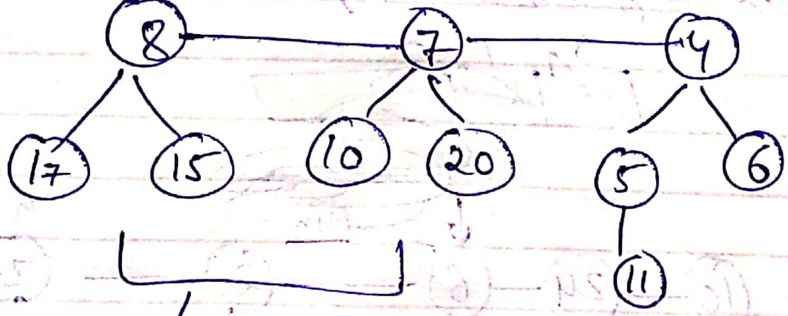


Now they have same orders so merging

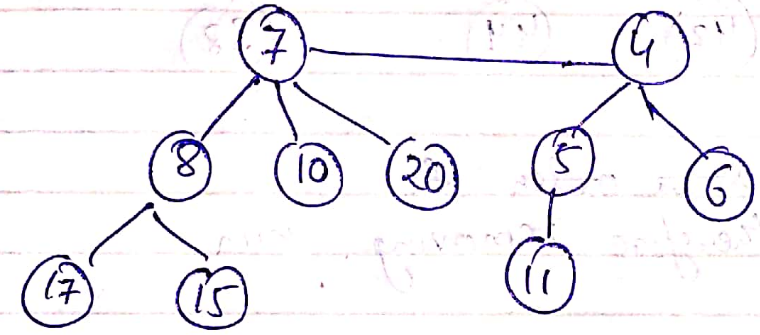


Now extracting min once more

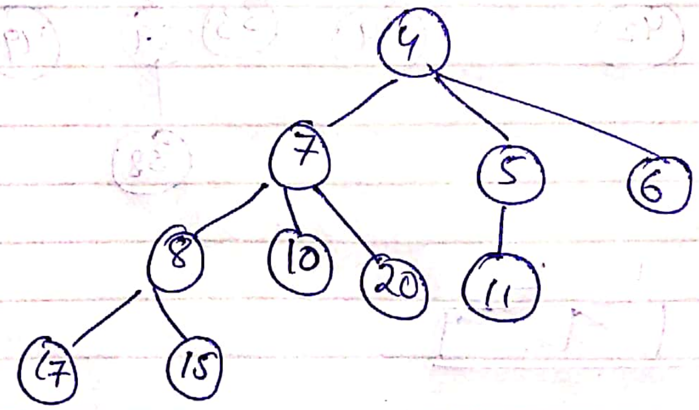
Therefore



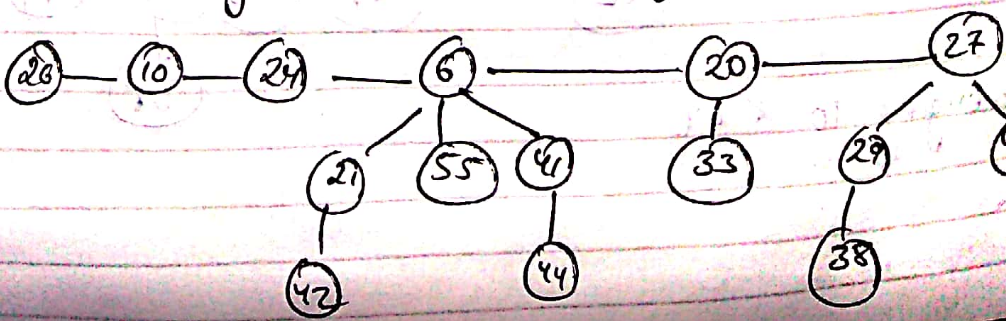
They have same order



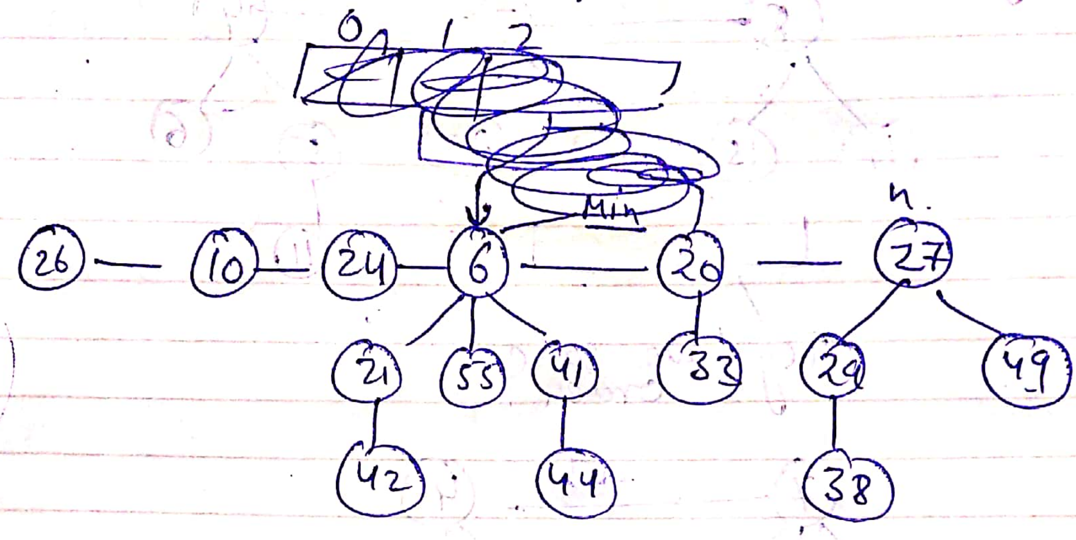
Now they also have same order.



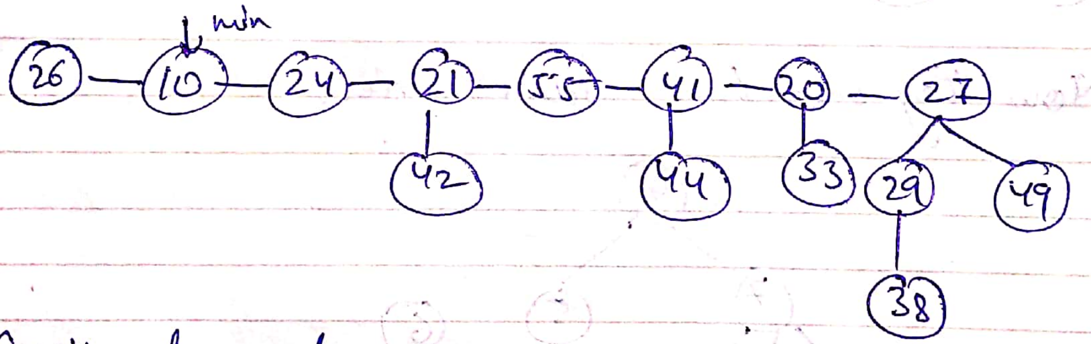
(3) Consider a fibonnaci heap and run the fib-heap-extract-min and give the resultant fibonnaci heap.



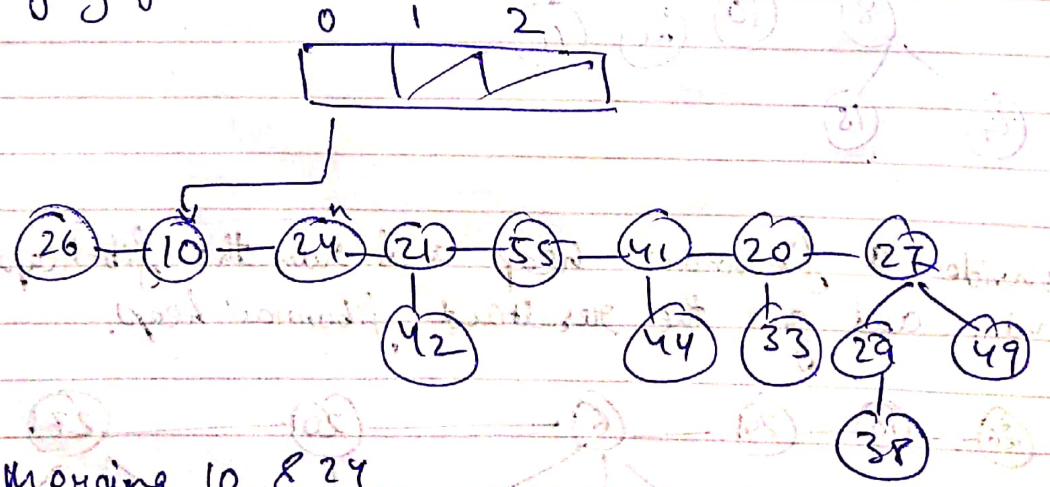
Sol Extract Min in fibonnaci heap.



~~Min & n have same order~~ so
Extract min therefore removing min

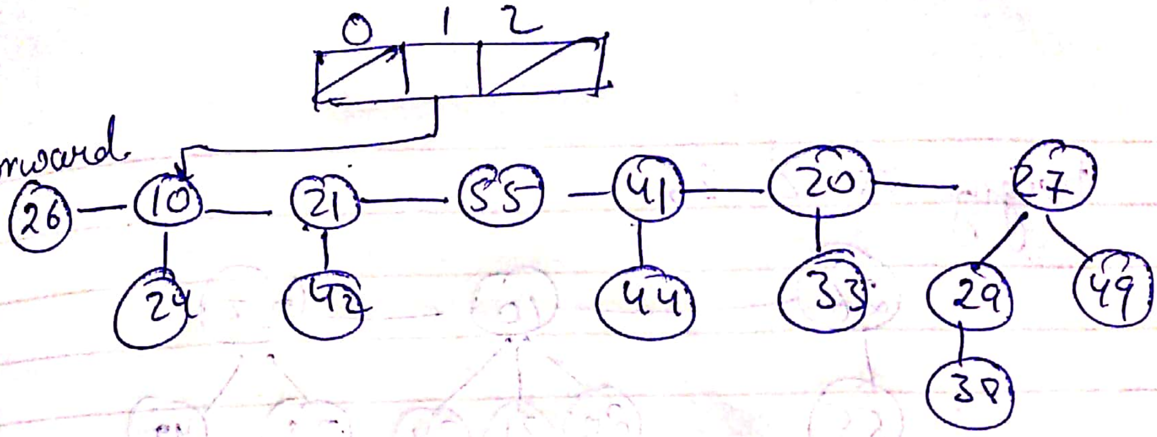


Shifting forward

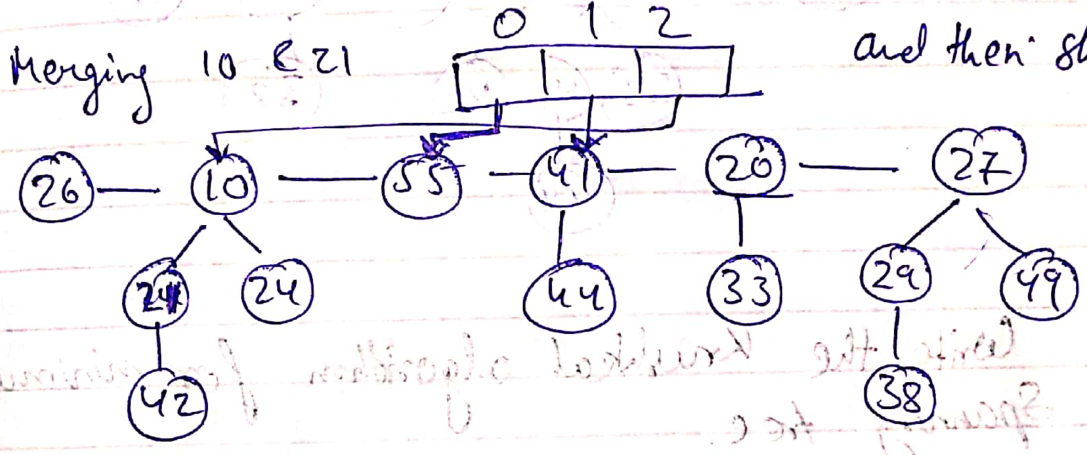


Merging 10 & 24.

Shifting forward

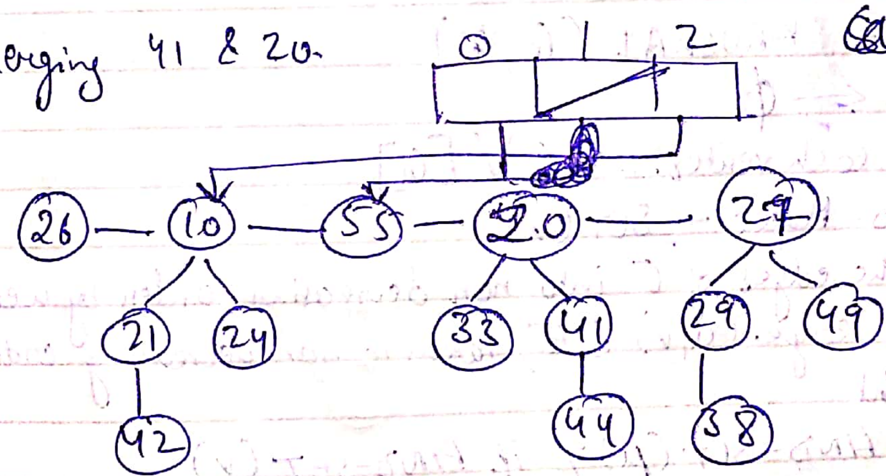


Merging 10 & 21 and then shifting forward

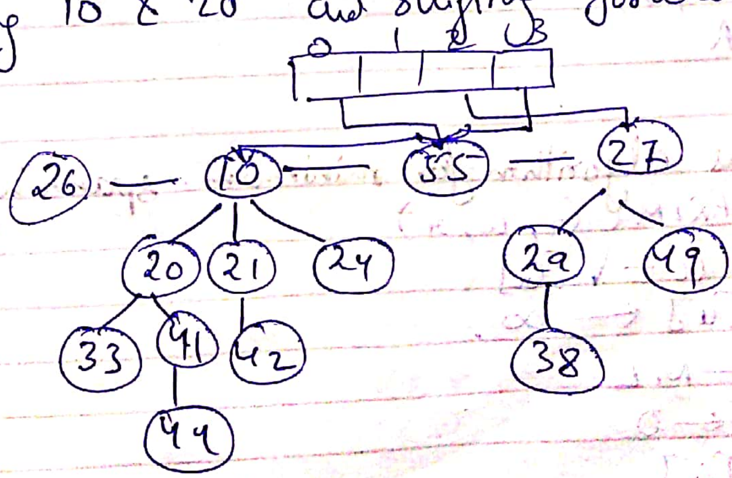


Merging 41 & 20.

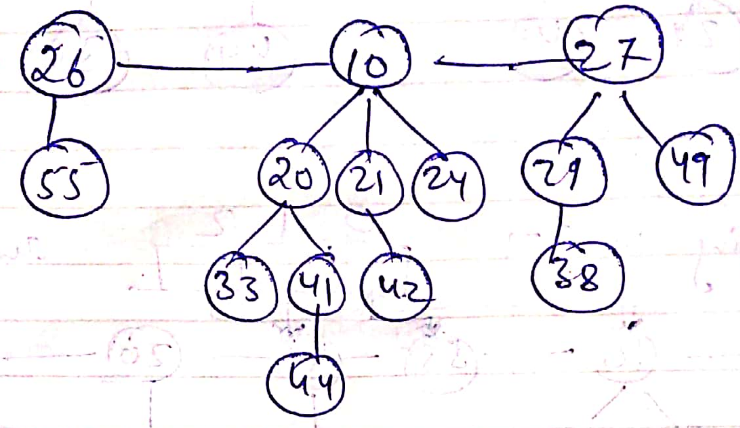
Now both 10 & 20 have same order



Merging 10 & 20 and shifting forward



Merging 55 & 26



(4) Write the Kruskal algorithm for minimum Spanning tree.

Ans MST – KRUSKAL (G, w)

(1) $A \leftarrow \phi$

(2) for each vertex $v \in V[G]$.

(3) do Make – SET (v).

(4) Sort the edges of E into non decreasing order by weight w

(5) for each edge $(u, v) \in E$, taken in non decreasing order by weight

(6) do if FIND–SET $(u) \neq$ if FIND-SET (v)

(7) then $A \leftarrow A \cup \{(u, v)\}$

(8) UNION (u, v).

(9) return A

(5) Write prims algorithm for minimum spanning tree.

Ans MST – PRIM (G, w, r)

(1) for each $u \in V[G]$

(2) do key $[u] \leftarrow \infty$

(3) $\pi[u] \leftarrow$ NIL

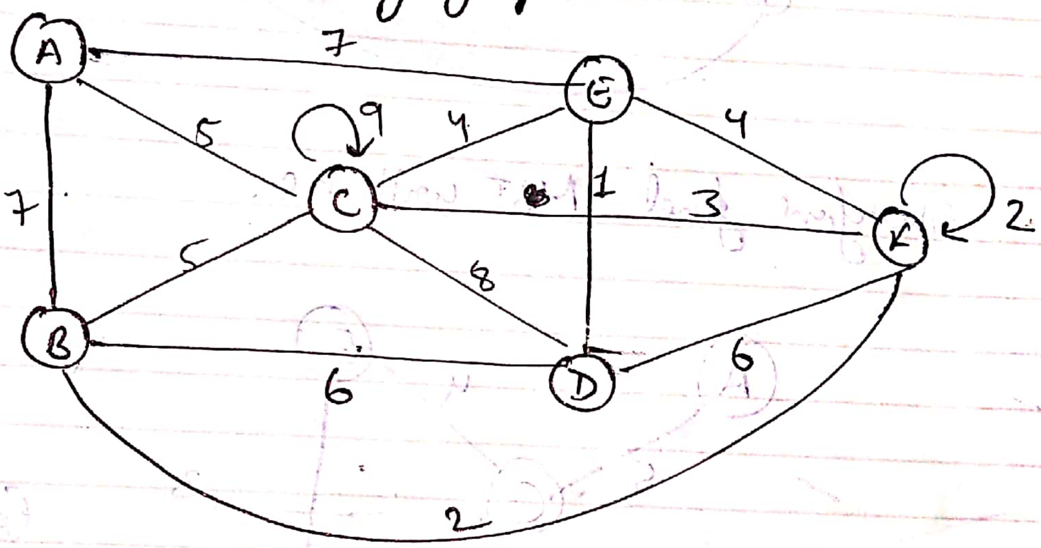
(4) key $[r] \leftarrow 0$

(5) $Q \leftarrow V[G]$

(6) while Q ? ϕ

(7) do $u \leftarrow$ EXTRACT-MIN(Q)
(8) for each $v \in Adj[u]$
(9) do if $v \in Q$ and $w(u,v) < key[v]$
(10) then $\pi[v] \leftarrow u$
(11) $key[v] \leftarrow w(u,v)$.

(6) Find MST for the following graph.

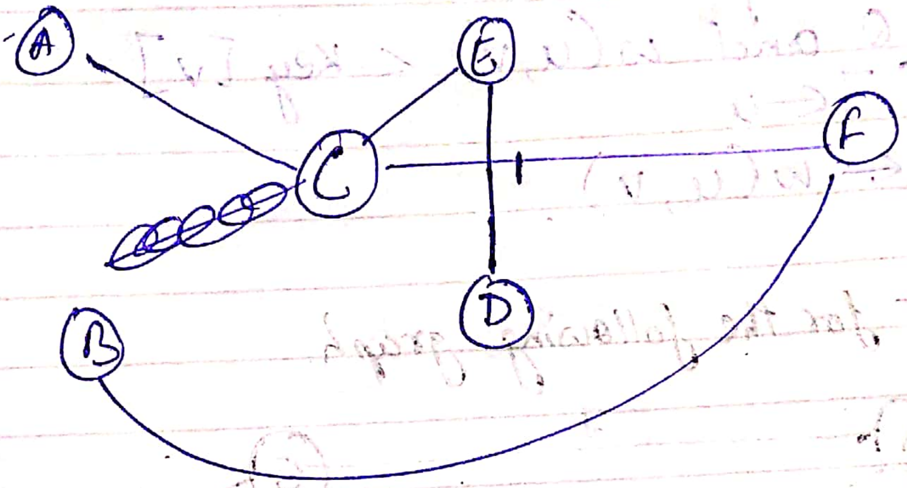


Sol

Weight	Src	Dest
1.	E	D
2.	F	F
2.	B	F
3.	C	F
4.	C	E
4.	E	F
5.	A	C
5.	B	C
6.	B	D
6.	D	F
7.	A	B
7.	A	E
8.	C	D
9	C	C

Graph contains 6 vertices and 14 edges, So MST formed will be having (6-1) = 5 edges

Now forming the tree ~~so~~ so that there will be no cycle formed.



Therefore final MST will be