JYOTHIS BENNY
20PMCA135

Question 1

III.A

Quick Sort

66	77	11	88	99	22	33	44	55

taking 66 as pivot.

pivot = 66

0	1	2	3	4	5	6	7	8
66	77	11	88	99	22	33	44	55

↑ pivot, i, j

i = 1 , j = 8

Here i is greater than pivot, so checking j < pivot

Since j < pivot we swap i and j

66	55	11	88	99	22	33	44	77

p, i, j

i = i+1 , incrementing i

JYOTHIS BENNY
20PMCL135

j = j-1

→ So, i = 2, j = 7

array[i] = 11

Here i < pivot, 11 < 66

So incrementing i, i+1

i = 3, j = 7

array[3] = 88

i > pivot so checking j value.

array[j] = 44

44 < pivot so swapping i and j

0	1	2	3	4	5	6	7	8
66	55	11	44	99	22	33	88	77

incrementing ~~both~~ i value ~~and j~~

JYOTHIS BENNY
20PMCI35

decrementing j

→ i = 4 j = 6

Here array[i] > pivot so checking with j

array[j] < pivot ⟶ True

So swapping j and i

0	1	2	3	4	5	6	7	8
66	55	11	44	33	22	99	88	77
				i		j		

incrementing i and decrementing j

→ Here i and j are at same index so we check is pivot.

i > pivot ⟶ False

So we swap i with pivot

0	1	2	3	4	5	6	7	8
22	55	11	44	33	66	99	88	77

Now we get left of ~~66 less~~ pivot is less than 66 and right is greater than 66

JYOTHIS BENNY
20PMEI35

we are doing the same procedure for both left and right

Left sub array = | 22 | 55 | 11 | 44 | 33 |

Right sub array = | 99 | 88 | 77 |

Final Sorted array =

0	1	2	3	4	5	6	7	8
11	22	33	44	55	66	77	88	99

Time Complexity - Worst Case

$$T(n) = [(n+1) + (n) + (n-1) + \cdots + 3 + 2 + T(0)]$$

$$= 1 + 2 + 3 + \cdots + n + n+1$$

$$= (n+1)(n+2)/2$$

JYOTHIS BENNY
20PMCB35

$= (n^2 + 12n + 2)/2 < n^2/2$

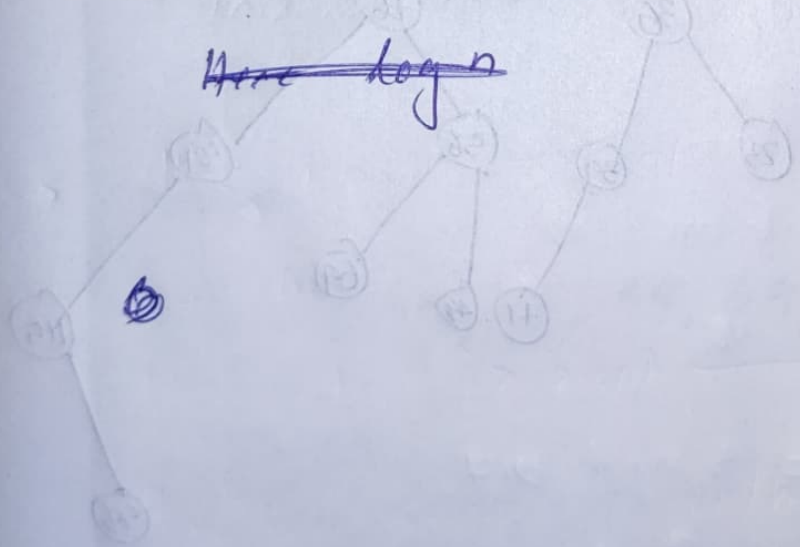
$= O(n^2)$

Time complexity of Best case

$T(n) = O(n \log n)$

Because it works n times $\log n$.

~~$\log n$~~

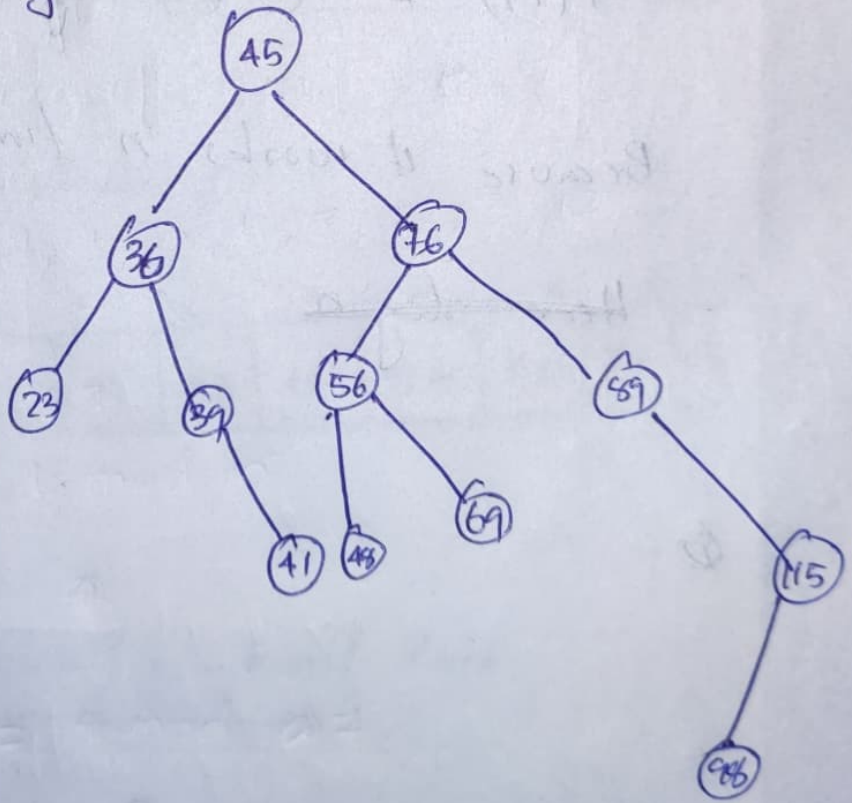


JYOTHIS BENNY
20PMCS35

Question 2

1A.

BST → Binary Search Tree



Inorder (Left, Root, Right)

23, 36, 39, 41, 45, 46, 56, 69, 76, ~~986, 115~~

a)

JYOTHIS BENNY
20PMCA35

89, 98, 115

preoder (Root, left, Right)

45, 36, 23, ~~41~~, 39, 41, 76, 56, 48,

69, 89, 115, 98,

postoder (~~Right, Left~~ Left Right, Root)

~~41, 39,~~

23, 41, 39, 36, 48, 69, 56, 98, 115, 89,

76, 45

JYOTHIS BENNY
20BMEI25

Question 3

11. B

Division Method.

k = key

m = size of array, 7

8, 20, 9, 4, 15, 10, 7, 22, 3, 12

$h(k) = k \bmod m$

$= 0$

$h(8) = 8 \bmod 7$

$= 1$

$h(20) = 6$

$h(9) = 2$

JYOTHIS BENNY
20PMCI35

$h(4) = 4$

$h(15) = 1$

$h(10) = 0$ here 0 is already exists so we have to do other operation to avoid collision
3

$h(7) = 0$

$h(22) = 1$

$h(3) = 3$

$h(12) = 2$ here also collion, so we have to do other operation, like other hash functions.
5

Floading Method

Jyothis Benny
20BMC135

Floading method.

key	8	20	9	4	15	10	7	22	3	12
parts	8	2,0	9	4	1,5	1,0	7	2,2	3	1,2
sum	8	2	9	4	6	1	7	4	3	3
Hash value	8	2	9	4	6	1	7	4	3	3