ABES Engineering College, Ghaziabad

B.Tech Odd Semester Sessional Test-1

Printed Pages: 02
Session: 2022-23
Roll No.
Date of Exam: 19th Oct 22
Time: 2 hours

Course Code: KCS 503
Course Name: Design and Anlaysis of Algorithms
Maximum Marks: 75

Instructions:
1. Attempt All sections.
2. If require any missing data, then choose suitably.

Q.No.	Question	Marks	CO	KL	PI
	Section-A				
1	**Attempt ALL Parts**	(5x2=10)			
a)	Can the Master method be applied to solve recurrence T(n)=2T(n/2)+nlogn? Why or why not? (GATE 2008)	2	CO1	K3	2.1.3
b)	Show that n element heap has a height atmost ⌊logn⌋ (GATE 2011)	2	CO1	K3	1.4.1.
c)	If x is a nonroot node in a binomial tree within a binomial heap, how does degree[x] compare to degree[p[x]]?	2	CO2	K2	2.3.1
d)	Suppose a node is being inserted in to Red Black Tree using the RB TREE INSERT FIX UP & then same node is being deleted with RB TREE DELETE FIX UP. Is the Red Black is same as Intial Red Black Tree. (GATE 2009)	2	CO2	K2	1.3.1
e)	Discuss about the Naïve Matrix Multiplication alogorithm with its complexity	2	CO3	K1	2.1.3
	Section-B				
2	**Attempt ANY ONE part from the following**	(1x5=5)			
a)	Explain in brief various asymptotic notations and give their significance. Let f (n) & g (n) be asymptotic positive functions. Prove or disprove following conjectures: (GATE 2014) a. f(n)=O(g(n)) implies g(n)=O(f(n)) b. f(n)+g(n)=theta(min(f(n),g(n))	5	CO1	K3	2.3.1
b)	Why do we want the loop index i in line 2 of BUILD-MAX-HEAP to decrease from ⌊length[A]/2⌋ to 1 rather than increase from 1 to ⌊length[A]/2⌋? Explain with Example.	5	CO1	K3	1.4.1
3	**Attempt ANY ONE part from the following**	(1x5=5)			
a)	Discuss the relationship between inserting into a Binomial Heap & incrementing a binary number and the relationship between uniting two binomial Heaps and adding two binary numbers.	5	CO2	K2	2.1.3
b)	Prove that red black tree with n internal nodes has height at most 2log(n+1).	5	CO2	K2	2.1.3
4	**Attempt ANY ONE part from the following**	(1x5=5)			
a)	Discuss about the Convex hull algorithm with the help of an example.	5	CO3	K2	2.1.3
b)	What is the largest number of key comparisons made by the binary search in searching for the key in the following arrray? A= (3, 14, 27, 31, 39, 42, 55, 70, 74, 81, 85, 93 and 98)	5	CO3	K3	1.4.1
	Section-C				
5	**Attempt ANY ONE part from the following**	(1x10=10)			
a)	Suppose that the for loop header in line 9 of the COUNTING-SORT procedure is rewritten as for j ← 1 to length[A] .Show that the algorithm still works properly. Is the modified algorithm stable? (GATE 2012)	10	CO1	K3	2.1.3
b)	Suppose that the splits at every level of quicksort are in the proportion 1 - α to α, where $0 < \alpha \leq 1/2$ is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately $- \lg n/ \lg \alpha$ and the maximum depth is approximately $-\lg n/ \lg(1 - \alpha)$. (Don'tworry about integer round-off.) (GATE 2016)	10	CO1	K3	1.3.1
6	**Attempt ANY ONE part from the following**	(1x10=10)			
a)	Which of the following sorting algorithm(s) are stable: insertion sort, merge sort, heap sort, and quick sort? Argue that any comparison based sorting algorithm can be made stable without effecting the running time by more than a constant factor	10	CO1	K3	2.1.3

b)	Draw recursion tree for the operation of merge sort on array A= { 3, 41, 52, 26, 38, 57, 9, 49 } i) How many levels are there? ii)How many comparisons are done at each level? iii) What is total number of comparisons needed? iv) Generalize i to iii for any n in terms of O()?	10	CO1	K3	2.1.2
7	**Attempt ANY ONE part from the following**	**(1x10=10)**			
a)	Show the result of inserting the following keys in an initially empty B-tree of order 5. Keys are 25,31,38,76,5,60,38,8,30,15,35,17,23,53,27,43,65,48.	10	CO2	K3	2.3.1
b)	Let A=(7,2,4,17,1,11,6,8,15,10,20) (i) Draw a binomial heap whose keys are the elements of A. (ii) Insert the new element with Key 5 into this heap. (iii) To a binomial heap obtained this way, apply the operation of extracting the node with minimum key two times. After each change in the structure of the heap. draw its current diagram.	10	CO2	K3	1.4.1
8	**Attempt ANY ONE part from the following**	**(1x10=10)**			
a)	Show the red black tree that result after inserting the keys 5,10,15,20,25,30 and 35 into an initially empty Red-Black Tree	10	CO2	K3	2.3.1
b)	Discuss the Procedure for extracting the minimum node from the Fibonacci Heap. Also, Write Fibonacci Heap Extract Min & Consolidate Algorithm.	10	CO2	K2	1.3.1
9	**Attempt ANY ONE part from the following**	**(1x10=10)**			
a)	Show all the steps of Strassen's matrix multiplication algorithm to multiply the following matrices $X = \begin{matrix} 2 & 5 \\ 4 & 3 \end{matrix}$ $\quad Y = \begin{matrix} 2 & 6 \\ 5 & 4 \end{matrix}$	10	CO3	K3	2.4.1
b)	Compute 2101*1130 by applying divide and conquer algorithm outlined in the text.	10	CO3	K3	1.4.1

CO Course Outcomes mapped with respective question

KL Bloom's knowledeg Level (K1, K2, K3, K4, K5, K6)

K1- Remember, K2- Understand, K3-Apply, K4- Analyze, K5: Evaluate, K6- Create

Printed Pages:

ABES Engineering College, Ghaziabad

B.Tech Odd Semester Sessional Test-1

Session: 2022-23
Roll No.
Date of Exam: 19th Oct 22
Time: 2 hours

Course Code: KCS 503
Course Name: Design and Anlaysis of Algorithms
Maximum Marks: 75

Instructions:
1. Attempt All sections.
2. If require any missing data, then choose suitably.

Q.No.	Question	Marks	CO	KL	PI
	Section-A	**(5x2=10)**			
1	**Attempt ALL Parts**				
a)	Can the Master method be applied to solve recurrence T(n)=2T(n/2)+nlogn? Why or why not? **(GATE 2008)**	2	CO1	K3	2.1.3
b)	Show that n element heap has a height atmost ⌊logn⌋ **(GATE 2011)**	2	CO1	K3	1.4.1.
c)	If x is a nonroot node in a binomial tree within a binomial heap, how does degree[x] compare to degree[p[x]]?	2	CO2	K2	2.3.1
d)	Suppose a node is being inserted in to Red Black Tree using the RB TREE INSERT FIX UP & then same node is being deleted with RB TREE DELETE FIX UP. Is the Red Black is same as Intial Red Black Tree. **(GATE 2009)**	2	CO2	K2	1.3.1
e)	Discuss about the Naïve Matrix Multiplication alogorithm with its complexity	2	CO3	K1	2.1.3
	Section-B	**(1x5=5)**			
2	**Attempt ANY ONE part from the following**				
a)	Explain in brief various asymptotic notations and give their significance. Let f (n) & g (n) be asymptotic positive functions. Prove or disprove following conjectures: **(GATE 2014)** a. f(n)=O(g(n)) implies g(n)=O(f(n)) b. f(n)+g(n)=theta(min(f(n),g(n))	5	CO1	K3	2.3.1
b)	Why do we want the loop index i in line 2 of BUILD-MAX-HEAP to decrease from ⌊length[A]/2⌋ to 1 rather than increase from 1 to ⌊length[A]/2⌋? Explain with Example	5	CO1	K3	1.4.1
		(1x5=5)			
3	**Attempt ANY ONE part from the following**				
a)	Discuss the relationship between inserting into a Binomial Heap & incrementing a binary number and the relationship between uniting two binomial Heaps and adding two binary numbers.	5	CO2	K2	2.1.3
b)	Prove that red black tree with n internal nodes has height at most 2log(n+1).	5	CO2	K2	2.1.3
		(1x5=5)			
4	**Attempt ANY ONE part from the following**				
a)	Discuss about the Convex hull algorithm with the help of an example.	5	CO3	K2	2.1.3
b)	What is the largest number of key comparisons made by the binary search in searching for the key in the following arrray? A= (3, 14, 27, 31, 39, 42, 55, 70, 74, 81, 85, 93 and 98)	5	CO3	K3	1.4.1
	Section-C	**(1x10=10)**			
5	**Attempt ANY ONE part from the following**				
a)	Suppose that the for loop header in line 9 of the COUNTING-SORT procedure is rewritten as for j ← 1 to length[A] .Show that the algorithm still works properly. Is the modified algorithm stable? **(GATE 2012)**	10	CO1	K3	2.1.3
b)	Suppose that the splits at every level of quicksort are in the proportion 1 - α to α, where 0 < α ≤ 1/2 is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately - lg n/ lg α and the maximum depth is approximately -lg n/ lg(1 - α). (Don'tworry about integer round-off.) **(GATE 2016)**	10	CO1	K3	1.3.1
		(1x10=10)			
6	**Attempt ANY ONE part from the following**				
a)	Which of the following sorting algorithm(s) are stable: insertion sort, merge sort, heap sort, and quick sort? Argue that any comparison based sorting algorithm can be made stable without effecting the running time by more than a constant factor	10	CO1	K3	2.1.3
b)	Draw recursion tree for the operation of merge sort on array A= { 3, 41, 52, 26, 38, 57, 9, 49 } i) How many levels are there? ii)How many comparisons are done at each level? iii) What is total number of comparisons needed? iv) Generalize i to iii for any n in terms of O()?	10	CO1	K3	2.1.2
		(1x10=10)			
7	**Attempt ANY ONE part from the following**				
a)	Show the result of inserting the following keys in an initially empty B-tree of order 5. Keys are 25,31,38,76,5,60,38,8,30,15,35,17,23,53,27,43,65,48.	10	CO2	K3	2.3.1
b)	Let A=(7,2,4,17,1,11,6,8,15,10,20) (i) Draw a binomial heap whose keys are the elements of A. (ii) Insert the new element with Key 5 into this heap. (iii) To a binomial heap obtained this way, apply the operation of extracting the node with minimum key two times. After each change in the structure of the heap. draw its current diagram.	10	CO2	K3	1.4.1

8	**Attempt ANY ONE part from the following**	**(1x10=10)**			
a)	Show the red black tree that result after inserting the keys 5,10,15,20,25,30 and 35 into an initially empty Red-Black Tree	10	CO2	K3	2.3.1
b)	Discuss the Procedure for extracting the minimum node from the Fibonacci Heap. Also, Write Fibonacci Heap Extract Min & Consolidate Algorithm.	10	CO2	K2	1.3.1
9	**Attempt ANY ONE part from the following**	**(1x10=10)**			
a)	Show all the steps of Strassen's matrix multiplication algorithm to multiply the following matrices $X = \begin{matrix} 2 & 5 \\ 4 & 3 \end{matrix}$ $Y = \begin{matrix} 2 & 6 \\ 5 & 4 \end{matrix}$	10	CO3	K3	2.4.1
b)	Compute 2101*1130 by applying divide and conquer algorithm outlined in the text.	10	CO3	K3	1.4.1

CO Course Outcomes mapped with respective question

KL Bloom's knowledeg Level (K1, K2, K3, K4, K5, K6)

K1- Remember, K2- Understand, K3-Apply, K4- Analyze, K5: Evaluate, K6- Create

Section-A

1 (a) Can Master Method be applied to solve Recurrence

$T(n) = 2T(n/2) + n\log n$? Why or why not.

Ans Master Method can not be applied to solve

$$T(n) = 2T(n/2) + n\log n.$$

$$n^{\log_b a} = n^{\log_2 2} = n^1$$

$$f(n) = n\log n$$

$$f(n) > n^{\log_b a}$$

function $f(n)$ is asymptotically larger than $n^{\log_b a} = n$

but Problem is that it is not Polynomially larger.

Ratio $\dfrac{f(n)}{n^{\log_b a}} \Rightarrow \dfrac{n\log n}{n} \Rightarrow \log n$. is

Consequently, Recurrence falls into gap between Case 2 and Case 3.

So, Master theorem can not apply for this recurrence.

(2) Marks

Qus 1(b) Show that n element heap has a height at most $\lfloor \log n \rfloor$.

Ans We know that minimum and maximum number of nodes in a heap of height h range from 2^h to $2^{h+1}-1$ i.e.

$$2^h \leq n \leq 2^{h+1}-1$$

where n is number of nodes in a heap.

After taking log both side

$$n \leq \log n \leq h+1$$

$\boxed{\text{therefore height of heap is } \lfloor \log n \rfloor}$ —— 2 Marks

Qus 1(c) If x is a non root node in a binomial tree with in a binomial heap, how does degree [x] compare to degree [P[x]].

Ans If x is a non root node in a binomial tree with in a binomial heap then

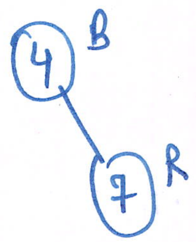
$$\text{degree}[P[x]] = \text{degree}[x] + 1$$

—— 2 Marks

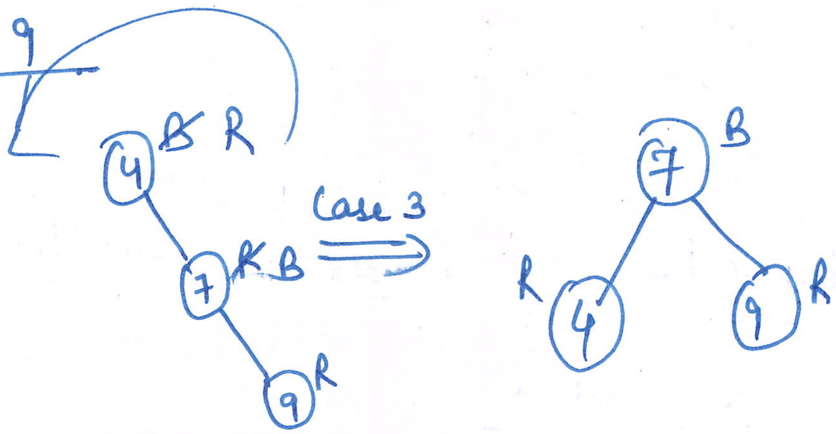
Qus 1(d) Suppose a node is being inserted into Red Black tree using RB-Tree-fixup and then same node is being deleted with RB-Tree delete fixup. Is Red Black is same as initial Red Black tree.

Ans For Example

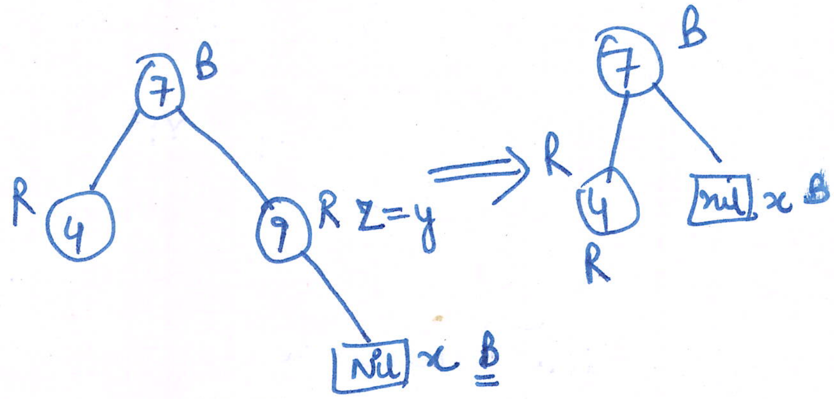
Initial R-B Tree is



Now Insert 9



Now Delete -9



Now Color [y] = Red So No fixup will be called.

So Resultant tree will not be same as initial Red Black tree.

—— ② Marks

Ans 1(e) Discuss about Naive Matrix multiplication algorithm with its complexity

Ans
$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

So in Naive Matrix Multiplication we have 8 Multiplication & 4 addition operation.

So Recurrence will be

$$T(n) = \begin{cases} O(1), & \text{if } n \leq 1 \\ 8T\left(\frac{n}{2}\right) + n^2, & \text{if } n > 1 \end{cases}$$

Solve This Recurrence

$n^{\log_b a} \Rightarrow n^{\log_2 8} \Rightarrow n^3 > n^2$

Case 1 lies so

$\boxed{T(n) = \Theta(n^3)}$ —— ② Marks

Section - B

Ques ② (a) Explain asymptotic Notations & their Significance

f(n) & g(n) be asymptotic Positive functions. Prove or disprove following Conjectures.

a) $f(n) = O\,g(n)$ implies $g(n) = O\,f(n)$

b) $f(n) + g(n) = \theta(\min(f(n), g(n)))$

Ans **Asymptotic notations**

Big oh → worst case (O)

Omega → Best case (Ω)

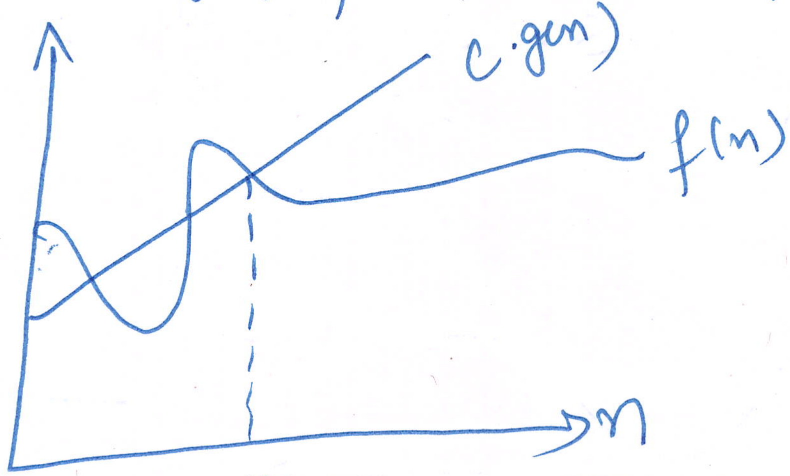
Theta → Average case (θ)

Big oh :→ Indicates Maximum time required by an algorithm.

→ It describes worst case of an algorithm.

→ function $f(n) = O(g(n))$ iff there exists a Positive constants c & n_0 such that

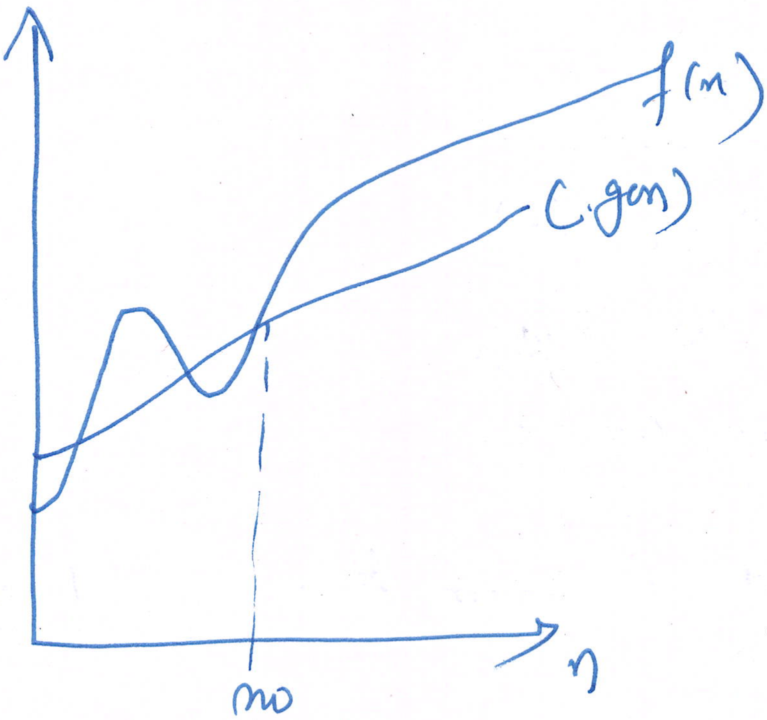
$$f(n) \le c \cdot g(n) \text{ for all } n,\ n \ge n_0$$



Omega (Ω) → It describes minimum time required by an algo. for all I/P.

→ It describes best case of algorithm.

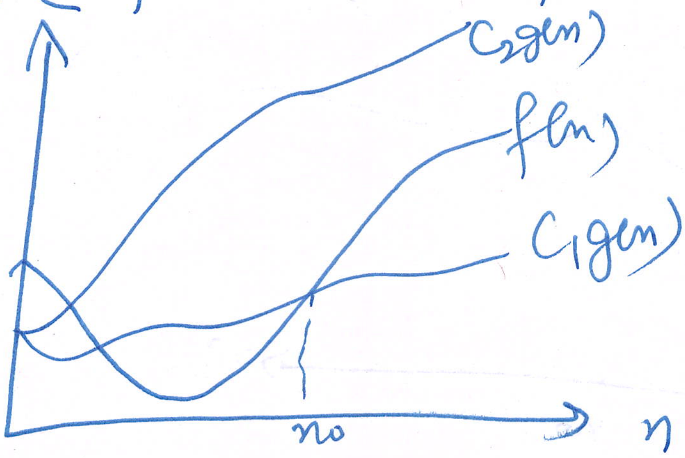
→ $f(n) = \Omega\, g(n)$ iff there exists a positive constants c and n_0 such that $f(n) \geq c \cdot g(n)$ for all $n, n \geq n_0$.



Theta (θ) →

$f(n) = \theta(g(n))$ iff there exists positive constants c_1, c_2 & n_0 such that

$c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n, n \geq n_0$



2 Marks

(a) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$

This Conjecture is false

Let $f(n) = n$, $g(n) = n^2$

$f(n) = O(g(n))$ $\qquad$ $g(n) = O(f(n))$

$n = O(n^2)$ $\qquad$ but $n^2 \neq O(n)$

n^2 is upper bound on n but n is not upper bound on n^2. ——— 1.5 Marks

(b) ~~Let~~ $f(n) + g(n) = \Theta(\min(f(n), g(n)))$

This Conjecture is false

Lets take $f(n) = n$, $g(n) = n^2$

$n^2 + n \neq \Theta(\min(n^2, n))$

$\boxed{n^2 + n \neq \Theta(n)}$ ——— 1.5 Marks

Que 2(b) Why do we want the loop index i in line 2 of Build Max heap to decrease from $\lfloor length[A]/2 \rfloor$ to 1 rather than increase from $\lfloor 1$ to $length[A]/2 \rfloor$. Explain with Example

Ans → Max-heapify (A, i) works under the assumption that subtrees rooted at two children of A[i] : A[left(i)] and A[right(i)] are both max heaps.

→ Now all nodes A[x] such that $x > n/2$ are leaf nodes and therefore are also max heaps. —— 2 Marks

→ So going in reverse order A[n/2] is the first non leaf node, ie the subtrees rooted at all nodes A[y] : $1 <= y <= n/2$ might not max heaps and Max-heapify must be performed on each of them.

→ Looping in reverse order ensures children of node are heapified before the node and necessary requirement for Max-heapify is fulfilled for each node.

→ if we have to loop from 1 to length[A/2] then necessary assumption for Max-heapify ( ) to work won't be true for each node. It will not result in a max-heap at all. —— 3 Marks

Ans 4(a) Discuss about Convex Hull with Example

Ans → Convex hull is line completely enclosing a set of Points in a Plane so that there are no concavities in line.

→ We can describe it as smallest convex Polygon which encloses a set of Points ~~a set~~ such that each Point in set lies within in Polygon. —— 1 Mark

Algo

1. Find P_0, the Point with minimum y coordinate
2. Sort all remaining Points in order of their Polar angle from P_0
3. Initialize a Stack S
4. Push(S, P_0)
5. Push (S, P_1)
6. Push (S, P_2)
7. for i = 3 to n do
8. while (angle formed by topnext(S), top(S) and P_i makes a right turn)
9. Pop(S)
10. Push (S, P_i)
11. return S. —— 3 Marks

This algo will take O(n logn) time. —— 1 Mark

Ques 4 (B) What is largest no. of key comparisions made by binary search in searching for key in following array.

A = {3, 14, 27, 31, 39, 42, 55, 70, 74, 81, 85, 93, 98}

n = 13

Ans Largest no. of key comparisions — 5 Marks

$= \log_2 (n+1)$

$= \log_2 (13+1)$

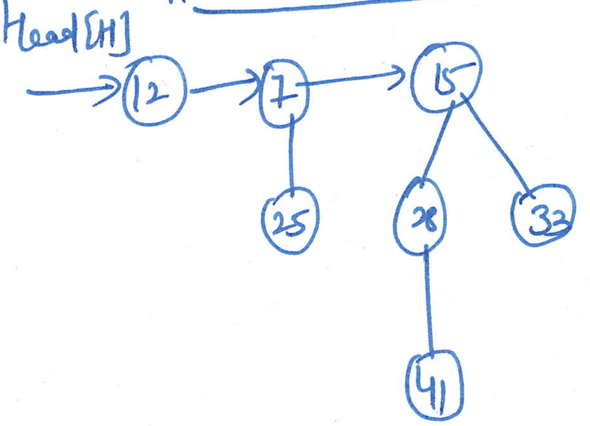
$= \log_2 (14)$

$= 3.80 \simeq 3$

Ques 3 (a) Discuss Relationship between inserting into Binomial Heap & incrementing a binary number and relationship between uniting two Binomial Heaps & adding two binary numbers

Ans

Relation between Inserting a element & incrementing binary number



Total no. of nodes = 7

Binary Representation

= 7

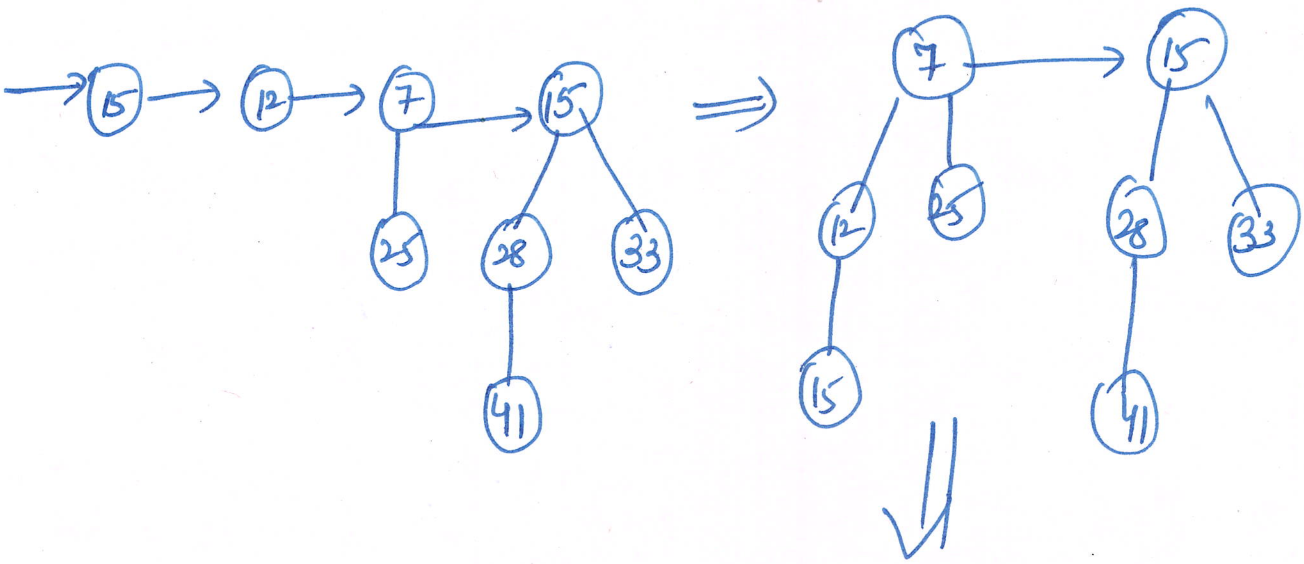
= 1 1 1

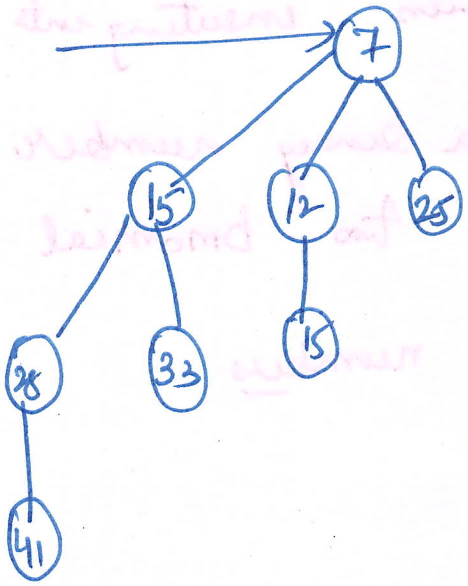
$\uparrow$ $\uparrow$ $\uparrow$

B_2 B_1 B_0

So binomial Heap is having B_0, B_1 & B_2 degree Binomial tree. —— 2 1/2 Marks

If we insert 15 in Heap we get





Now no. of nodes

8

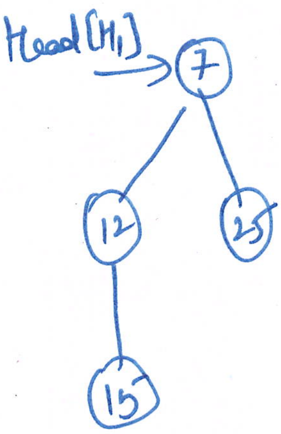
1000

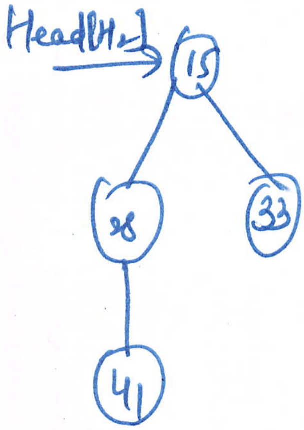
B_3 B_2 B_1 B_0

So After adding 15 Binary number is incremented by 1

2 1/2 Marks

Relationship between uniting two heaps and adding two Binary numbers





Total nodes → 4

0100

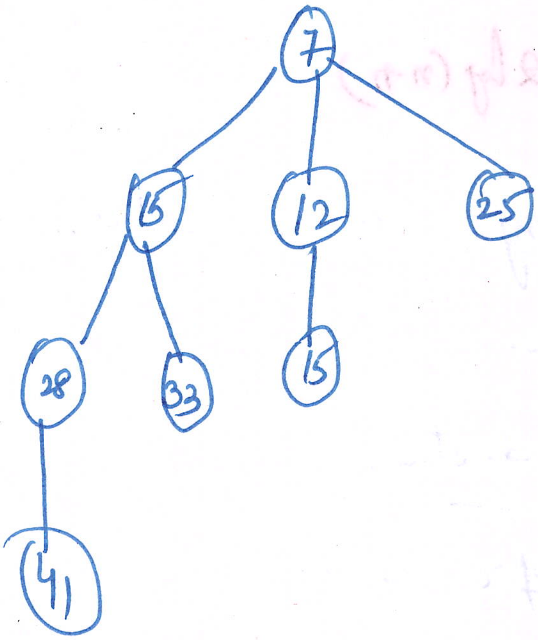
B_2 B_1 B_0

Total nodes → 4

0100

B_2 B_1 B_0

After union



Total nodes $\rightarrow$ 8

$$\begin{array}{cccc}1&0&0&0\\\downarrow&\downarrow&\downarrow&\downarrow\\B_3&B_2&B_1&B_0\end{array}$$

So adding

$$\begin{array}{r}0100\\0100\\\hline 1000\end{array} \rightarrow 8$$

Hence Proved

~~3 Marks~~

Qus 3(b) Prove that Red black tree with n internal nodes has height at most $2\log(n+1)$

Ans Sub tree rooted at any node x contains at least $2^{bh(x)}-1$ internal nodes.

→ We can prove this by induction.

If x is nil[T] ie height of x is 0.

$$2^{bh(x)}-1 = 2^0-1$$
$$= 0 \text{ internal nodes}$$

——— 1 Mark

→ Consider an internal node x with black height $bh(x)$ with two children. Each child has black height of either $bh(x)$ or $bh(x)-1$, depending on whether its color is red or black.

Assume $bh(x)-1$ to be conservative

So each child has at least $2^{bh(x)-1}-1$ internal nodes so subtree rooted at x contains at least $(2^{bh(x)-1}-1)+(2^{bh(x)-1}-1)+1$

$$= 2^{bh(x)}-1 \text{ internal nodes}$$

——— 2 Marks

→ Let h be height of tree rooted at x

so at least half of nodes on any Path from root to leaf must be black.

So $bh(x) \geq h/2$

$n \geq 2^{h/2} - 1$

$n+1 \geq 2^{h/2}$

$\log(n+1) \geq h/2$

$\boxed{h \leq 2 \log(n+1)}$ ——— 2 Marks

Section C

Q 5(a) Suppose that the for loop header in line 9 of Counting Sort procedure is re-written as for $j \leftarrow 1$ to length[A]. Show that algorithm still works. Is Modified algorithm stable.

Ans Modified Counting Sort

Counting Sort(A, B, k)

1. for $i \leftarrow 0$ to k
2. $C[i] \leftarrow 0$
3. for $i \leftarrow 1$ to length[A]
4. $C[A[i]] \leftarrow C[A[i]] + 1$
5. for $i \leftarrow 1$ to k
6. $C[i] \leftarrow C[i] + C[i-1]$
7. $\boxed{\text{for } i \leftarrow 1 \text{ to length}[A]}$ ← Modified Step
8. $B[C[A[i]]] \leftarrow A[i]$
9. $C[A[i]] \leftarrow C[A[i]] - 1$.

→ 5 Marks

Algorithm will still work but modified algorithm will not stable.

Let A =

1	2	3	4	5	6	7
3	0	2	5	2	1	4

B =

1	2	3	4	5	6	7

C =

0	1	2	3	4	5

$K = 5$

Step 1 :→

Initialize

C =

0	1	2	3	4	5
0	0	0	0	0	0

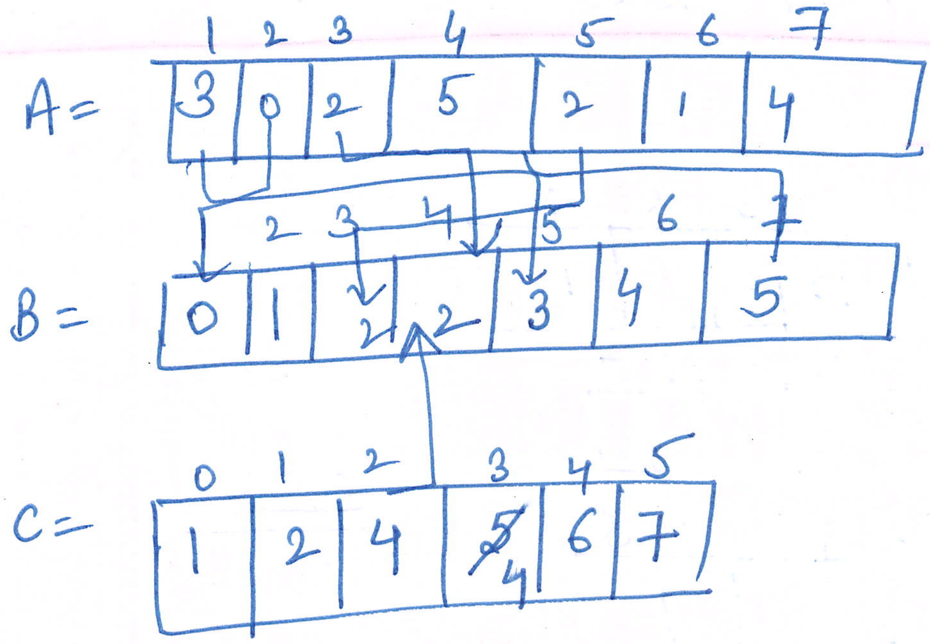
Step-2 :→

C =

0	1	2	3	4	5
1	1	2	[illegible]	1	1

C =

0	1	2	3	4	5
1	2	4	5	6	7



for $i \leftarrow 1$ to length[A]

$i \leftarrow 1$ to 7

Element at position 3 in unsorted array will occur at position 4 in sorted array hence Modified Counting sort will not be stable.

→ 5 Marks

Qus 5(b)

Suppose that the splits at every level of quick sort are in proportion $1-\alpha$ to α where $0 < \alpha \leq 1/2$ is a constant. Show that minimum depth of a leaf in recursion tree is approximately $-\lg n/\lg \alpha$ and maximum depth is approximately $-\lg n/\lg(1-\alpha)$.

Ans → Minimum depth occurs for the path that always takes smaller portion of the split ie nodes that takes α proportion of work from parent node.

→ first node in path gets α proportion of work and second node gets α^2.

→ Recursion bottoms when size of data becomes 1.

→ Assume recursion ends at level h.

$$\alpha^h n = 1$$

$$h = \log_\alpha 1/n = \lg(1/n)/\lg \alpha = \boxed{-\lg n/\lg \alpha}$$

→ 5 Marks

Similarly Maximum depth m is similar with minimum depth

$$(1-\alpha)^m n = 1$$

$$m = \log_{1-\alpha} 1/n = \lg(1/n)/\lg(1-\alpha)$$

$$\boxed{m = -\lg n / \lg(1-\alpha)}$$

5 Marks

6 (a)

Ques Which of following sorting algorithms are stable : Insertion sort, Merge sort, heap sort, Quick sort. Argue any comparison based sorting algorithms can be made stable without affecting running time by more than a constant factor.

Ans Insertion sort → Stable

Merge sort → Stable

Heap sort → unstable

Quick sort → unstable.

→ 5 Marks

→ Any comparison based sorting algorithm which is not stable can be modified to stable by changing the key comparison operation so that the comparison of two keys considers position as a factor for objects or items with equal key values.

→ 5 Marks

Ques 6 (b)

Draw recursion tree for operation of merge sort on array A = {3, 41, 52, 26, 38, 57, 9, 49}

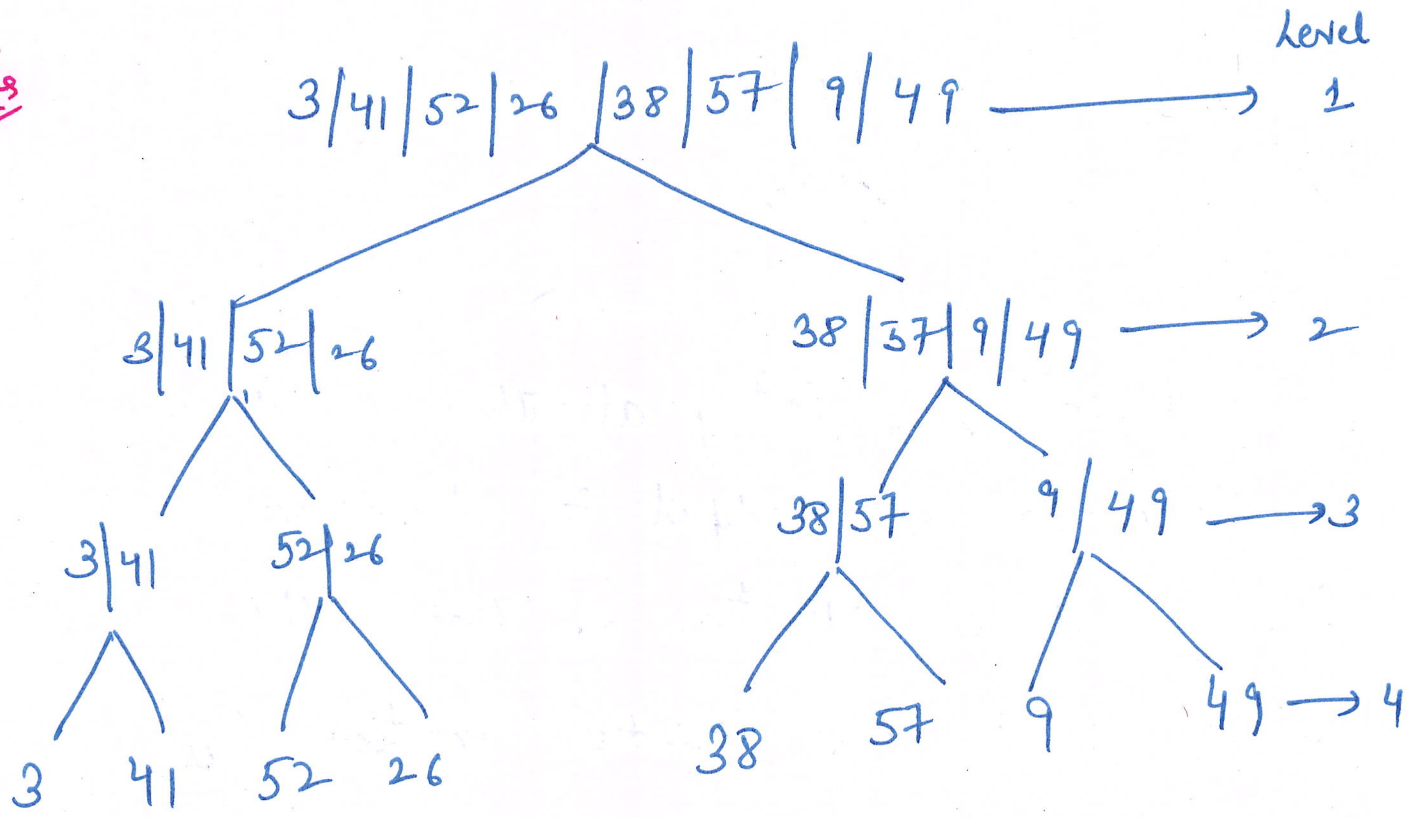
(i) How many levels are there?

(ii) How many comparisons are done at each level?

(iii) What is total numbers of comparison needed?

(iv) Generalize (i) to (iii) in terms of O()

Ans



(i) No. of levels → $\lfloor \log_2 n \rfloor + 1$

$= \lfloor \log_2 8 \rfloor + 1$

$= 3 + 1 = 4$

2/2

(II) Comparison done at each level in worst case :→ n 2½

$\rightarrow 8$

Comparison at level $4 \rightarrow 8$, Comparison at level $3 \rightarrow 8$
... comparison at level $2 \rightarrow 8$.

(III) Total no. of Comparisons $= n \log n$

$= 8 * \log_2 8$

$= 8 * 3 = 24$ 2½

(IV) Generalize

1. $\lfloor \log_2 n \rfloor + 1 \Rightarrow$ no. of levels
2. n comparison
3. $n \log n$ Total Comparisons 2½

Qus 7(a) Show the result of Inserting the following keys in an initially empty B-Tree of order 5. Keys are
25, 31, 38, 76, 5, 60, 38, 8, 30, 15, 35, 17, 23, 53, 27, 43, 65, 48

Ans

Order = 5

Maximum Keys = $m-1 = 5-1 = 4$

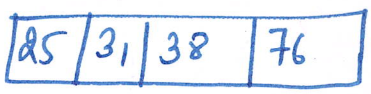
No. of children = $m = 5$

minimum keys = $\lceil \frac{m}{2} \rceil - 1 = \lceil 2.5 \rceil - 1 = 2$

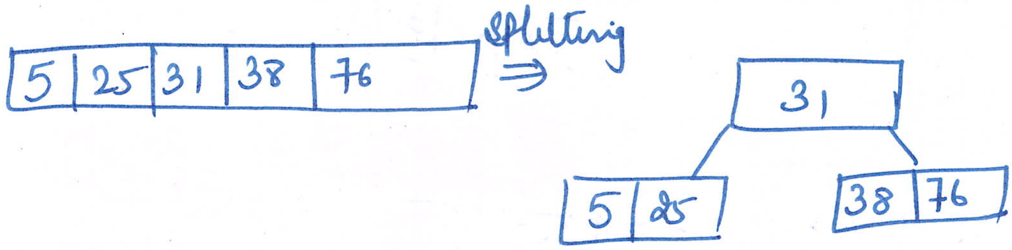
minimum no. of children = 3 $\longrightarrow$ 2 Marks

Insert 25.

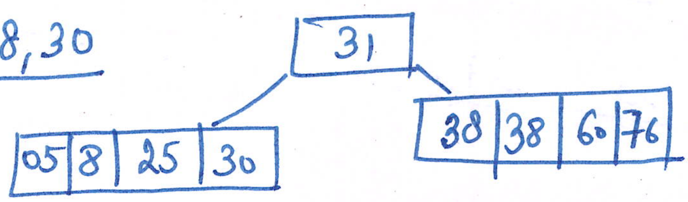
Insert 31, 38, 76



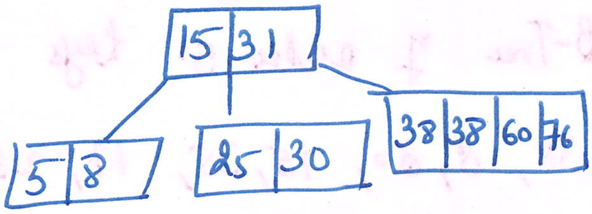
Insert 5



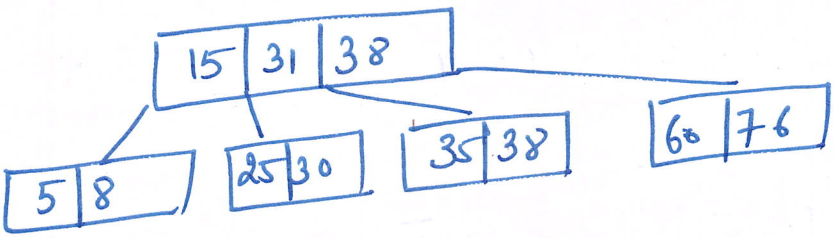
Insert 60, 38, 8, 30



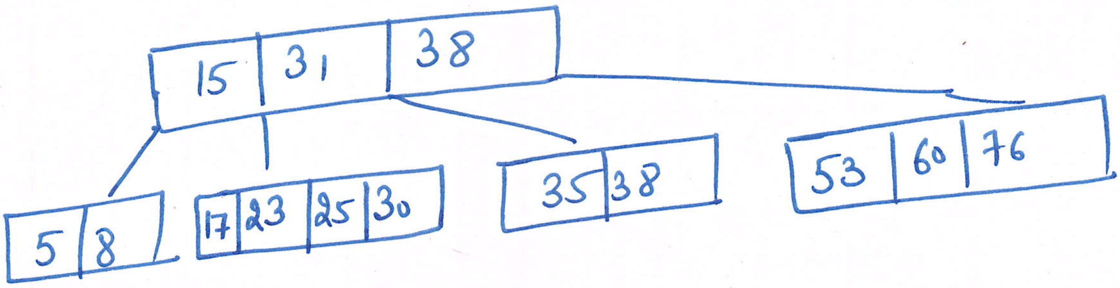
Insert 15



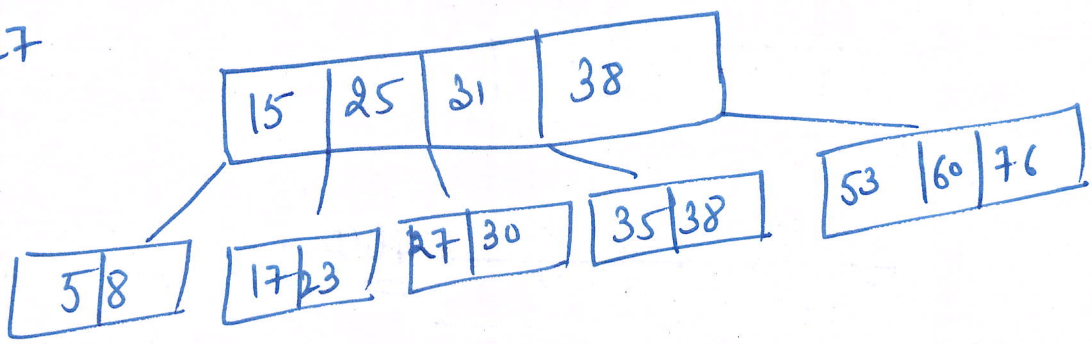
Insert 35



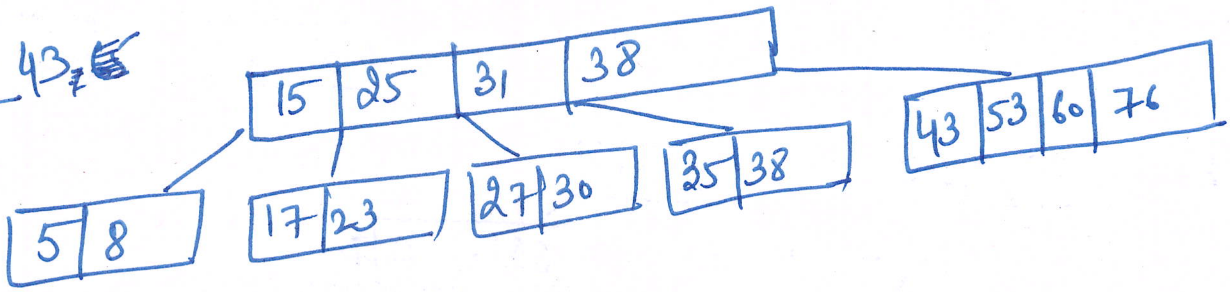
Insert 17, 23, 53



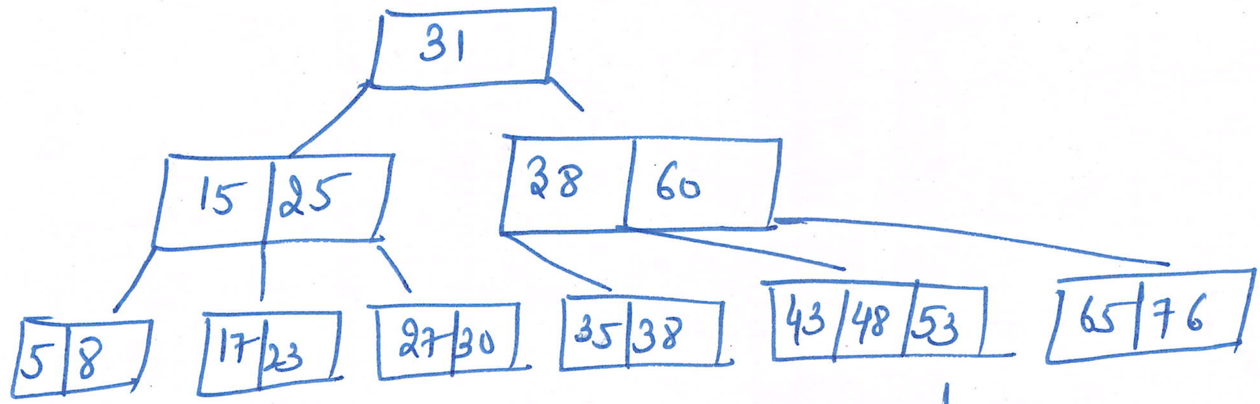
Insert 27



Insert 43,



Insert 65, 48



↳ 8 Marks

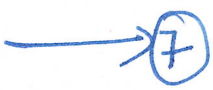
Ques 7(b) Let A = (7, 2, 4, 17, 1, 11, 6, 8, 15, 10, 20)

(i) draw a Binomial Heap whose elements of A.

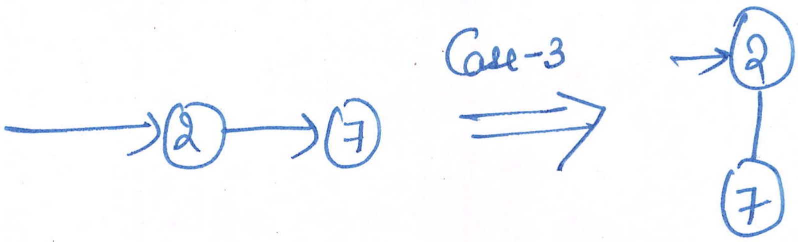
(ii) Insert a new element with key 5 into this heap.

(iii) To a Binomial heap obtained this way, apply the operation of extracting the node with minimum key two times. After each change in the structure of heap. draw its current diagram

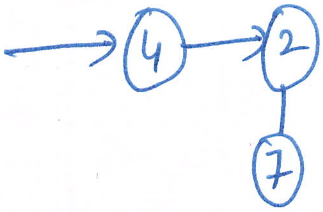
Ans Insert 7



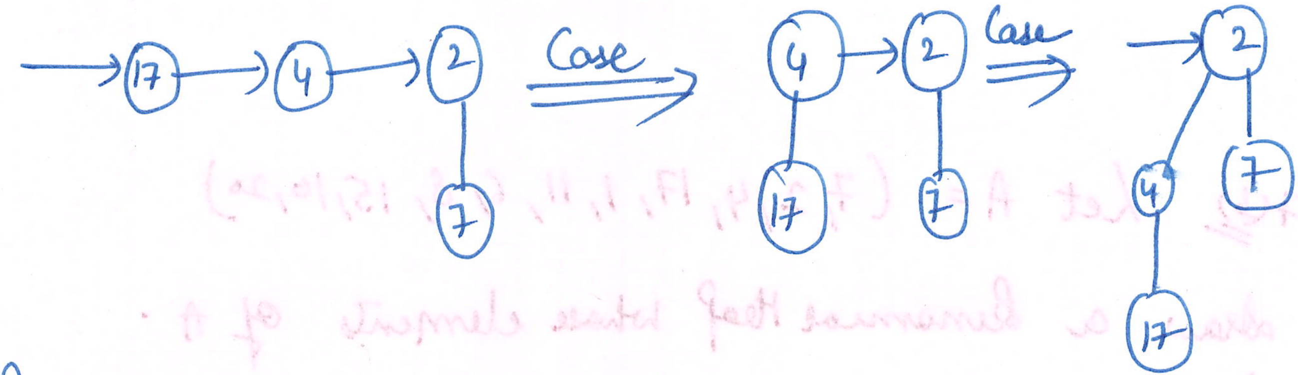
Insert 2



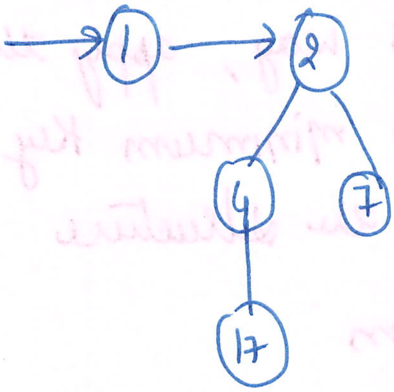
Insert 4



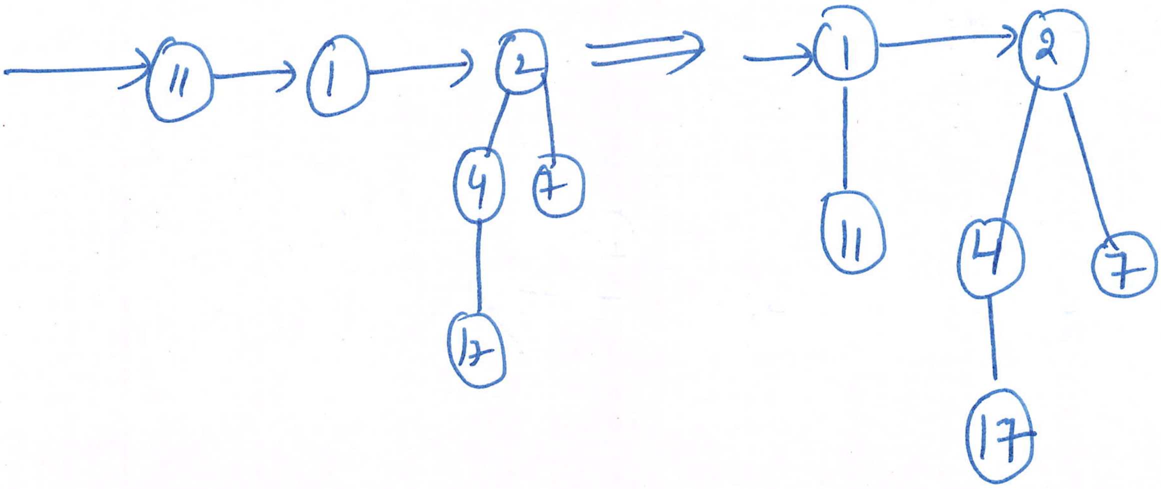
Insert 17



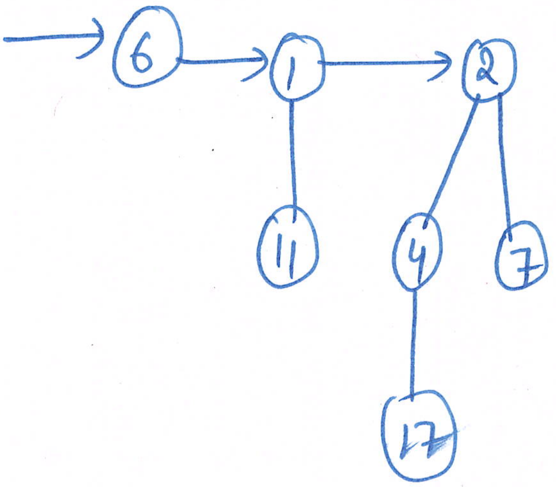
Insert 1



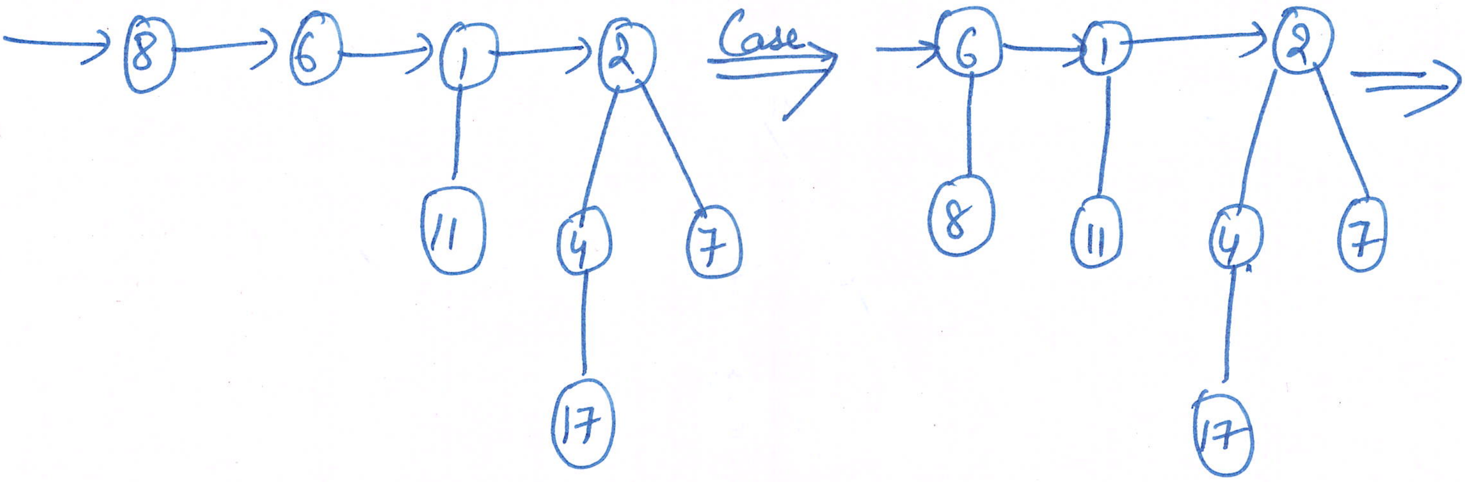
Insert 11 :>

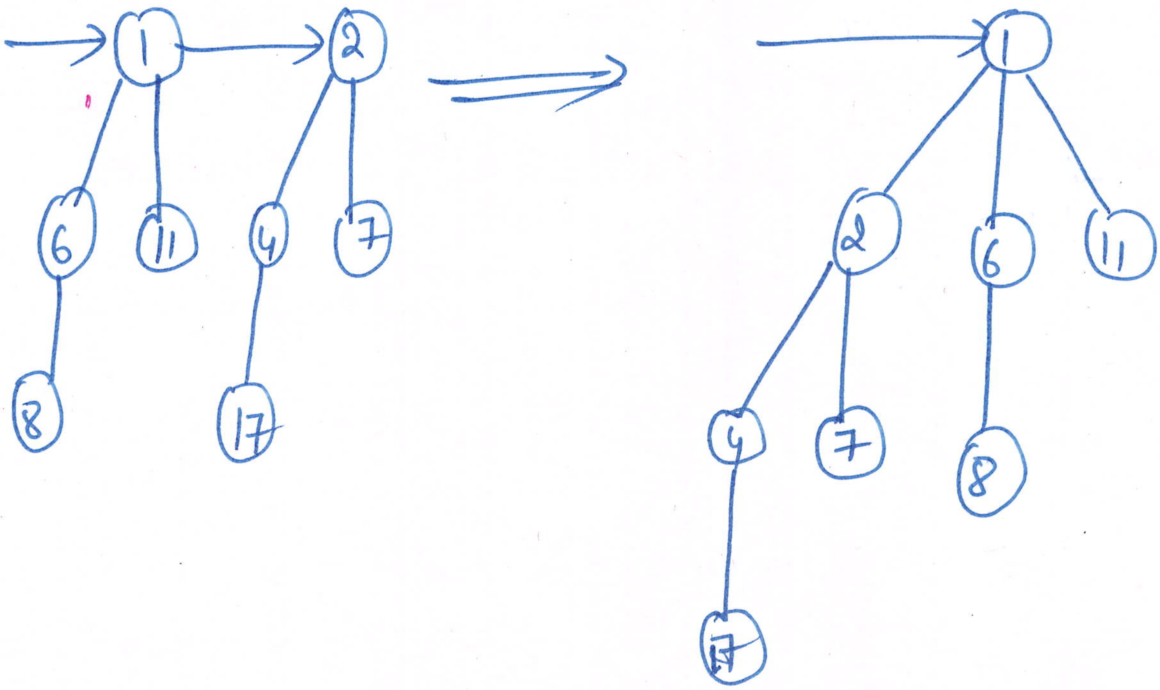


Insert 6

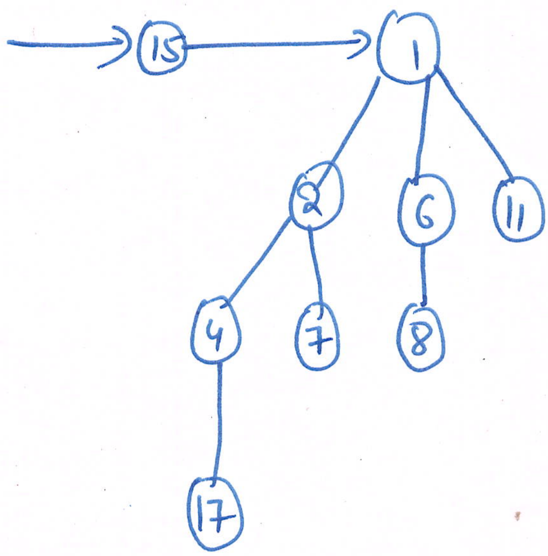


Insert 8

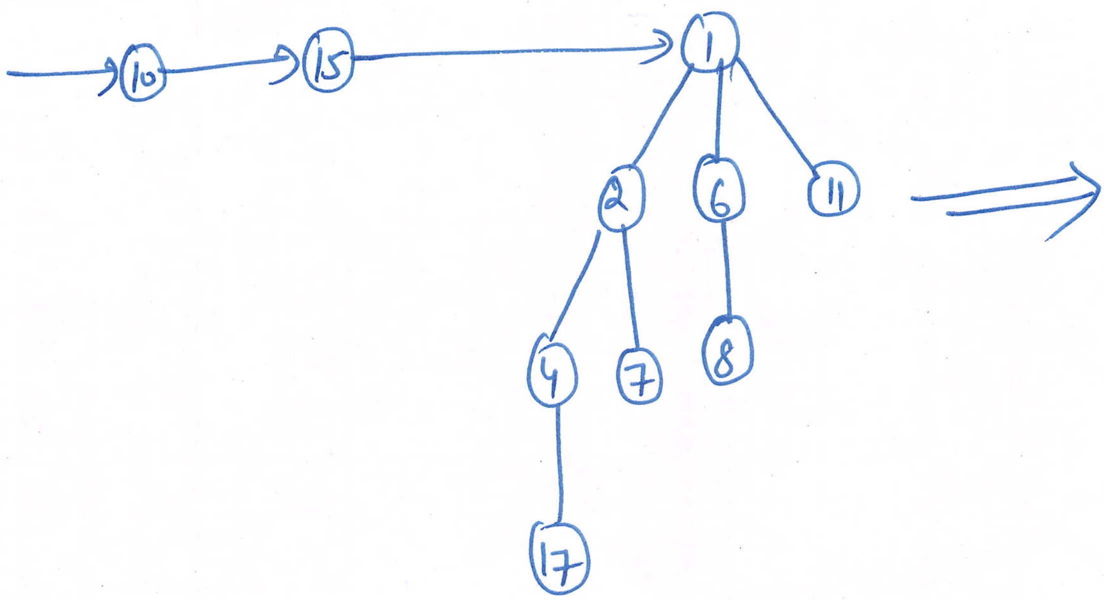


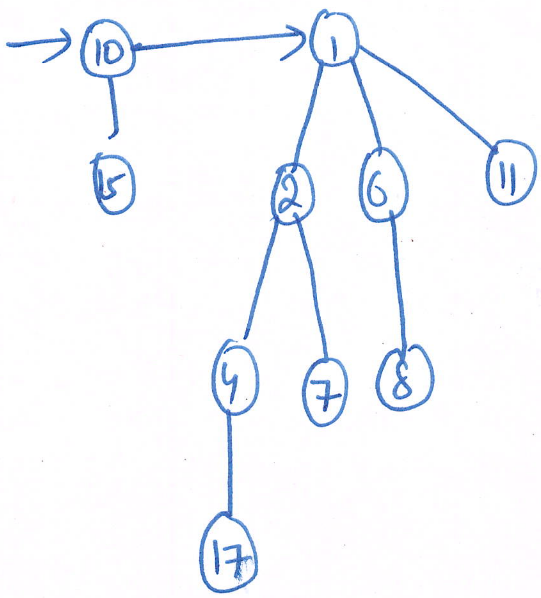


Insert 15

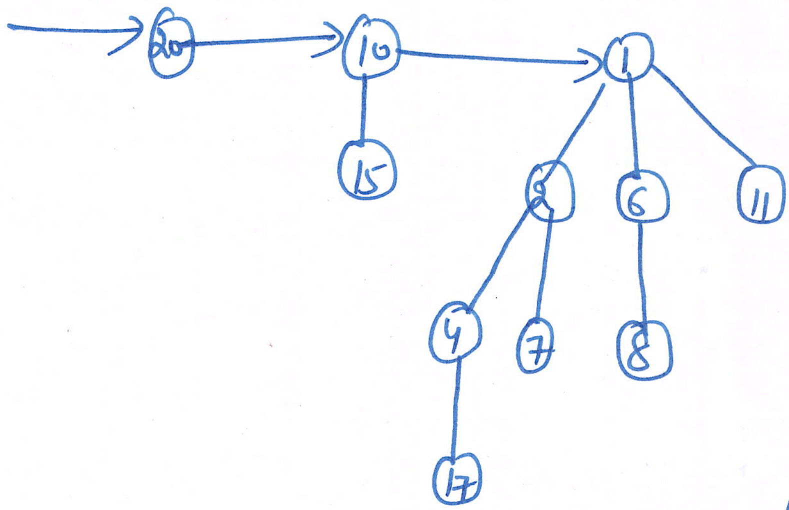


Insert 10





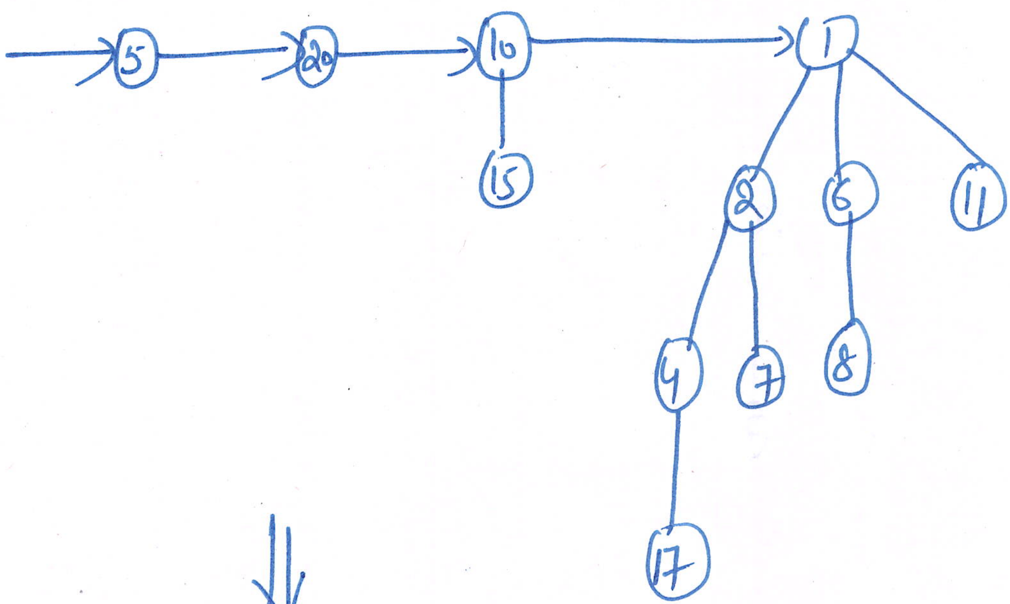
Insert 20

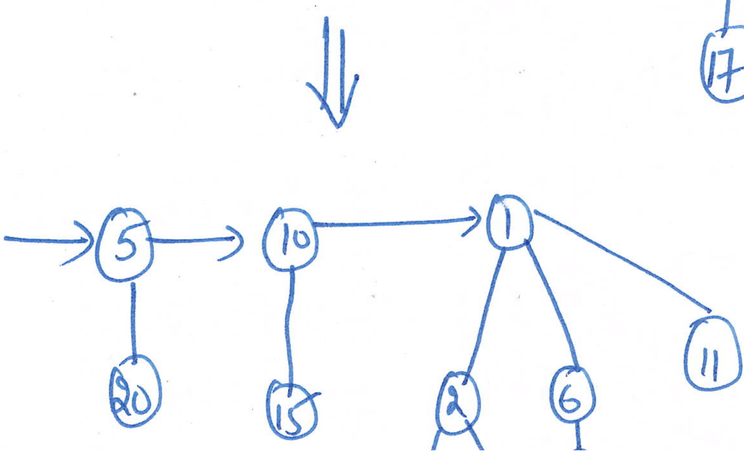


5 Marks

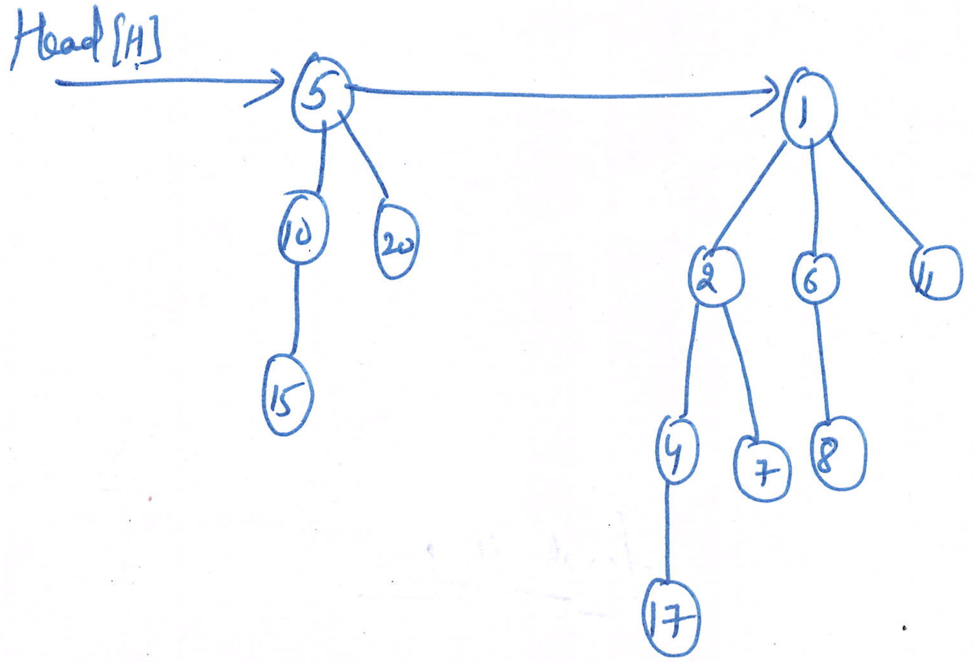
final Heap

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





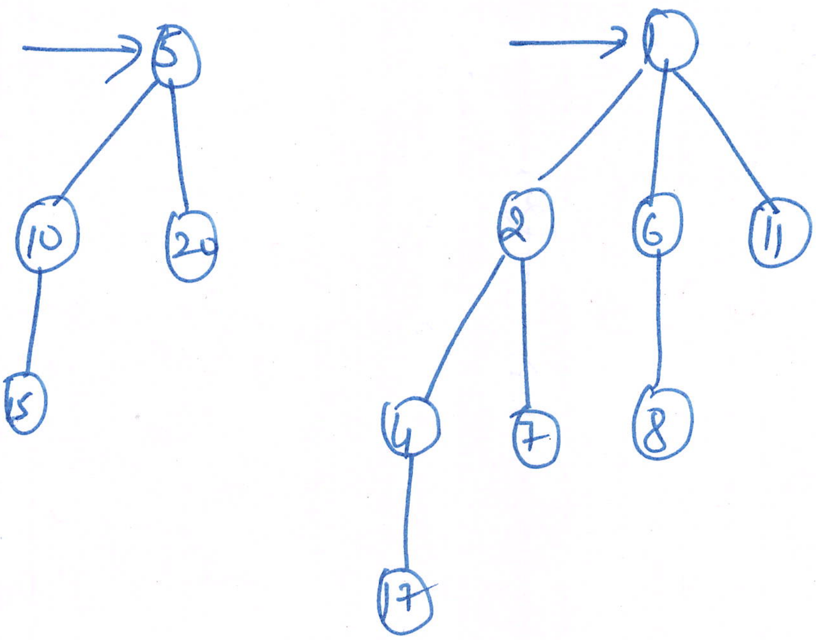
⇓

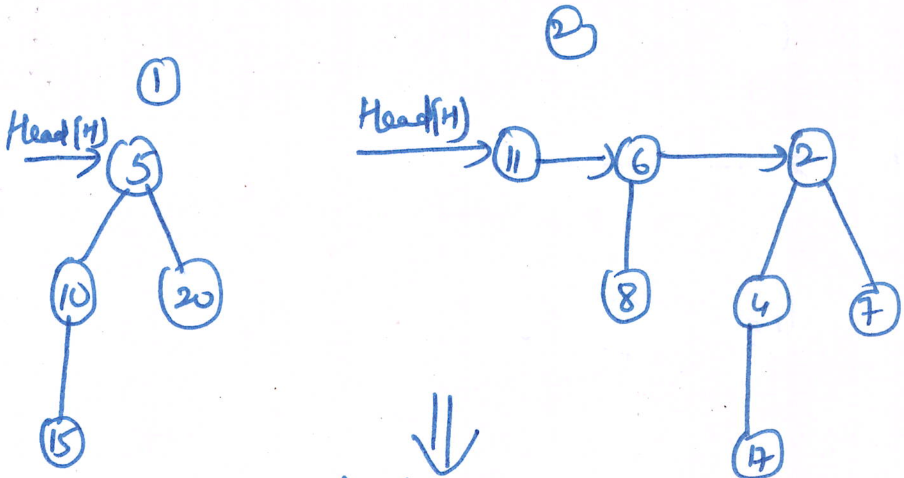


2½ Marks

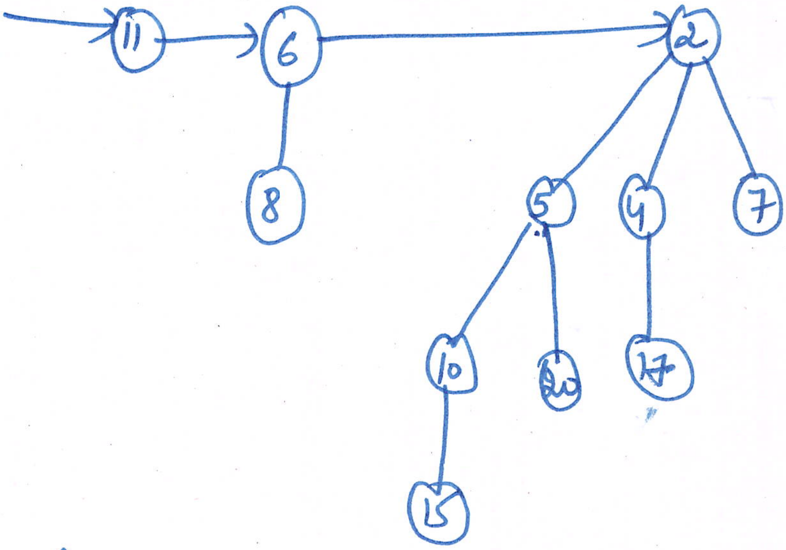
Final Binomial Heap after inserting 5

(iii)

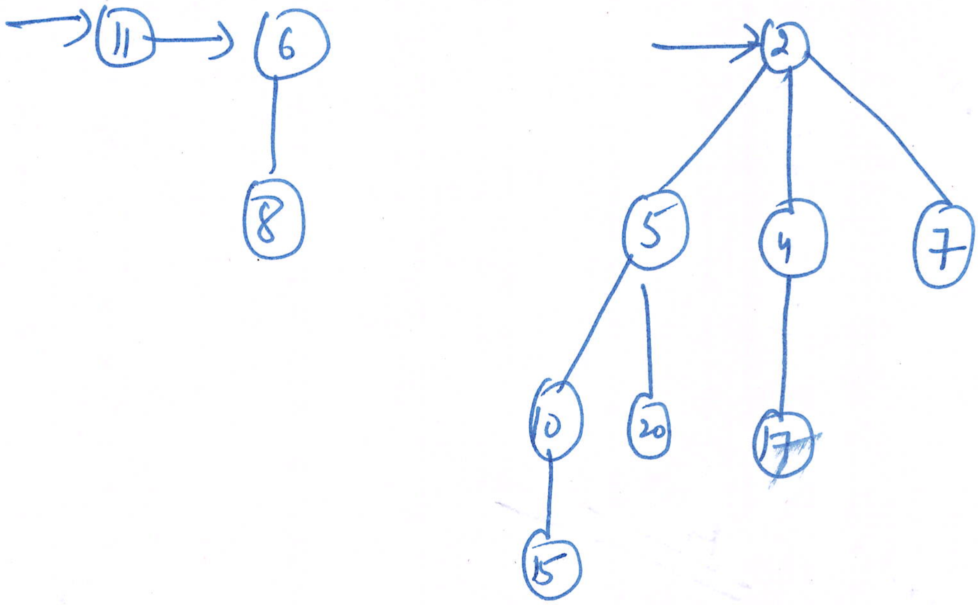




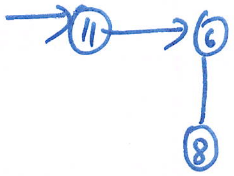
Apply union



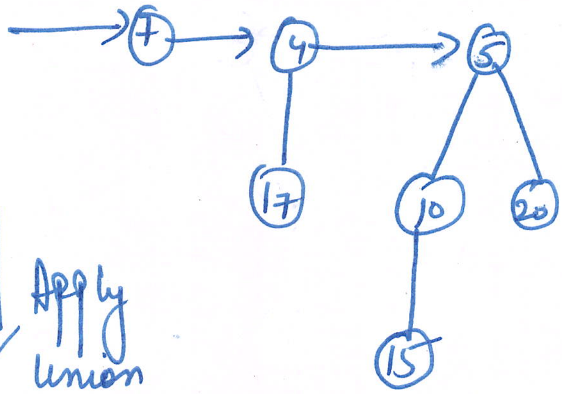
(5) Again Apply Extraction of Minimum key



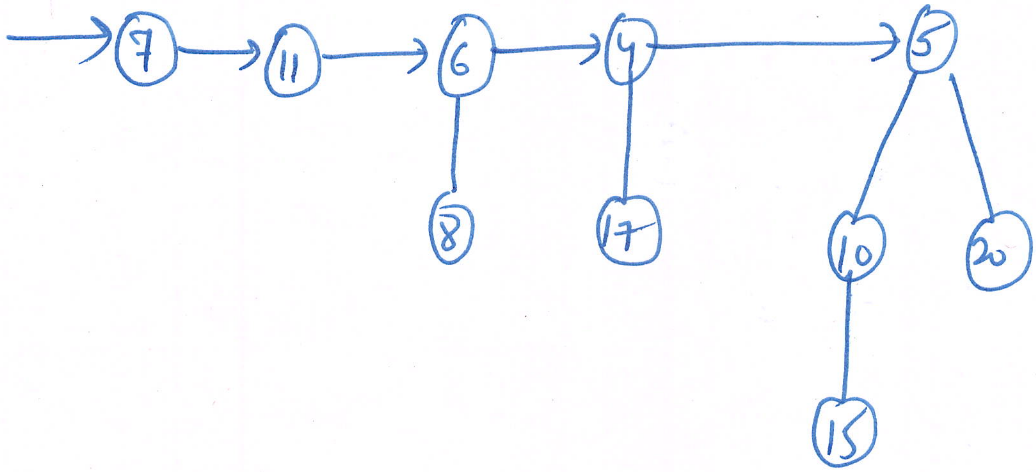
1

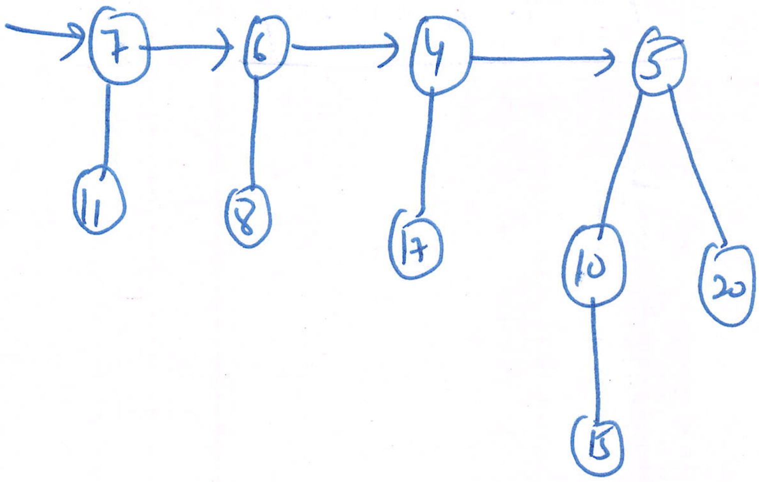


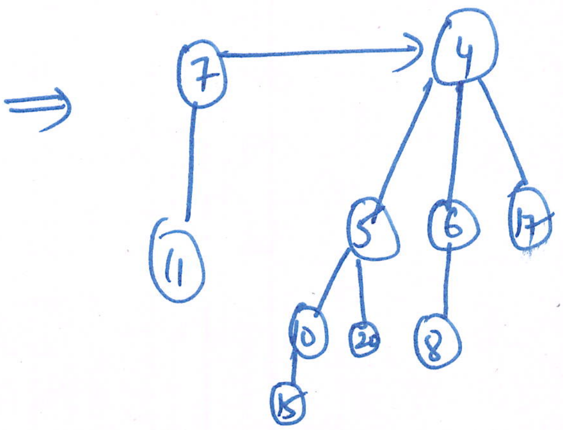
2



Apply union





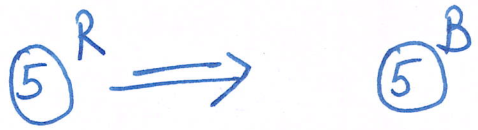


Ans

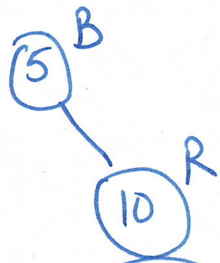
$2\frac{1}{2}$ Marks

8 (a) Show the Red Black tree that result after inserting keys 5, 10, 15, 20, 25, 30 and 35 into an initially empty Red Black tree

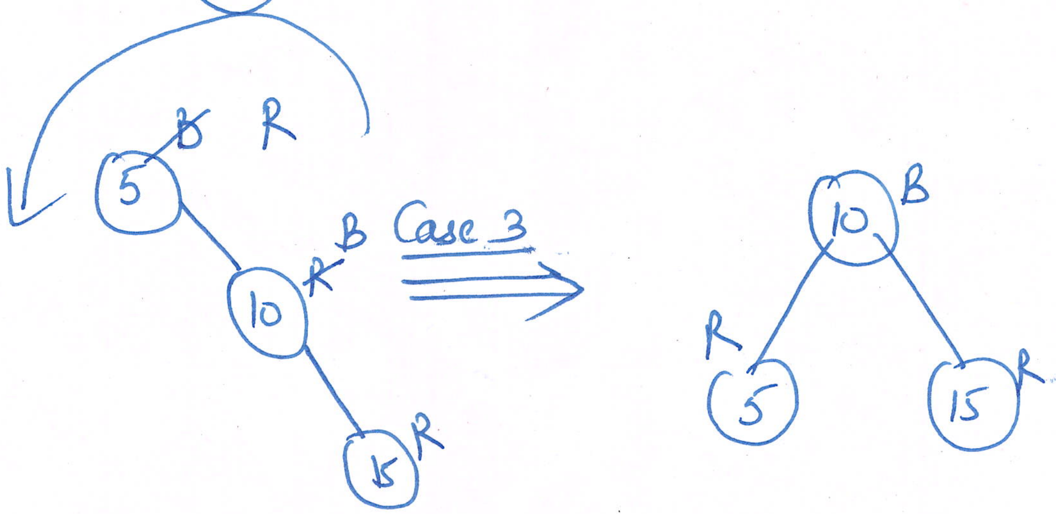
Ans. Insert 5



Insert → 10

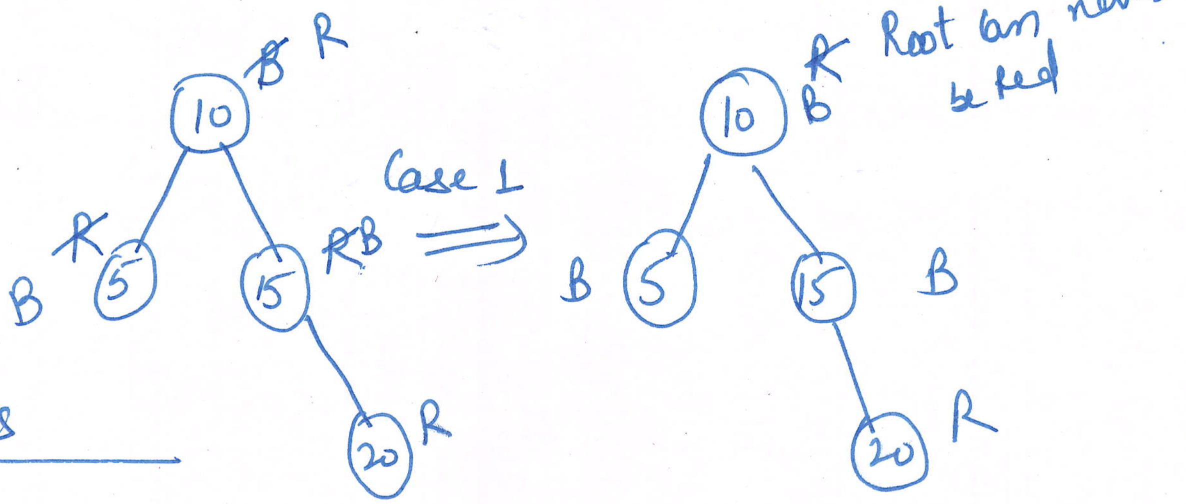


Insert 15



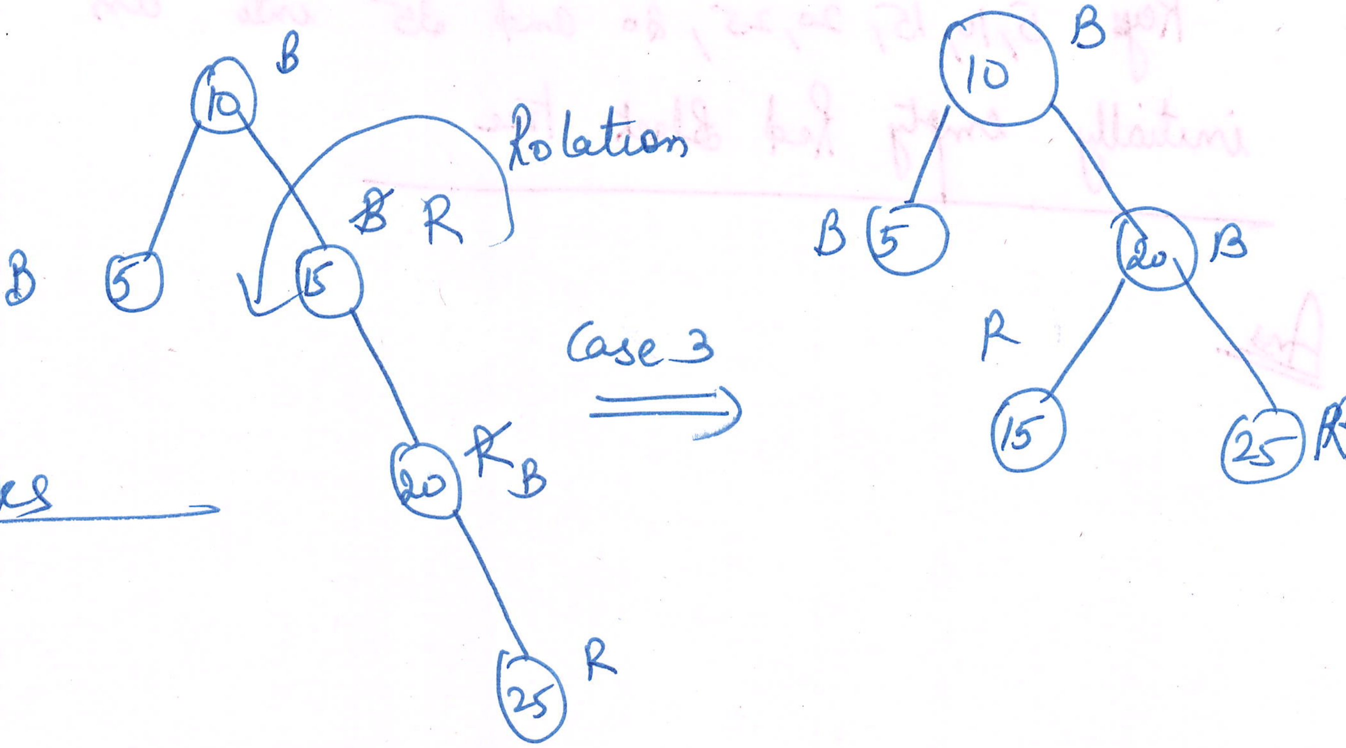
2 Marks

Insert 20



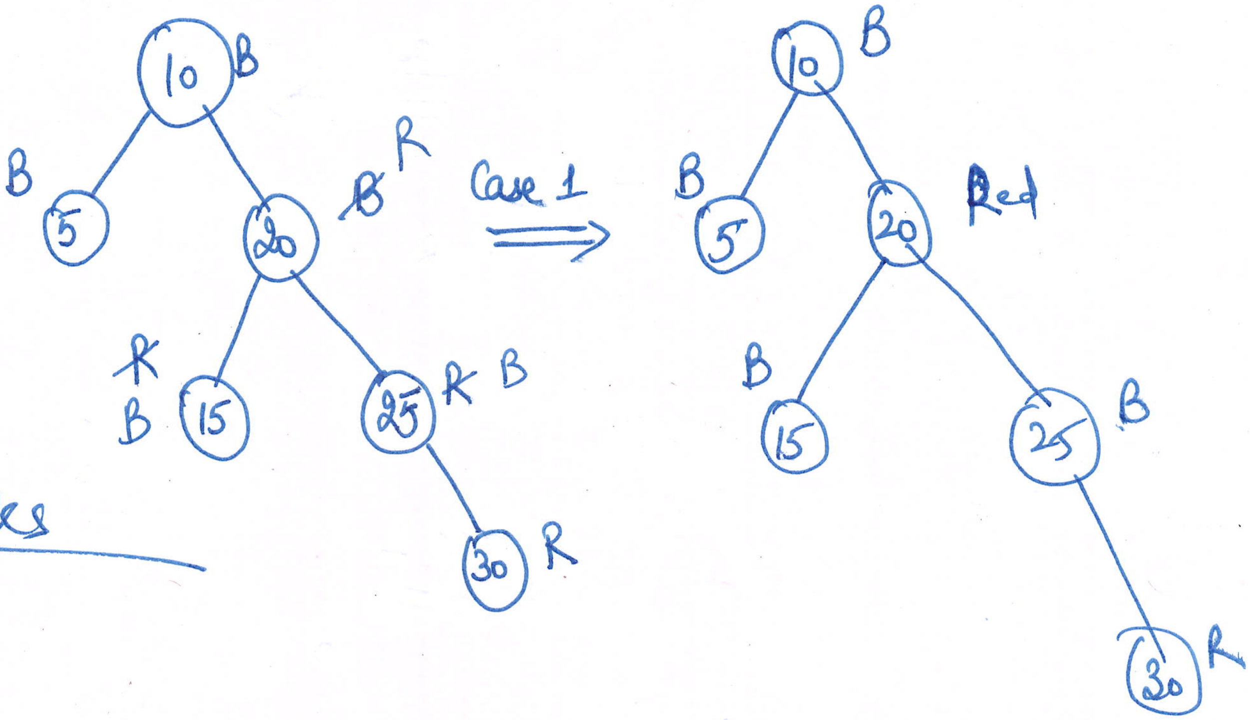
2 Marks

Insert-25



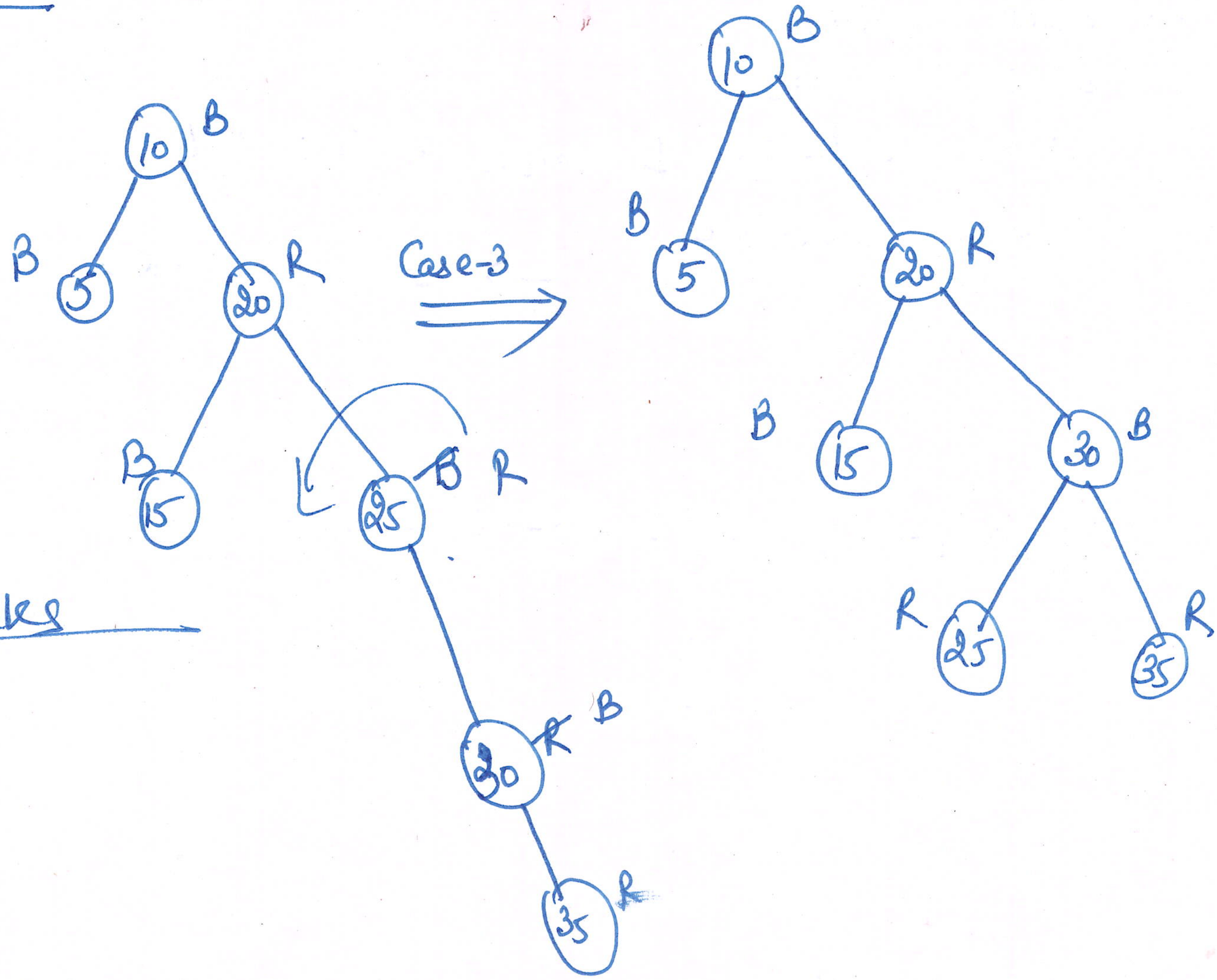
2 Marks

Insert 30



2 Marks

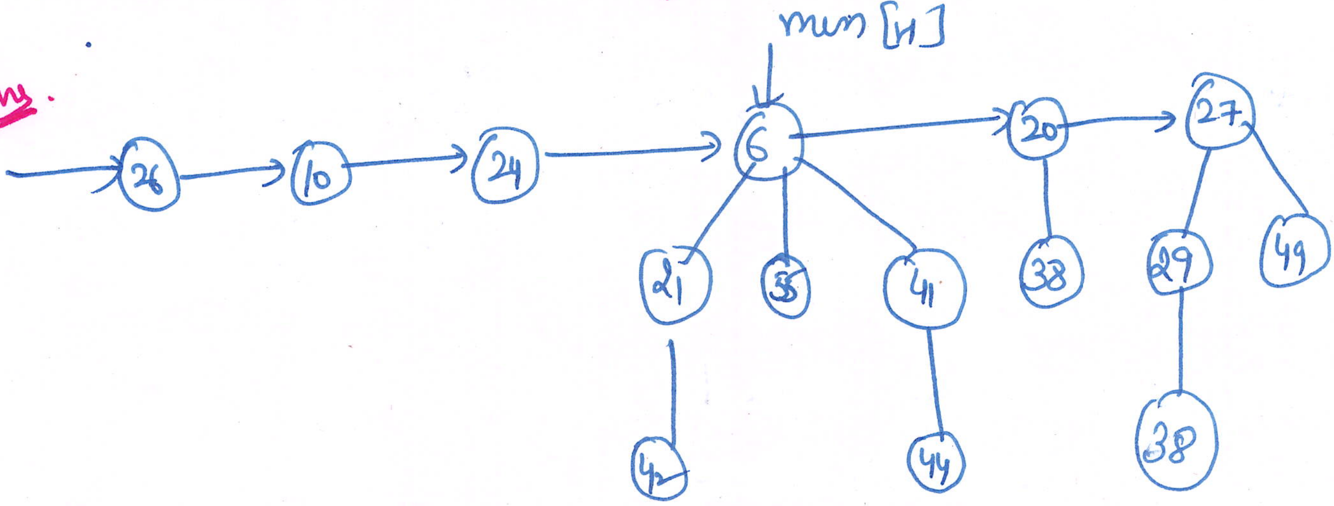
Insert -35



2 Marks

Ques 8(b) Discuss Procedure for Extracting minimum node from Fibonacci Heap. Also write Fibonacci Heap Extract Min & Consolidate Algorithms -

Ans.



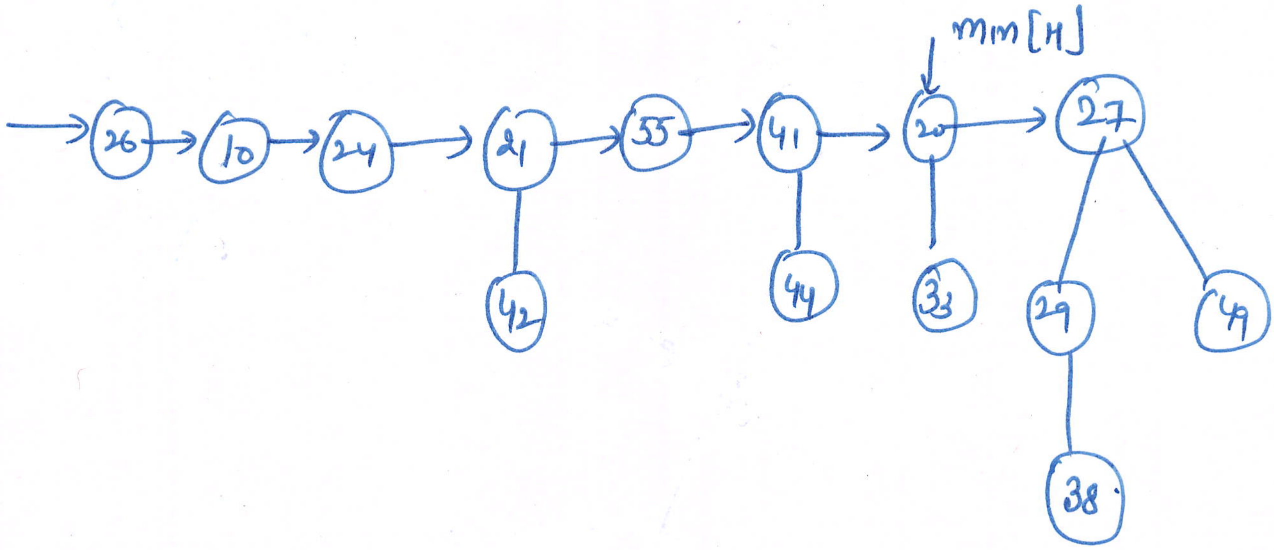
→ Extract min[H].

→ set z = min[H]

→ Move all children of min[H] to root list

→ Perform Consolidate

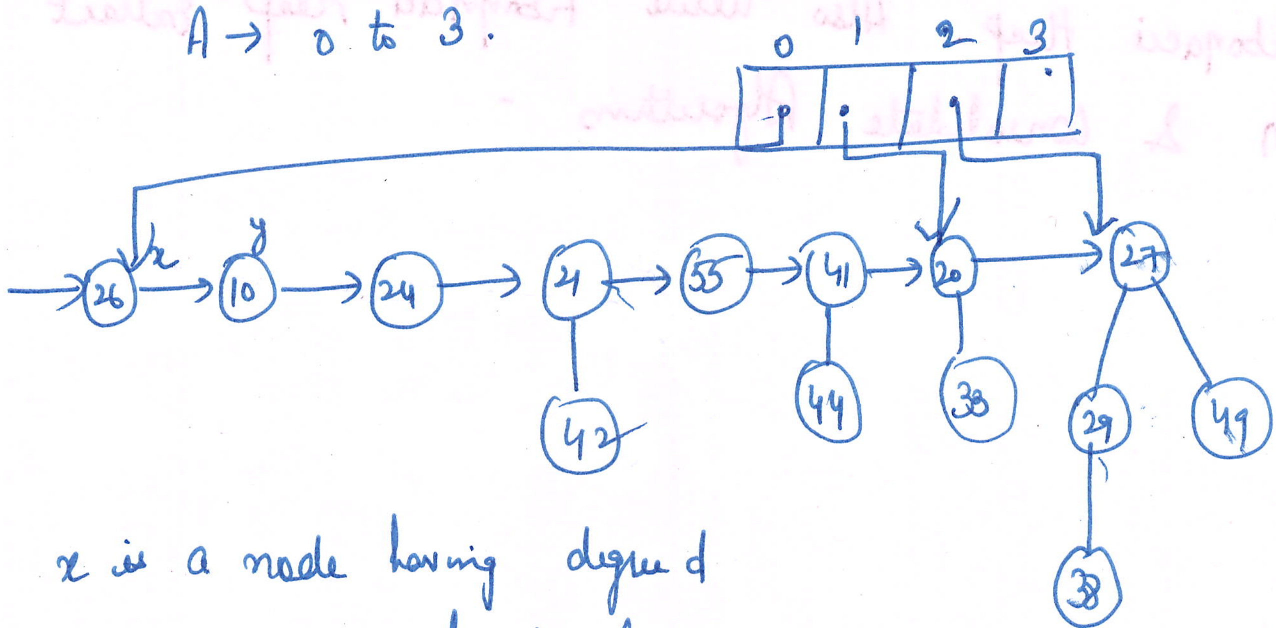
↳ 2½ Marks



Create an array

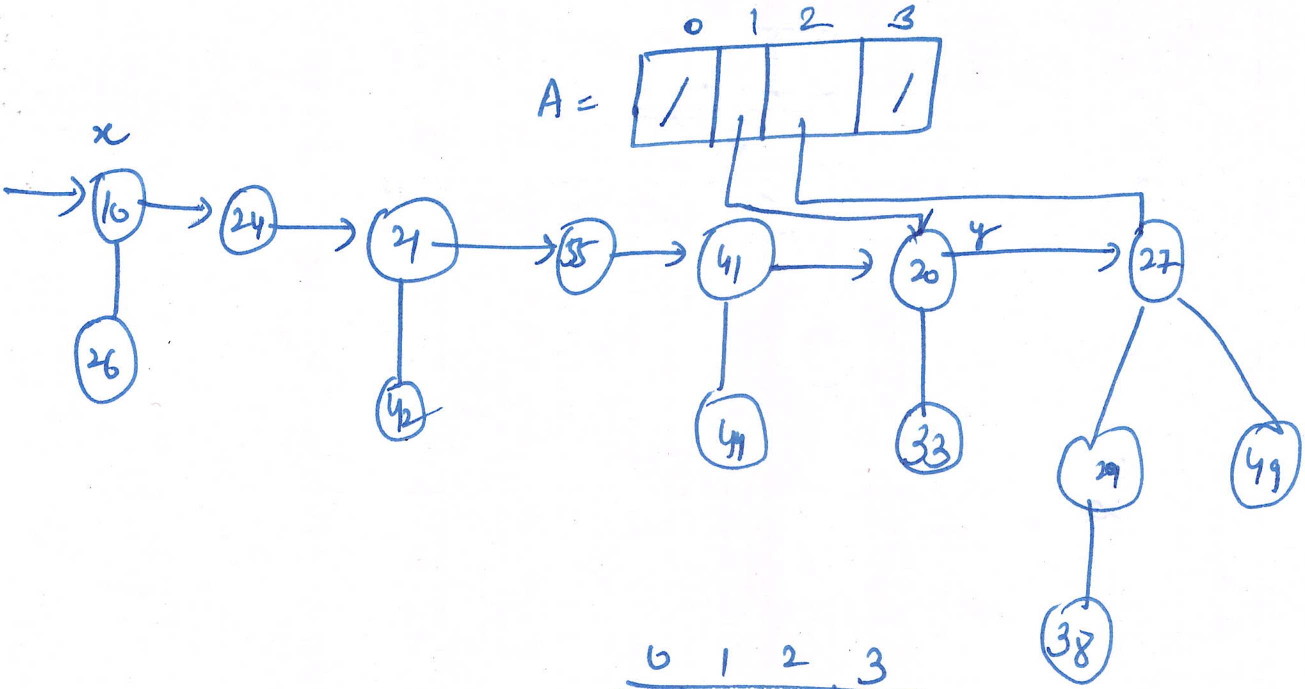
$A = 0$ to $\lfloor \log_2 n \rfloor$

$A \rightarrow 0$ to 3.

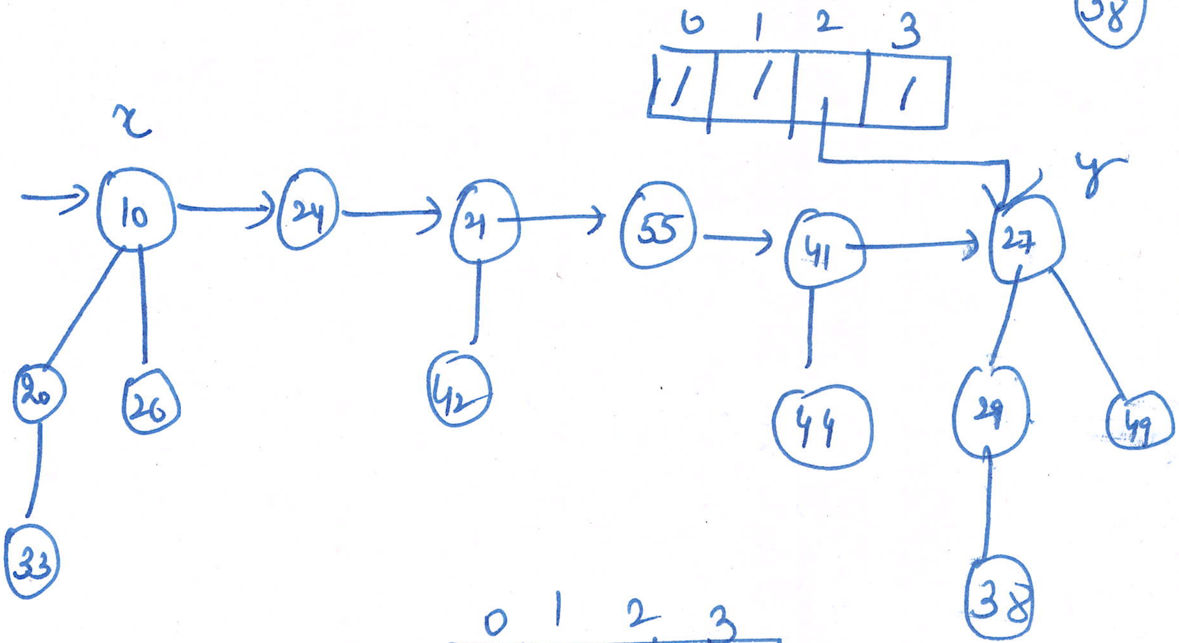


x is a node having degree d

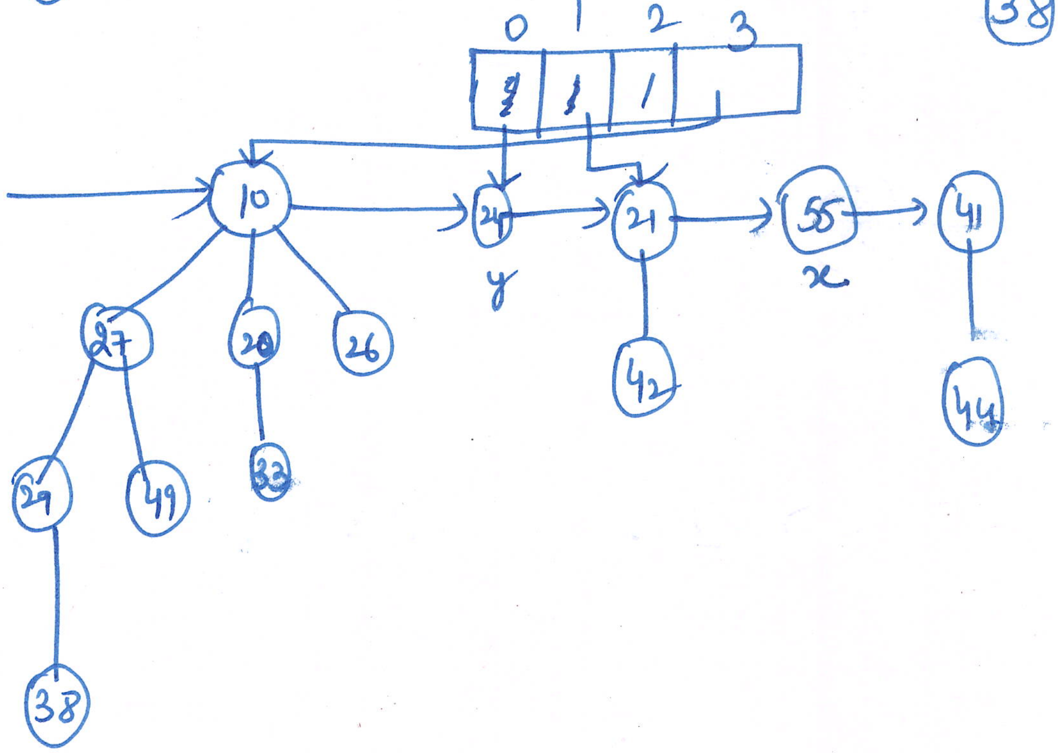
y is another node having same degree as of x.



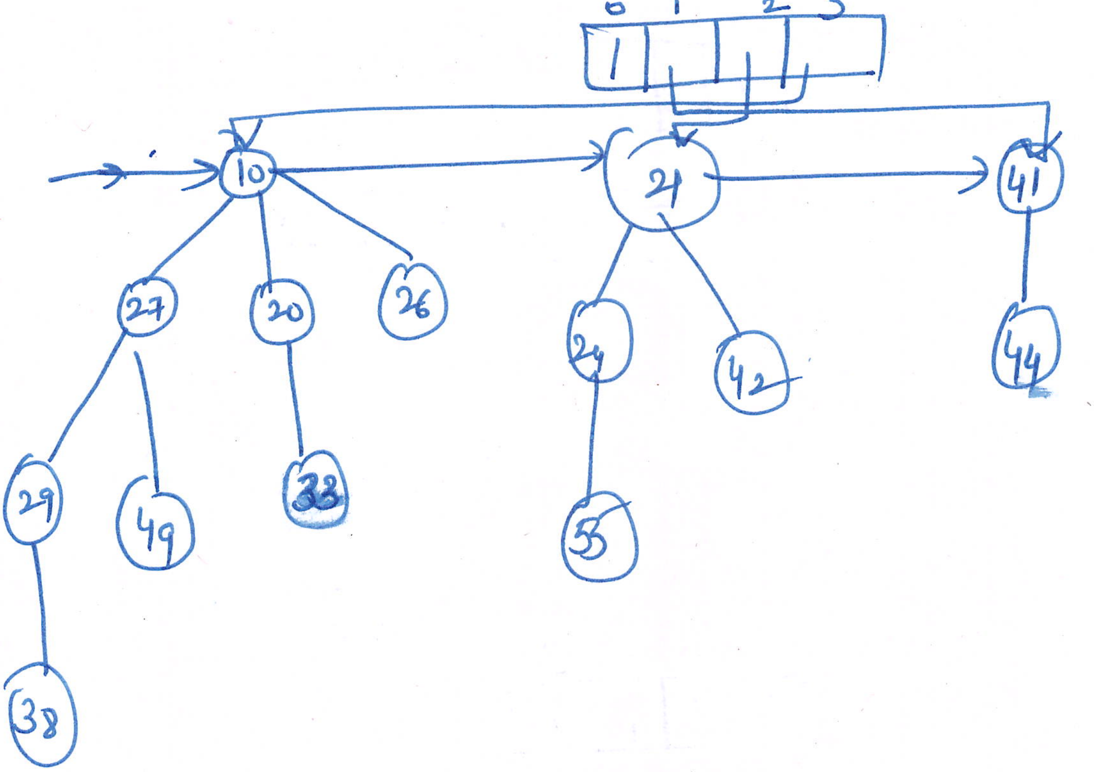
0 1 2 3
A =
x
y
10
24
21
55
41
20
27
26
42
44
33
29
49
38

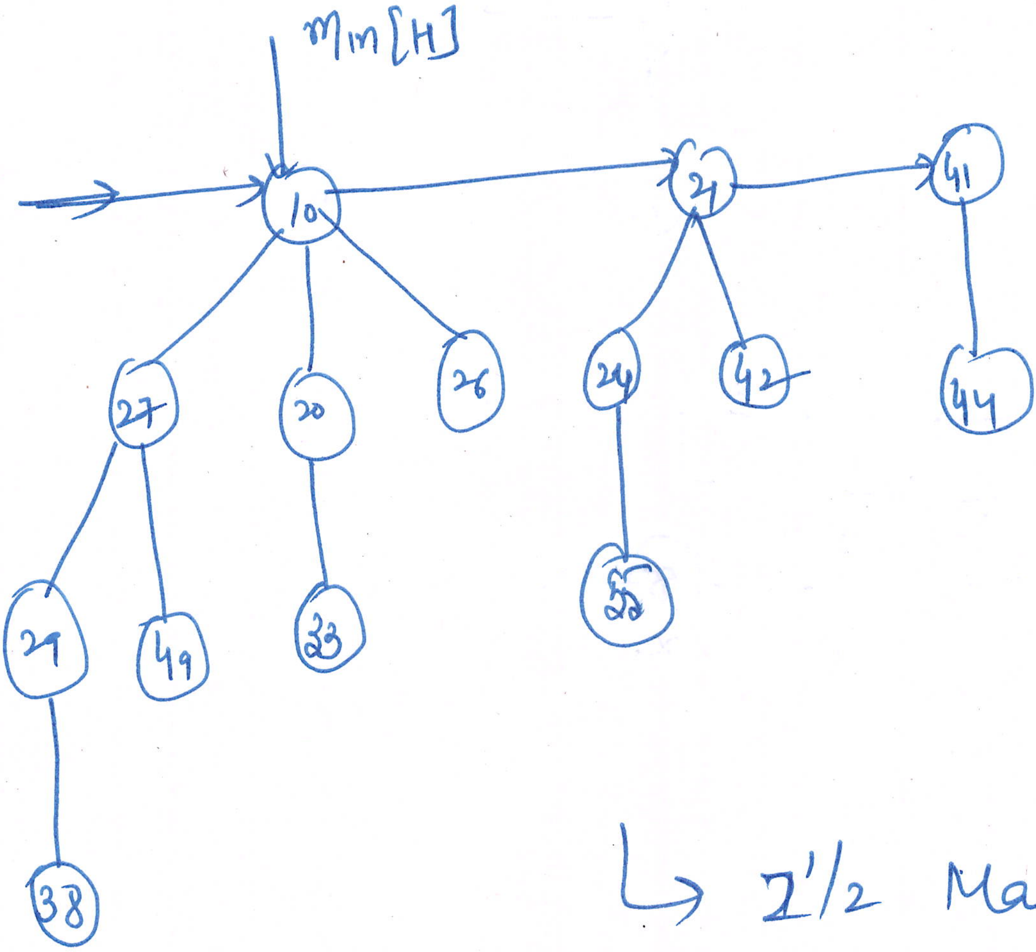


0 1 2 3
x
y
10
24
21
55
41
27
20
26
42
44
29
49
33
38



0 1 2 3
10
24
21
55
41
y
x
27
20
26
42
44
29
49
33
38


0 1 2 3
1
10
21
41
27
20
26
24
42
44
29
49
33
55
38


min[H]
10
21
41
27
20
26
24
42
44
29
49
23
55
38

↳ 2½ Marks

Fib_Heap_Extract_Min(H)

```
z ← min[H]
if z ≠ NIL
   then for each child x of z
           do add x to the root list of H
              p[x] ← NIL
        remove z from root list of H
        if z = right[z]
           then min[H] ← NIL
        else min[H] ← right[z]
             CONSOLIDATE(H)
        n[H] ← n[H] − 1
return z.
```

CONSOLIDATE(H)

```
for i ← 0 to ⌊log₂ n⌋
   do A[i] ← NIL
for each node w in root list of H
   do x ← w
```

```
d ← degree[x]
while A[d] ≠ NIL
do y ← A[d]
if key [x] > key [y]
then exchange x ↔ y
Fib-Heap-Link (H, y, x)
A[d] ← NIL
d ← d + 1
A[d] ← x
min[H] ← NIL
for i ← 0 to ⌊log2 n⌋
do if A[i] ≠ NIL
then add A[i] to root list of H
if min[H] = NIL or key [A[i]] < key [min [H]]
then min[H] ← A [i]
```

Fib-Heap-Link (H, y, x).

1. remove y from root list of H
2. make y a child of x, incrementing degree[x]
3. mark[y] ← false.

↳ 5 Marks

Qus (9) (a)

Show all steps of Strassen's Matrix Multiplication

$$X = \begin{array}{cc} A_{11} & A_{12} \\ \left[\begin{matrix} 2 & 5 \\ 4 & 3 \end{matrix}\right] & \\ A_{21} & A_{22} \end{array} \qquad Y = \begin{array}{cc} B_{11} & B_{12} \\ \left[\begin{matrix} 2 & 6 \\ 5 & 4 \end{matrix}\right] & \\ B_{21} & B_{22} \end{array}$$

$P = (A_{11} + A_{22})(B_{11} + B_{22}) \Rightarrow (2+3)(2+4) = 30$

$Q = (A_{21} + A_{22})\, B_{11} \Rightarrow (4+3) \cdot 2 = 14$

$R = A_{11}(B_{12} - B_{22}) \Rightarrow 2(6-4) \Rightarrow 4$

$S = A_{22}(B_{21} - B_{11}) \Rightarrow 3(5-2) \Rightarrow 9$

$T = (A_{12} + A_{11})\, B_{22} \Rightarrow (5+2)4 \Rightarrow 28$

$U = (A_{21} - A_{11})(B_{11} + B_{12}) \Rightarrow (4-2)(2+6) \Rightarrow 16$

$V = (A_{12} - A_{22})(B_{22} + B_{21}) \Rightarrow (5-3)(4+5) \Rightarrow 18$ → 5 Marks

$C_{11} = P + S - T + V = 30 + 9 - 28 + 18 = 29$

$C_{12} = R + T = 4 + 28 = 32$

$C_{21} = Q + S = 14 + 9 = 23$

$C_{22} = P + R - Q + U = 30 + 4 - 14 + 16 = 36$

→ 5 Marks

Qu 9(b) Compute 2101 * 1130 by applying divide & Conquer.

Ans

2101 * 1130

$C_2 = 21 * 11$

$C_0 = 01 * 30$

$C_1 = (21 + 01) * (11 + 30) - (C_2 + C_0)$

$= 22 * 41 - 21 * 11 - 01 * 30$ ⟶ 2½ Marks

For 21 * 11

$C_2 = 2 * 1 = 2$

$C_0 = 1 * 1 = 1$

$C_1 = (2+1) * (1+1) - (2+1)$

$= 3 * 2 - 3 = 3$

So $21 * 11 = 2 * 10^2 + 3 * 10^1 + 1 = 231$

⟶ 2½ Marks

For 01 * 30

$C_2 = 0 * 3 = 0$

$C_0 = 1 * 0 = 0$

$C_1 = (0+1) * (3+0) - (0+0)$

$= 1 * 3 - 0 = 3$

So $01 * 30 = 0 * 10^2 + 3 * 10^1 + 0 = 30$

⟶ 2½ Marks

For $22 * 41$

$C_2 = 2 * 4 = 8$

$C_0 = 2 * 1 = 2$

$C_1 = (2+2) * (4+1) - (8+2) = 4*5-10 = 10$

So $22 * 41 = 8 \cdot 10^2 + 10 \cdot 10^1 + 2 = 902$

Hence

$2101 * 1130$

$= 231 * 10^4 + (902-231-30) \cdot 10^2 + 30$

$\boxed{= 2374130}$

$\longrightarrow$ $2\frac{1}{2}$ Marks