Q1

findSmallest(arr[])

```
min = arr[0]
for i = 0 to arr.length - 1
        if (arr[i] < min)
                min = arr[i]
return min
```

Time Complexity : O(log n)

Q2 A = ⟨28, 52, 17, 35, 24, 48, 11, 20, 17, 30⟩

① Steps

I — 28, 52, 17 (17 < 52)

II — 17, 28, 52, 35 (35 < 52)

III — 17, 28, 35, 52, 24 (24 < 52)

IV — 17, 24, 28, 35, 52, 48 (48 < 52)

V — 17, 24, 28, 35, 48, 52, 11 (11 < 52)

VI — 11, 17, 24, 28, 35, 48, 52, 20 (20 < 52)

VII — 11, 17, 20, 24, 28, 35, 48, 52, 17 (17 < 52)

VIII — 11, 17, 17, 20, 24, 28, 35, 48, 52, 30 (30 < 52)

IX — 11, 17, 17, 20, 24, 28, 30, 35, 48, 52

Sorted array = ⟨11, 17, 17, 20, 24, 28, 30, 35, 48, 52⟩

(2) Worst Case

The worst case is when the array elements are in decending order

worst case : $O(n^2)$

Best Case

The best case is when the array is already sorted.

best case : $O(n)$

average case : $O(n^2)$

(3)

(3) Insertion Sort runs in $8n^2$ steps

Heap Sort runs in $64 n \log n$ steps

for a certain value of n, Insertion sort must outperform heap sort.

∴ Insertion sort must take less number of steps to execute than heap sort.

∴ $8n^2 < 64 n \log n$

$n^2 < 8 n \log n$

$n < 8 \log n$

∴ for $n > 1$, Insertion sort outperforms heap sort

(4)

(1) node i is present in i^{th} position

$\left\lfloor \frac{i}{2} \right\rfloor$

$\lfloor 9.5 \rfloor$

$= 9$

∴ The parent node of node i is present in 9^{th} position

and i is on the right side of parent node

$\therefore \frac{19}{2} = 9.5$

$\therefore$ node i is the right child

(2) height $= \lfloor \lg n \rfloor$

$= \lfloor \lg 19 \rfloor$

$= \lfloor 4.24 \rfloor$

height $= 4$

(3) no. of leaves $= \lceil n/2 \rceil$

$= \lceil \frac{100}{2} \rceil$

$= 50$

$\therefore$ leaves have no child

$\therefore$ There are 50 nodes with zero child

$\therefore$ There are even no. of leaves there is only 1 node with 1 child

$\therefore$ left 49 nodes all have 2 children

(4) The left subtree has 63 nodes
and the Right subtree has 36 nodes

(5)

(1) no of leaves $\lceil n/2 \rceil = \lceil 100/2 \rceil$

$= 50$

∴ There are 100 nodes ~~out of which 50 are leaves~~
out of which 50 are leaves

position of first leaf = 51

(2) Total no. of leaves = 50

(3) height of heap = $\lfloor \lg n \rfloor$

$= \lfloor \lg 100 \rfloor$

$= \lfloor 6.64 \rfloor$

$= 6$

(4) There are 8 nodes at height 3 of the heap