Practice - Sheet - 2

Q.3 Find the average no. of successful and unsuccessful search. Also find internal and external path length. Consider the array given below to perform Binary search. 15, -8, 0, 7, 10, 20, 40, 80, 92, 101, 110, 121, 138, 145, 160.

⇒ Step ① :- Sort in ascending order

-8, 0, 7, 10, 15, 20, 40, 80, 92, 101, 110, 121, 138, 145, 160

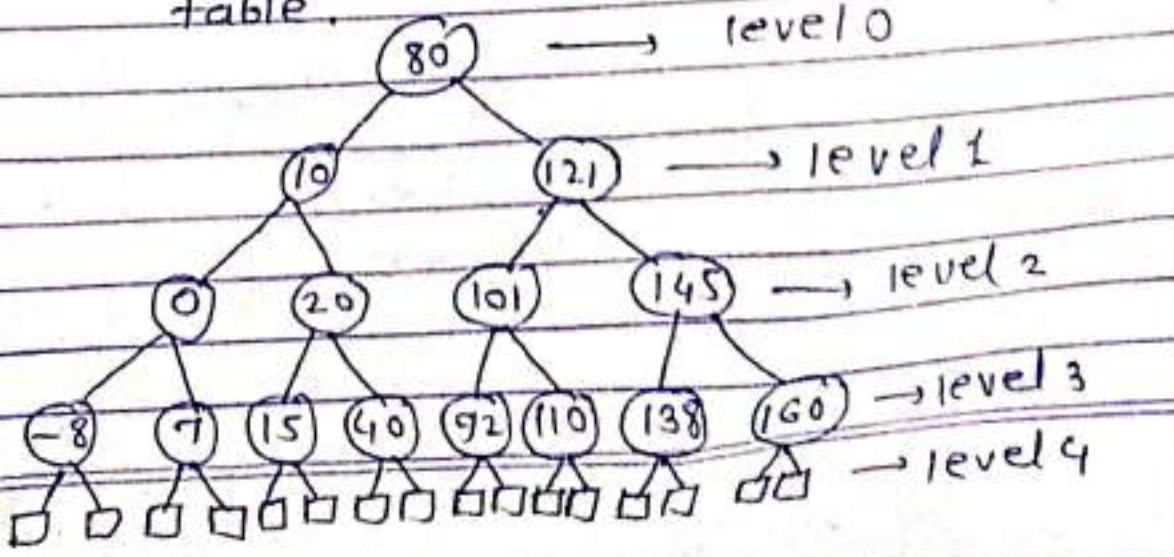
Step ② :- draw the comparision table

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
element	-8	0	7	10	15	20	40	80	92	101	110	121	138	145	160
comparison	④	③	④	②	④	③	④	①	④	③	④	②	④	③	④

$mid = \frac{1+15}{2} = \frac{16}{2} = 8$,

$mid = \frac{1+7}{2} = 4$, $mid = \frac{9+15}{2} = 12$, $mid = \frac{1+3}{2} = 2$

Step ③ :- make tree based on the comparison table.



Step ④ :- calculate IPL & EPL

IPL = node * level
= (1x0) + (2x1) + (4x2) + (8x3)
= 2 + 8 + 24
IPL = 34

EPL = dummy * level
= (16x4)
EPL = 64

Step ⑤ :-

Average number of comparison of successful search = $\frac{IPL}{n} + 1$

$= \frac{34}{15} + 1$

$= \frac{49}{15}$

$= 3.26$

Average number of comparison of unsuccessful search = $\frac{EPL}{n+1}$

$= \frac{64}{16}$

$= 4$

Q.4 Explain Analysis of binary serach for following elements and calculate Avg. no. of successful & Avg. no. of unsuccessful comparisons -10, 25, 15, 16, 18, -9, 4, 2.

⇒ Step ① :- sort in ascending order

-10, -9, 2, 4, 15, 16, 18, 25

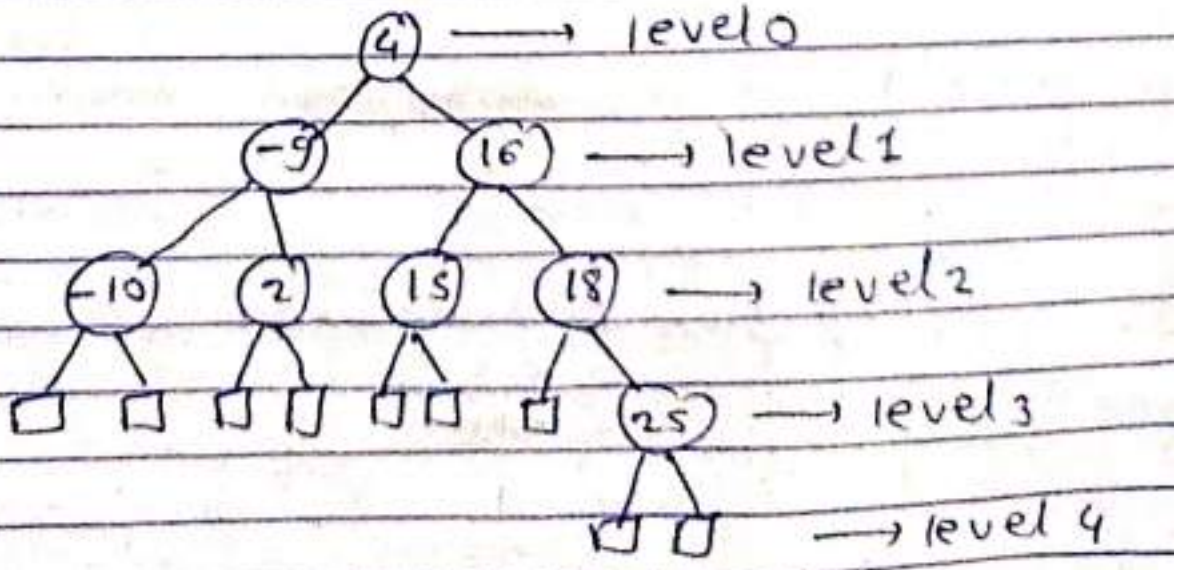
Step ② :- Draw comparison table

Index	1	2	3	4	5	6	7	8
element	-10	-9	2	4	15	16	18	25
comparison	③	②	③	①	③	②	③	④

$mid = \frac{1+8}{2} = 4$

$mid = \frac{1+3}{2} = 2$, $mid = \frac{5+8}{2} = \frac{13}{2} = 6$

Step ③ :- make tree based on comparison table



Step ④ :- calculate EPL of IPL

IPL = node * level
= (1×0) + (2×1) + (4×2) + (1×3) + (0×4)
= 0 + 2 + 8 + 3
IPL = 13

EPL = dummy * level
= (7×3) + (2×4)
= 21 + 8
EPL = 29

Step ⑤ :-

Avg. no. of comparision of successful search
$= \frac{IPL}{n} + 1$
$= \frac{13}{8} + 1$
$= \frac{21}{8}$
$= 2.62$

Avg. no. of comparison of unsuccessful sear
$= \frac{EPL}{n+1}$
$= \frac{29}{9}$
$= 3.22$

Q.5 Explain Analysis of binary search for following elements and calculate Avg. no. of successful and avg. no. of unsuccessful comparisons. -12, 23, 31, 45, 56, 78, 90, 103, 113, 126, 157.

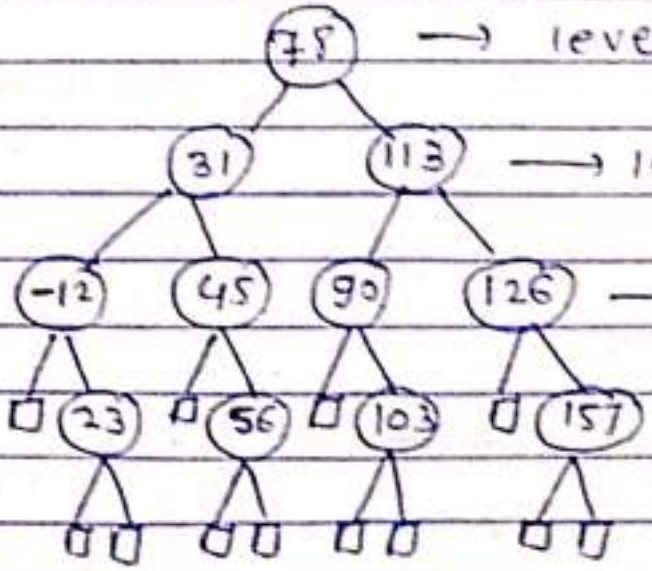
⇒ step (1) :- sort in ascending order

-12, 23, 31, 45, 56, 78, 90, 103, 113, 126, 157

step (2) :- Draw comparison table

Index	1	2	3	4	5	6	7	8	9	10	11
element	-12	23	31	45	56	78	90	103	113	126	157
comparison	(3)	(4)	(2)	(3)	(4)	(1)	(3)	(4)	(2)	(3)	(4)

step (3) :- make tree based on comparison table



step (4) :- calculate EPL & IPL

IPL = node * level

= (1×0) + (2×1) + (4×2) + (4×3) +

= 2 + 8 + 12

IPL = 22

EPL = dummy * level

= (9 × 3) + (8 × 4)

= 12 + 32

EPL = 44

Step (5) :-

Avg. no. of comparison of successful search = $\frac{IPL}{n} + 1$

$= \frac{22}{11} + 1$

$= 3$

Avg. no. of comparition of unsecessful search = $\frac{EPL}{n+1}$

$= \frac{44}{12}$

$= 3.66$

Q.6 Find the average no. of successful & unsuccessful search. Also find internal & external path length. consider the array given below to perform binary search : a = {-15, -6, 0, 7, 9, 23, 54, 82, 101, 112, 125, 131, 142, 151} Also give the complexity of binary search.

⟹ step (1) :- sort the element in ascending order.

-15, -6, 0, 7, 9, 23, 54, 82, 101, 112, 125, 131, 142, 151

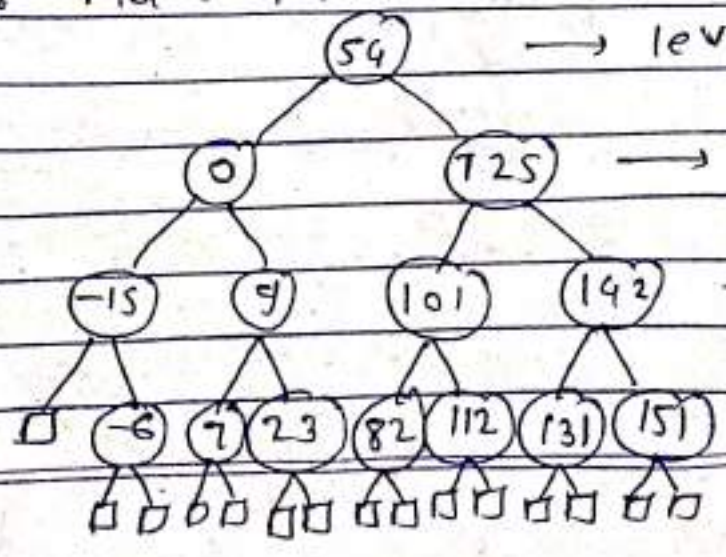
step (2) :- Draw comparison table.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
element	-15	-6	0	7	9	23	54	82	101	112	125	131	142	151
comparison	(3)	(4)	(2)	(4)	(3)	(4)	(1)	(4)	(3)	(4)	(2)	(4)	(3)	(4)

$$mid = \frac{1+14}{2} = \frac{15}{2} = 7$$

$$mid = \frac{1+6}{2} = 3, \quad mid = \frac{8+14}{2} = \frac{22}{2} = 11,$$

step (3) :- make tree based on comparison table.



Step (4) :- calculate EPL & IPL

IPL = node * level
= (1x0) + (2x1) + (4x2) + (7x3)
= 2 + 8 + 21
IPL = 31

EPL = dummy * level
= (1x3) + (14x4)
= 3 + 56
EPL = 59

Step (5) :-

Avg. no. of comparison of successful search
$= \frac{IPL}{n} + 1$
$= \frac{31}{14} + 1$
$= 3.21$

Avg. no. of comparision of unsuccessful search $= \frac{EPL}{n+1}$
$= \frac{59}{15}$
$= 3.93$

Q.7 Explain the complexity of binary search, Implement binary search on following array and find avg no. of comparison required for successful and unsuccessful search - -12, -4, 9, 32, 50, 79, 109, 135, 203, 230.

→ step ① :- sort in ascending order.

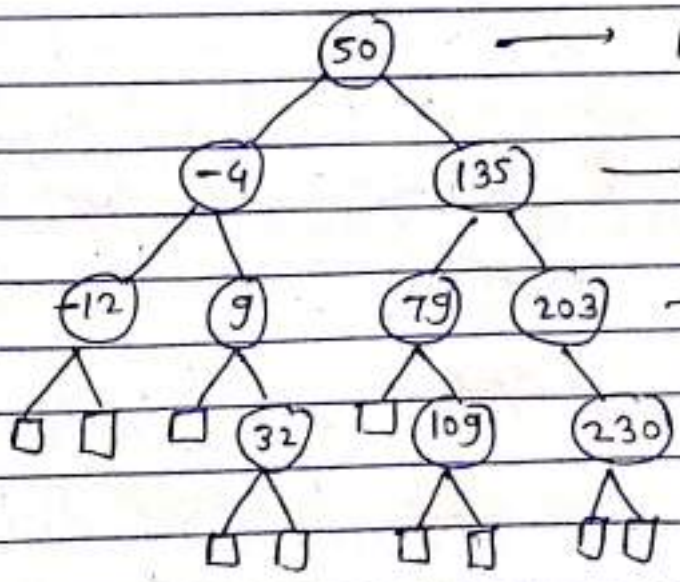
-12, -4, 9, 32, 50, 79, 109, 135, 203, 230

step ② :- Draw the comparison table

Index	1	2	3	4	5	6	7	8	9	10
element	-12	-4	9	32	50	79	109	135	203	230
comparison	(3)	(2)	(3)	(4)	(1)	(3)	(4)	(2)	(3)	(4)

$mid = \frac{1+10}{2} = \frac{11}{2} = 5$, $mid = \frac{1+4}{2} = 2$, $mid = \frac{6+10}{2} = 8$

step ③ :- make tree depending on comparison table



Step ⑦ :- calculate EPL & IPL

IPL = node * level
= (1x0) + (2x1) + (4x2) + (3x3)
= 2 + 8 + 9
IPL = 19

EPL = dummy * level
= (4x3) + (6x4)
= 12 + 24
EPL = 36

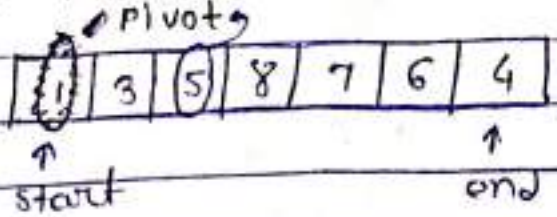
Step ⑧ :-

Avg. no. of comparison of successful search
$= \frac{IPL}{n} + 1$
$= \frac{19}{10} + 1$
$= \underline{\underline{2.9}}$

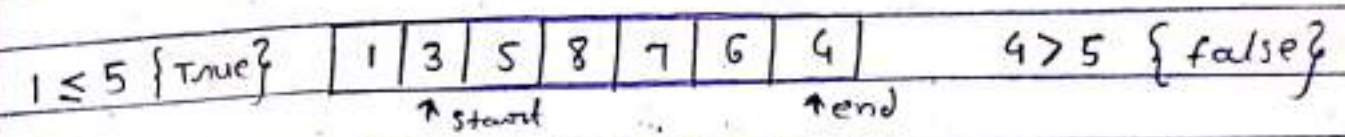
Avg. no. of comparison of unsuccessful search
$= \frac{EPL}{n+1}$
$= \frac{36}{11}$
$= \underline{\underline{3.27}}$

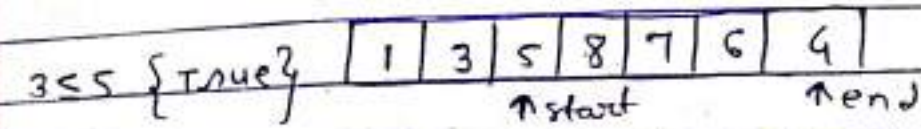
Q.8 Illustrate stepwise execution of Quick sort on the following array A. Also give its complexity by analysis the recurrence relation A = {1, 3, 5, 8, 7, 6, 4}

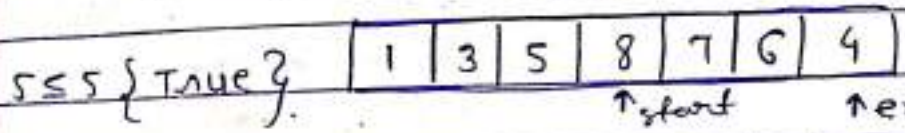
⇒

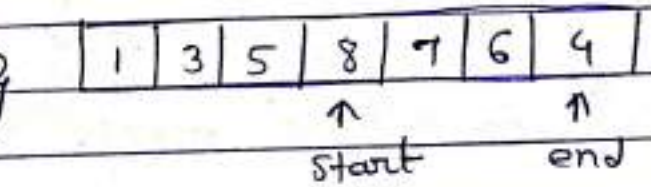


A[start] ≤ Pivot ⟶ increment start pointer
A[end] > Pivot ⟶ decrement end pointer



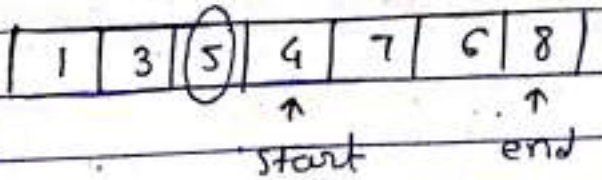


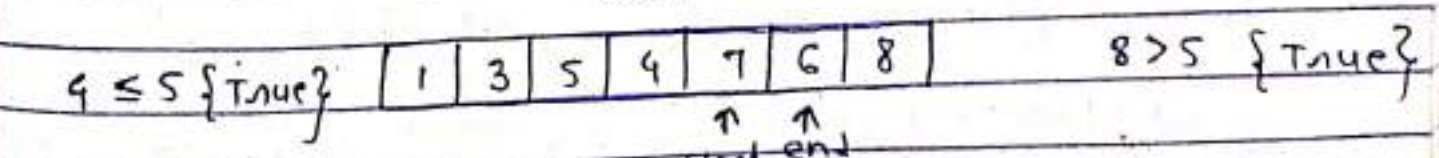


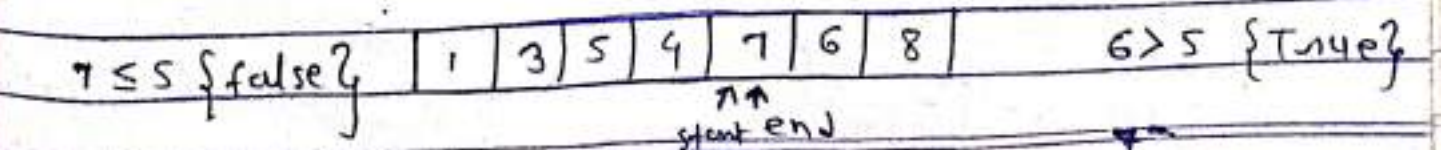


Both of the pointer are stuck then swap

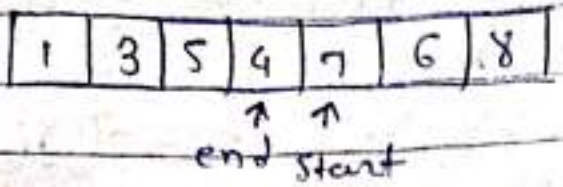
swap(A[start], A[end])



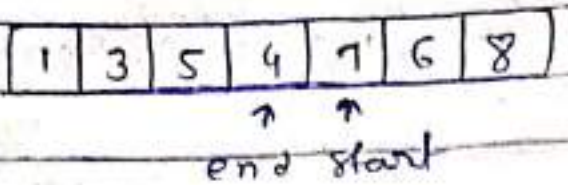




7 ≤ 5 {false}



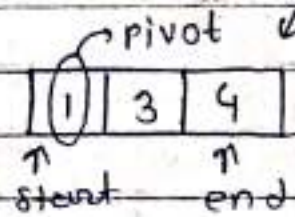
7 > 5 {True}

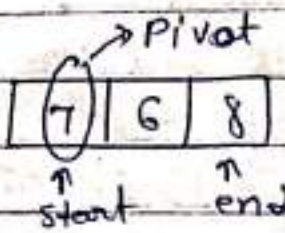


4 > 5 {false}

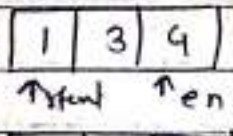
Both of the pointer are stuck but here end pointer pointing before start pointer then swap (Pivot, A[end])

Now,



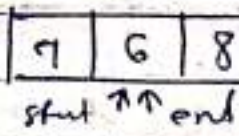


1 ≤ 1 {True}



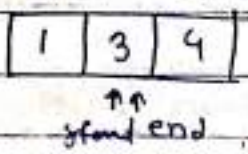
4 > 1 {True}

7 ≤ 7 {True}



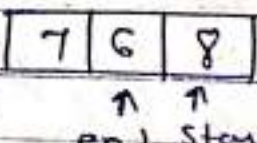
8 > 7 {True}

3 ≤ 1 {false}

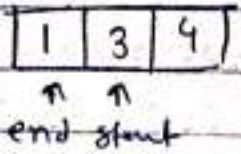


3 > 1 {True}

6 ≤ 7 {True}

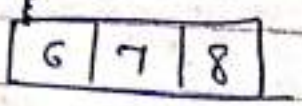


6 > 7 {false}

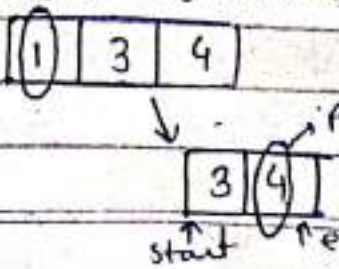


1 > 1 {false}

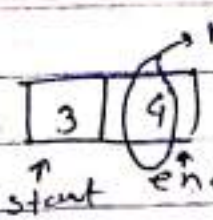
swap (A[end], Pivot)



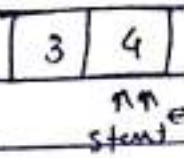
swap (A[end], Pivot)



PAGE NO.
DATE

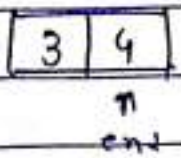


$3 \leq 4$ {True}



$4 > 4$ {false}

$4 \leq 4$ {True}



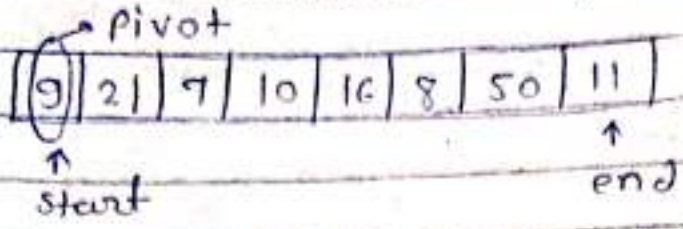
combine all of them

A	1	3	4	5	6	7	8

sorted element = {1, 3, 4, 5, 6, 7, 8}

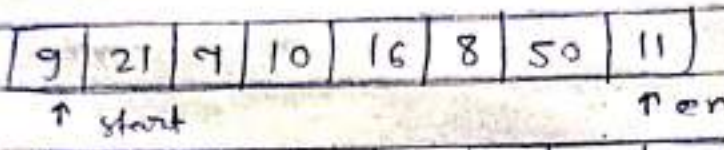
Time complexity = $O(n \log n)$

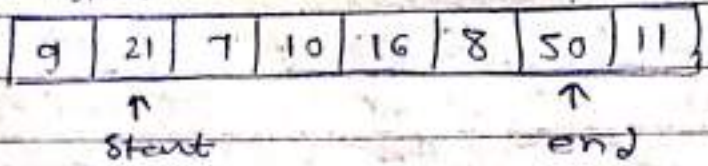
Q.9 sort the given array using Quick sort algorithm : a = { 9 21 7 10 16 8 50 11 } also explain the time complexity of quick sort

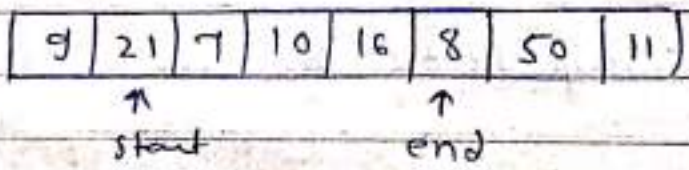


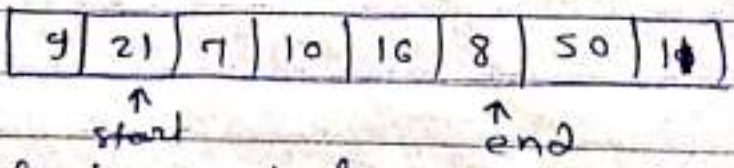
A[start] ≤ pivot → increment start pointer

A[end] > pivot → decrement end pointer



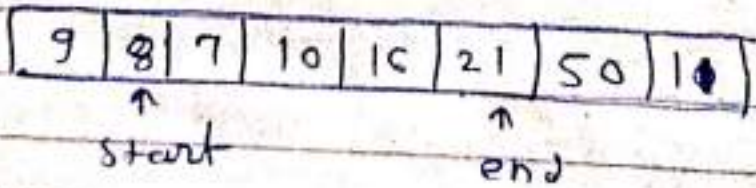
9 ≤ 9 {True}  11 > 9 {True}

21 ≤ 9 {false}  50 > 9 {True}

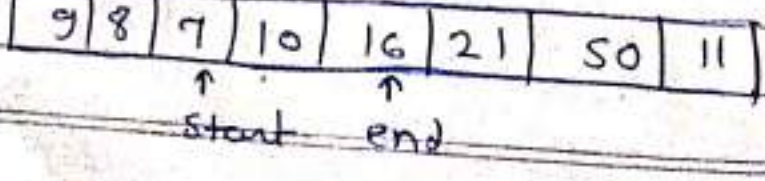


8 > 9 {false}

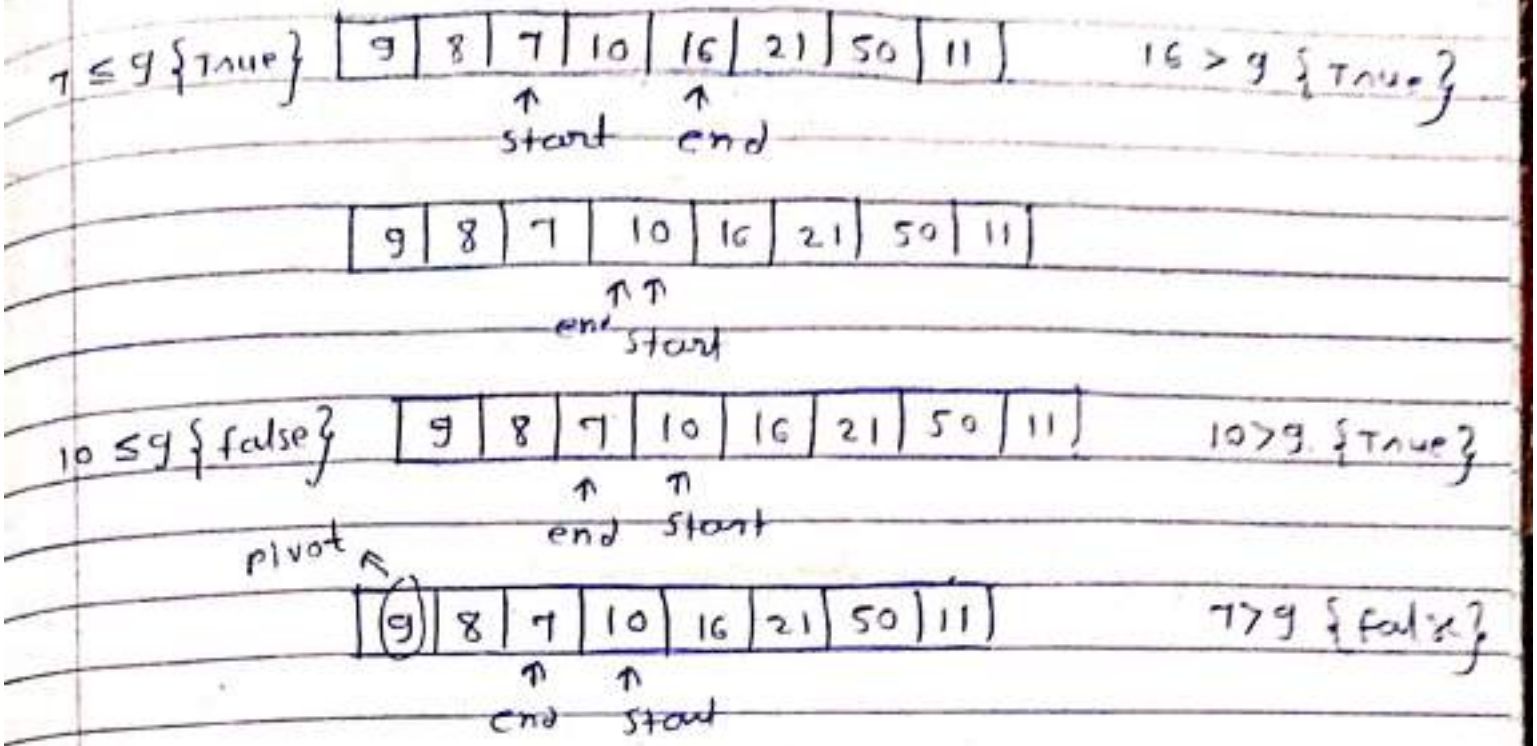
Both of the pointers are stuck then swap (A[start], A[end])



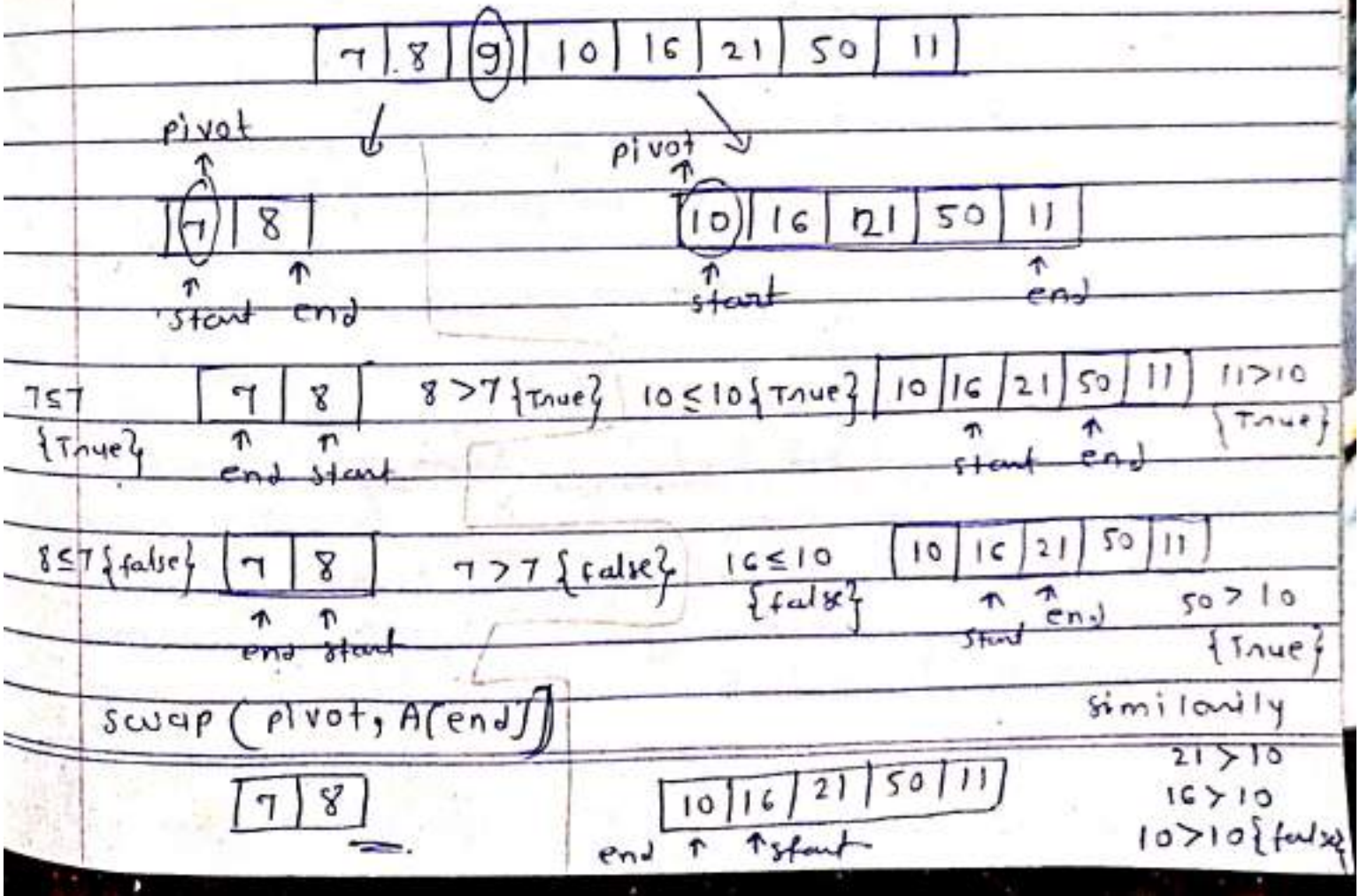
8 ≤ 9 {True}

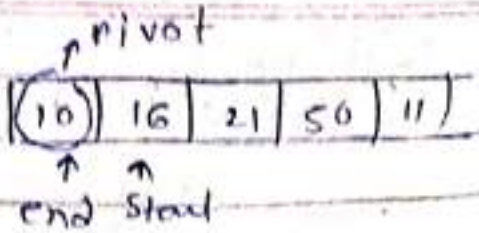


21 > 9 {True}

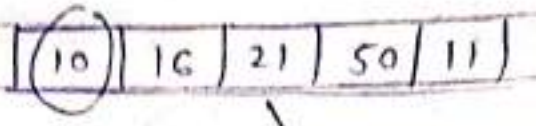


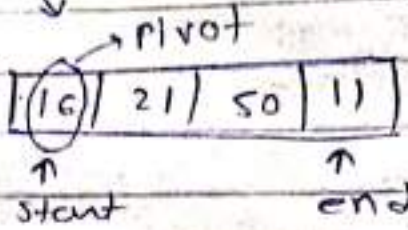
Both of the pointes are stuck but here end pointes pointing before start pointer then swap (pivot, A[end])



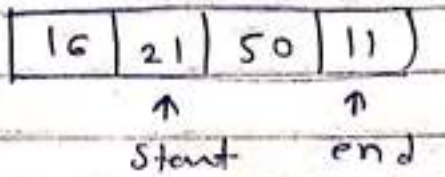


swap (pivot , A[end])



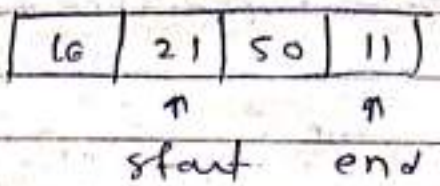


16 ≤ 16
{True}

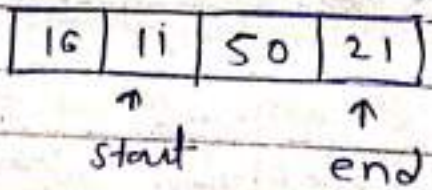


11 > 16
{false}

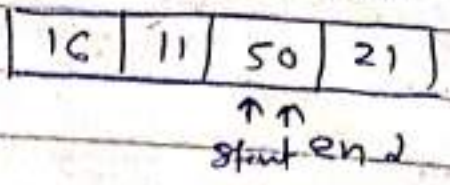
21 ≤ 16
{false}



Both pointer one stuck
swap (A[start], A[end])

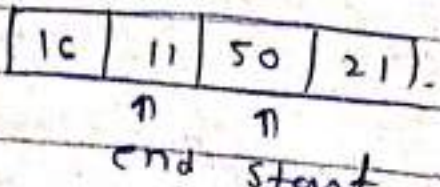


11 ≤ 16
{True}

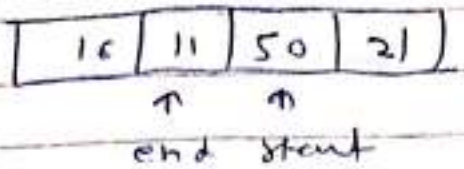


21 > 16
{True}

50 ≤ 16
{false}

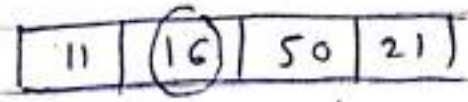


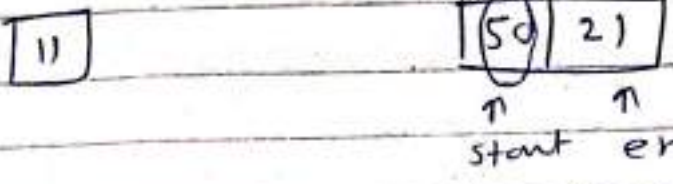
50 > 16
{True}



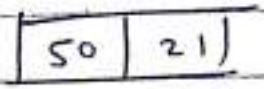
11 > 16 {false}

swap (pivot , A [end])



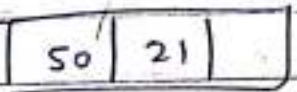


50 ≤ 50 {True}

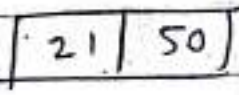


21 > 50 {false}

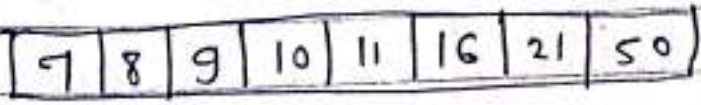
21 ≤ 50 {True}



swap (end , pivot)



combine all



Qn sorted - element = { 7, 8, 9, 10, 11, 16, 21, 50}

Time complexity = $O(n \log n)$

explanation

Q.12 Find the multiplication of the following matrixes A & B given below using strassen's matrix multiplication algorithm.

$A = \begin{bmatrix} 4 & 5 \\ 7 & 2 \end{bmatrix}$ & $B = \begin{bmatrix} 2 & 3 \\ 5 & 9 \end{bmatrix}$

write the complexity of strassen's matrix multiplication algo.

$\Rightarrow$ $A = \begin{bmatrix} 4 & 5 \\ 7 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 3 \\ 5 & 9 \end{bmatrix}$

step (1) :- compute matrix of size $n/2 \times n/2$ from given matrix of size (nxn)

$A = \begin{bmatrix} 4 & 5 \\ 7 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 3 \\ 5 & 9 \end{bmatrix}$

(1x1 1x1) (1x1 1x1)

compair with $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\begin{bmatrix} e & f \\ g & h \end{bmatrix}$

step (2) :- compute 7 product P_1 to P_7

$P_1 = a(f-h) = 4(3-9) = 4 \times -6 = -24$

$P_2 = (a+b) \cdot h = (4+5) \cdot 9 = 9 \times 9 = 81$

$P_3 = (c+d) \cdot e = (7+2) \cdot 2 = 9 \times 2 = 81$

$P_4 = d \cdot (g-e) = 2 \cdot (5-2) = 2 \times 3 = 6$

$P_5 = (a+d)(e+h) = (4+2)(2+9) = 6 \times 11 = 66$

$P_6 = (b-d)(g+h) = (5-2)(5+9) = 3 \times 14 = 42$

$P_7 = (a-c)(e+f) = (4-7)(2+3) = -3 \times 5 = -15$

step ③ :- reconstruction of resultant matrix

$r = P_5 + P_4 - P_2 + P_6 = 66 + 6 - 81 + 42 = 33$

$s = P_1 + P_2 = -24 + 81 = 57$

$t = P_3 + P_4 = 18 + 6 = 24$

$u = P_5 + P_1 - P_3 - P_7 = 66 + (-24) - 18 + 15$

$= 66 - 24 - 18 + 15$

$= 39$

Time complexity = $O(n^3)$

approximately between $O(n^2)$ & $O(n^3)$

$$\therefore \begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} 33 & 57 \\ 24 & 39 \end{bmatrix}$$

Q.13 Use strassen's algorithm to compute matrix product. Show the steps for following matrices.

$$A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 6 & 4 \\ 3 & 2 \end{bmatrix}$$

⇒ step ① :- compute matrices of size $n/2 \times n/2$ from given matrix of size $(n \times n)$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}_{2\times 2}, \quad \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} \qquad B = \begin{bmatrix} 6 & 4 \\ 3 & 2 \end{bmatrix}$$

1×1 1×1 1×1 1×1

PAGE NO.
DATE

compair with $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. $B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$

step (2) :- compute 7 product P_1 to P_7

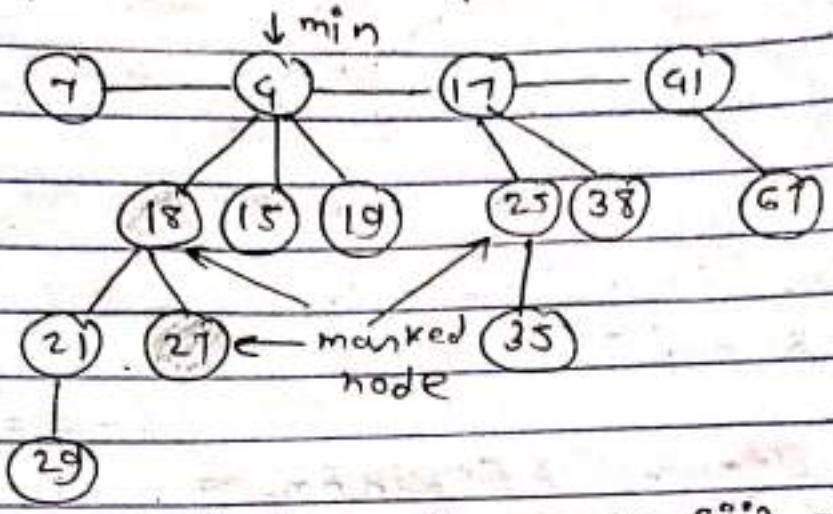
$P_1 = a(f-h) = 1(4-2) = 2$
$P_2 = (a+b)\cdot h = (1+3)\cdot 2 = 4 \times 2 = 8$
$P_3 = (c+d)\cdot e = (5+7)\cdot 6 = 12 \times 6 = 72$
$P_4 = d\cdot(g-e) = 7\cdot(3-6) = 7 \times -3 = -21$
$P_5 = (a+d)(e+h) = (1+7)(6+2) = 8 \times 8 = 64$
$P_6 = (b-d)(g+h) = (3-7)(3+2) = -4 \times 5 = -20$
$P_7 = (a-c)(e+f) = (1-5)(6+4) = -4 \times 10 = -40$

step (3) :- reconstruction of resultant matrix

$r = P_5 + P_4 - P_2 + P_6 = 64 + (-21) - 8 + (-20) = 15$
$s = P_1 + P_2 = 2 + 8 = 10$
$t = P_3 + P_4 = 72 + (-21) = 51$
$u = P_5 + P_1 - P_3 - P_7 = 64 + 2 - 72 - (-40) = 34$

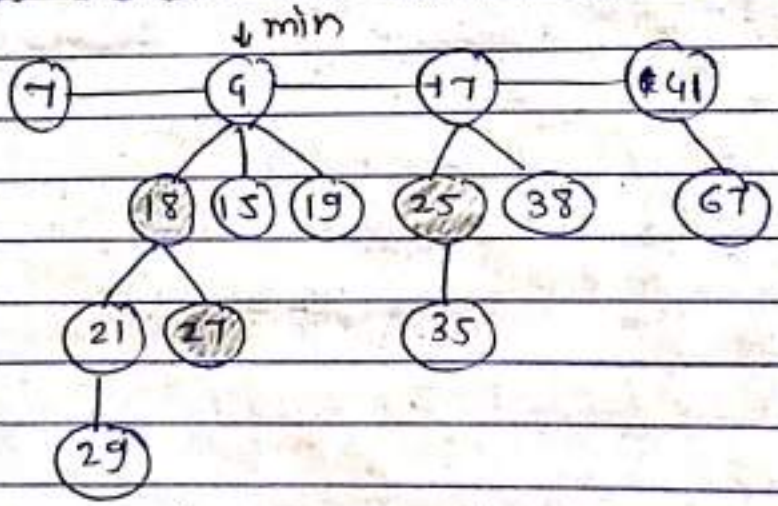
$\therefore \begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} 15 & 10 \\ 51 & 34 \end{bmatrix}$

Q.16 Perform the following operation on fibonacci heap.

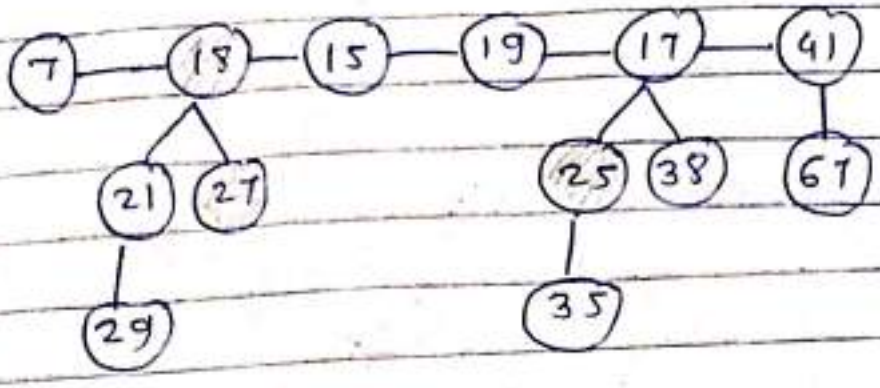


(i) Delete the min node (ii) calculate potential f^n of given heap (iii) Decrement 35 by 5. (iv) Insert 21 in the given fibonacci heap.

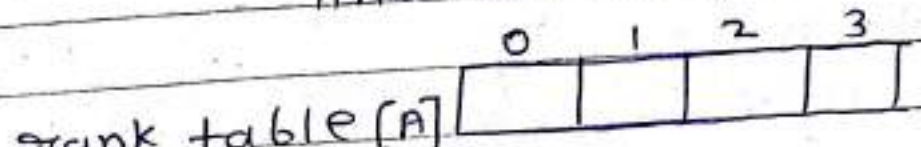
⇒ (i) Delete the min node :-



step (1) :- remove minimum pointer

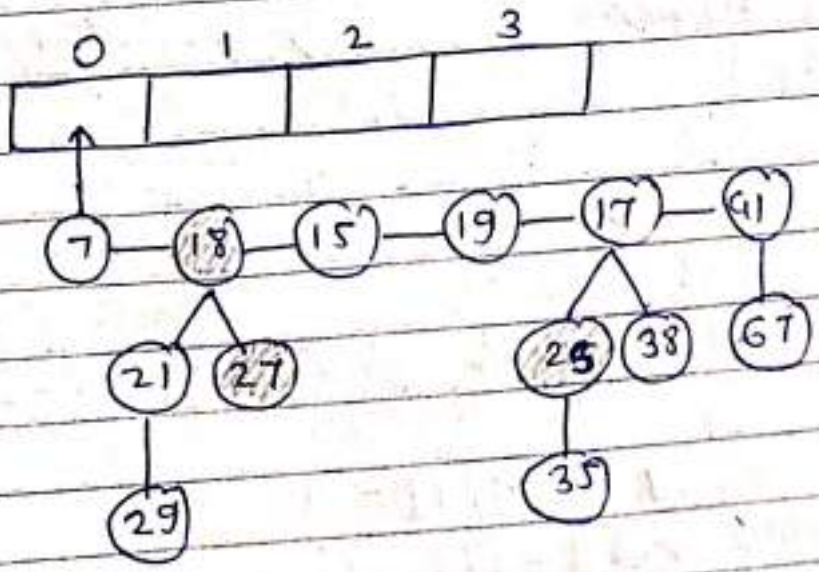


step (2) :- Scan node from left side and link the rank in rank table



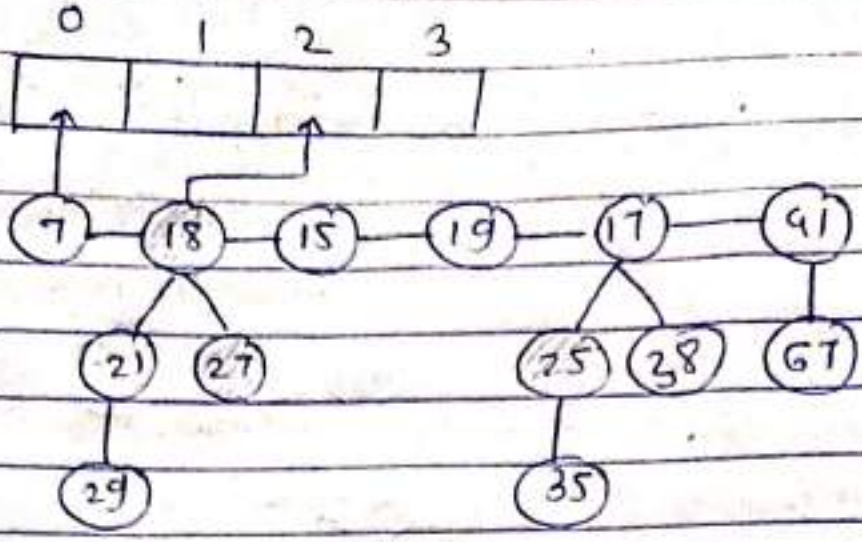
$x = 7$, degree of $7 \Rightarrow d[7] = 0$

$A[0] = NIL$



$x = 18$, $d[18] = 2$

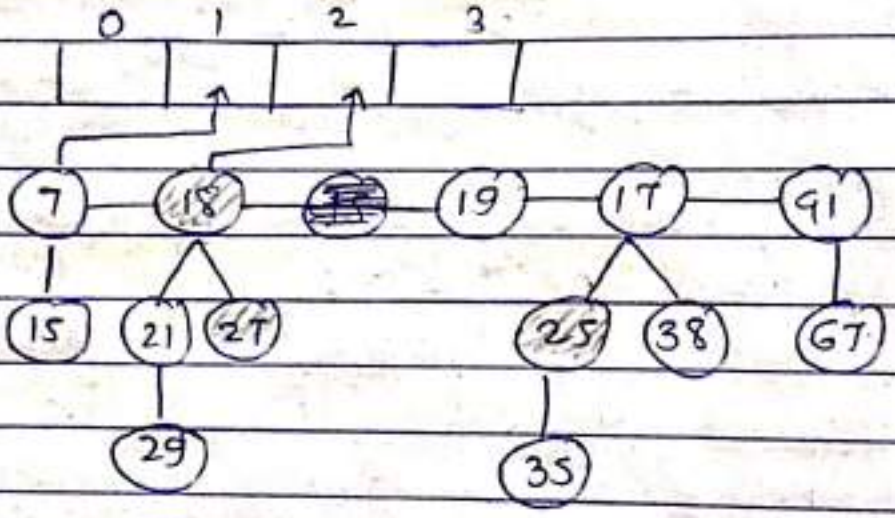
$A[2] = NIL$



$x = 15$, $d[15] = 0$

$A[0] \neq NIL$

To the consolidation ⇒



$x = 7$, $d[7] = 1$

$A[1] = NIL$

&

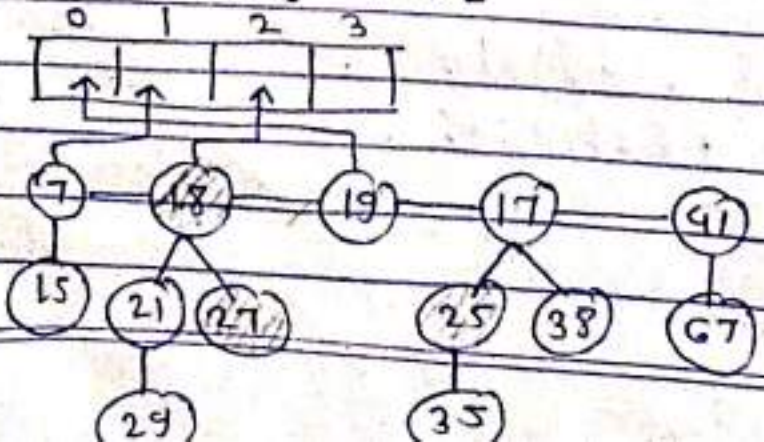
$x = 18$, $d[18] = 2$

$A[2] = NIL$

Now,

$x = 19$, $d[19] = 0$

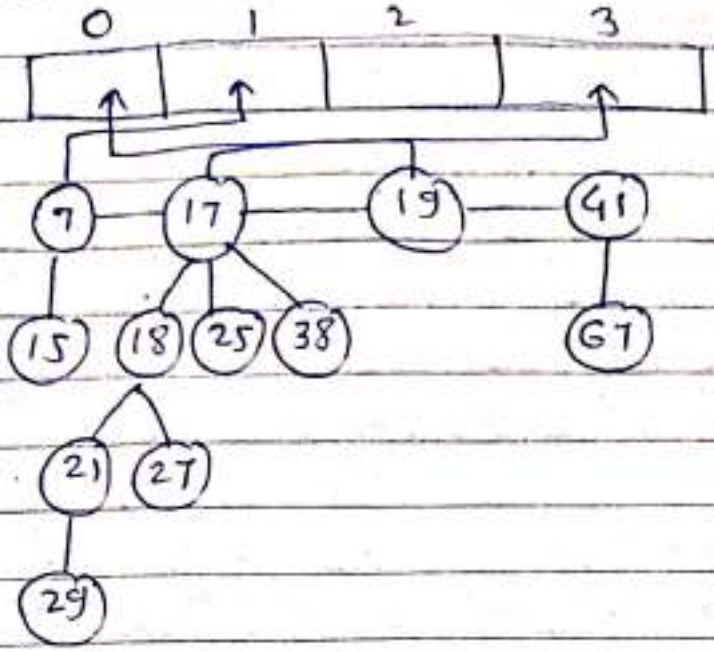
$A[0] = NIL$



PAGE NO.:
DATE

$x = 17$, $d[17] = 2$

$A[2] \neq NIL$



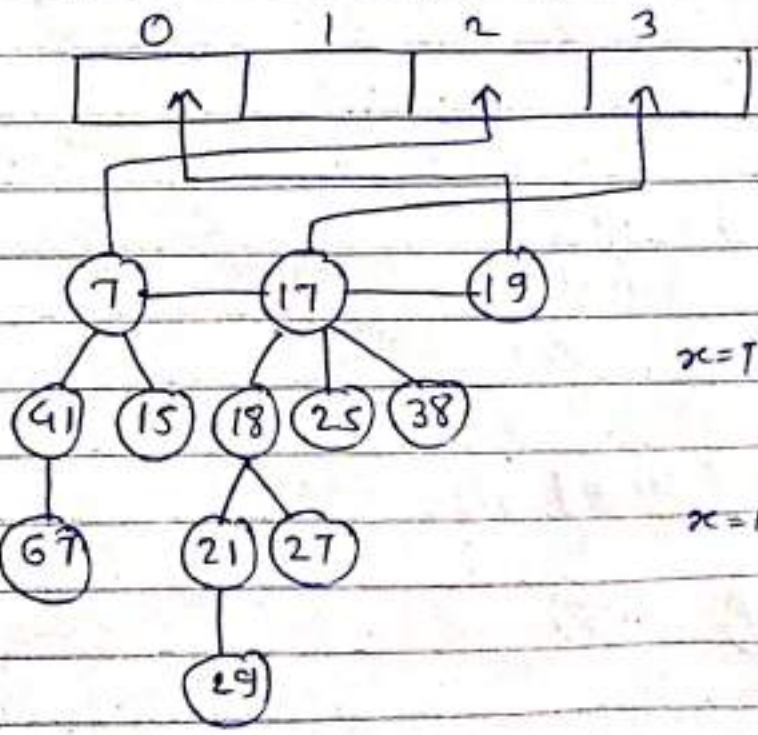
$x = 17$, $d[17] = 3$

$A[3] = NIL$

$x = 41$, $d[41] = 1$

$A[1] \neq NIL$

To the consolidation $\Rightarrow$



$x = 7$, $d[7] = 2$

$A[2] = NIL$

$x = 17$, $d[17] = 3$

$A[3] = NIL$

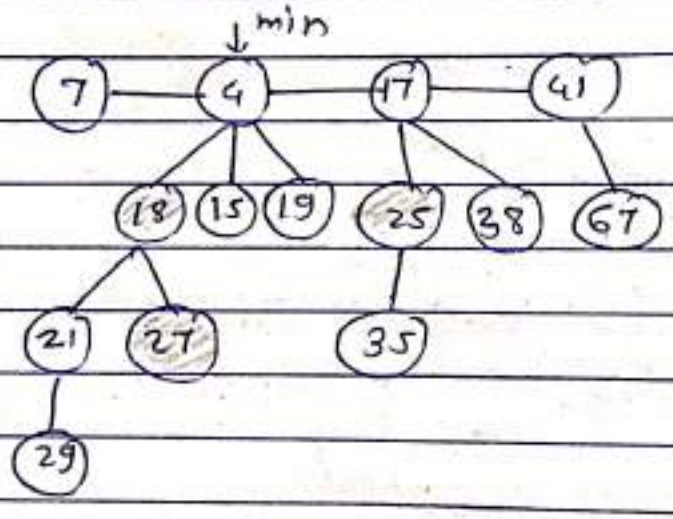
(ii) calculate potential function of given heap.

- trees(H) = 4
  marks(H) = 3

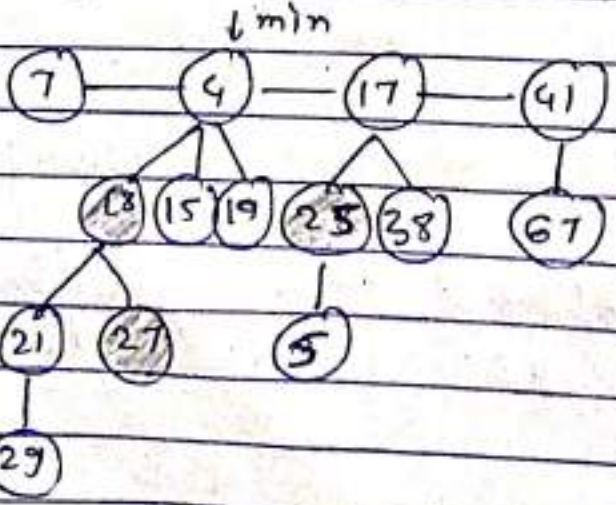
potential function = $\phi(H)$ = trees(H) + 2 marks(H)
= 4 + 2×3
= 4 + 6
∴ $\boxed{\phi(H) = 10}$

(iii) decrement 35 by 5

given,

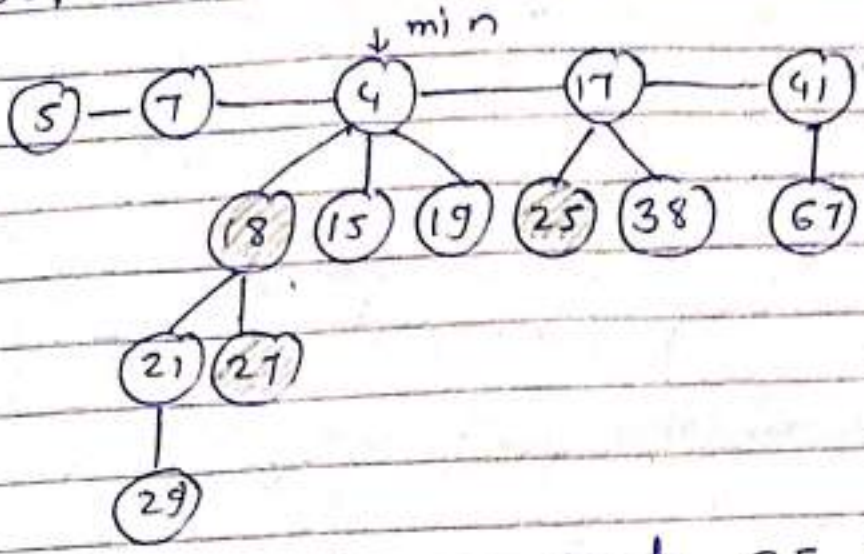


Now, 35 → 5

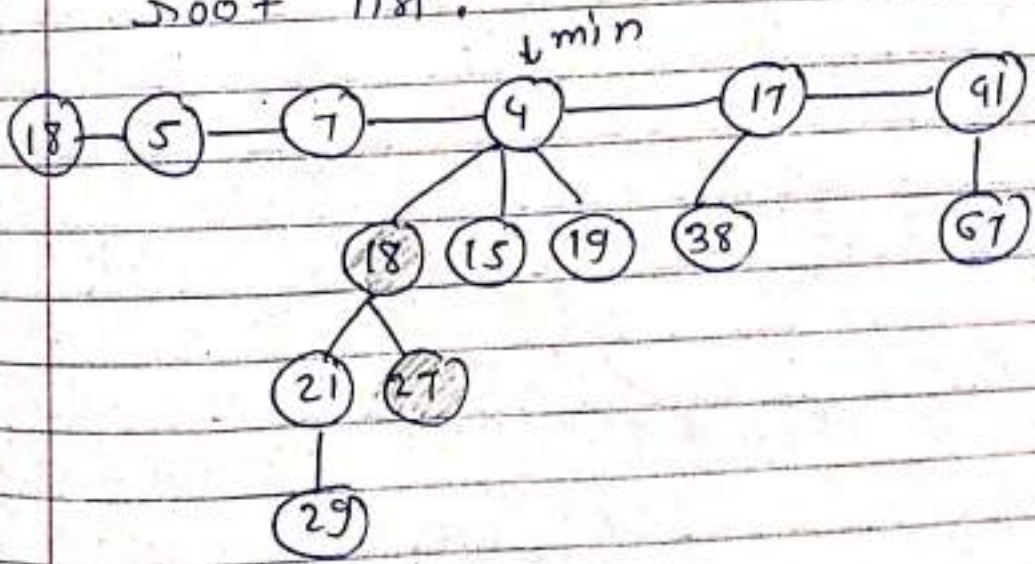


Property is violating,

5 is violating the heap tree property so, meld it into the root list.

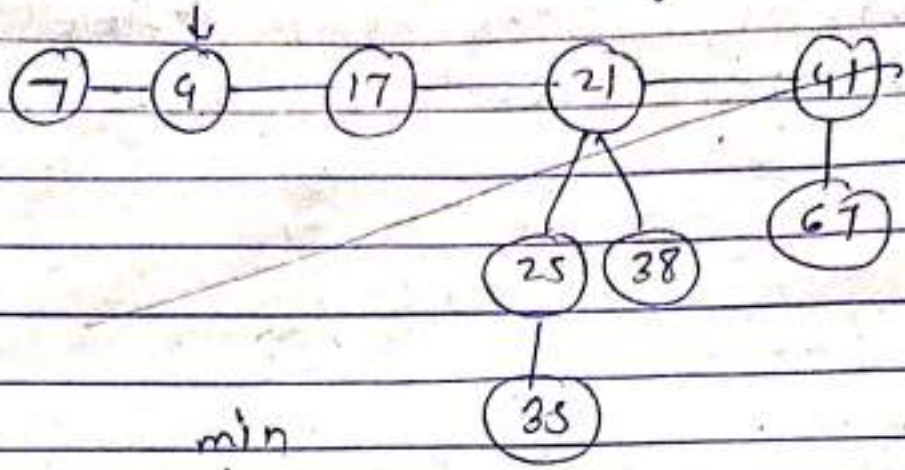


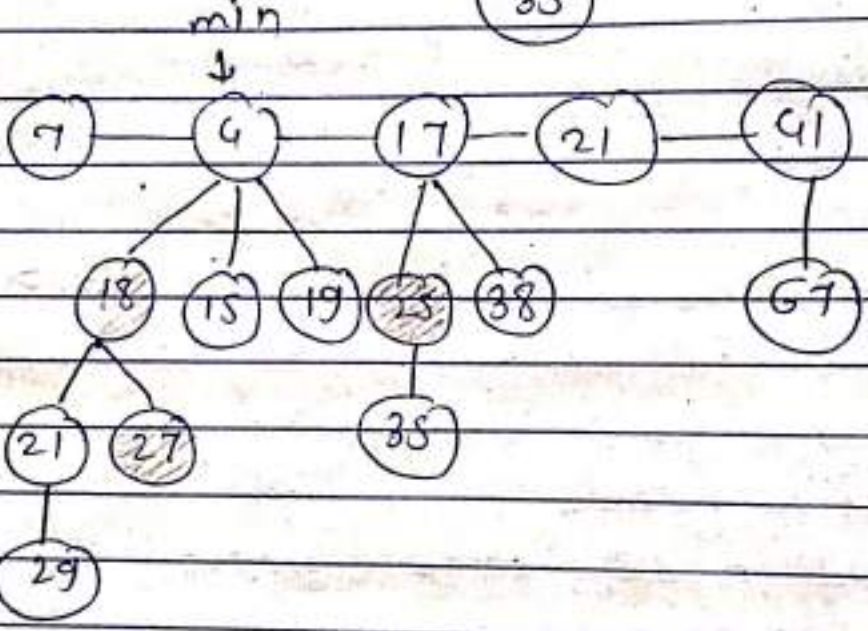
Now, we know that 25 is mark node that means it already lossed his child in history and also now he lossed 5 as it's child node so, 25 will become unmarked node and meld it into root list.



PAGE NO.:
DATE

(iv) insert 21 in the given fibonacci heap.





Q.18 Explain disjoint set representation DS with resp. operation (i) create set (ii) merge set / union (iii) Find. set perform the following sequence of operation. consider set = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

(i) union(2,1) (iii) find(8,9)
(ii) union(4,3)

⇒ given, set = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

step (1) :- Represent all subset individually

(0) (1) (2) (3) (4) (5) (6) (7) (8) (9)

step (2) :- Represent this individual subset in an array.

	0	1	2	3	4	5	6	7	8	9
array	0	1	2	3	4	5	6	7	8	9

union(x, y) → x get merge with y

(i) union (2, 1)

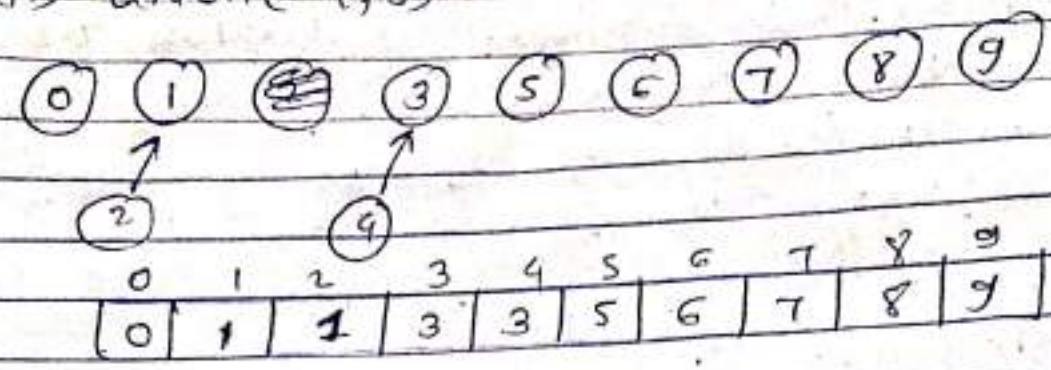
(0) (1) (3) (4) (5) (6) (7) (8) (9)

(2) → (1)

0	1	2	3	4	5	6	7	8	9
0	1	1	3	4	5	6	7	8	9

Ye right hai but ak doubt hai ki que (1) k aage proceed kaise kar ki separately ???

(ii) union(9,3)



0	1	2	3	4	5	6	7	8	9
0	1	1	3	3	5	6	7	8	9

(iii) find(8,9)
→ false
8, 9 are disconnected.

Q.19 Perform union operation in following Binomial heap.