Rahul Saini

11912068

I. T.

① Best Case time complexity of Insertion sort :→

Given array:- 2, 3, 4, 5

i	0	1	2	3	4
Element					
j					
key			3	4	5
no. of compulsion			0	0	1
Movements			1	1	1

Rahul

Time complexity = 1+1+1 = 3

$\simeq O(n-1)$

$\simeq O(n)$

② find Elements smaller than 10

main

10 20 30 40 ∞

j i

step:→3 Swap the main element with jth element.

$\underbrace{1+}_{\text{swap}} \quad \underbrace{n}_{\text{comparisons}}$

$\Rightarrow$ 20 30 40 50 so.

do step 1 with 20 — do step 2 with 20

40 50

do step 1 with 40 — do step 2 with 50

40 50 Rahul.

$\underbrace{1}_{\text{swap}}$ $\quad$ $\underbrace{n-3}_{\text{comparisons}}$

If a list contain 'n' element than time complexity :⇒

$(1+n)+(1+n-1)+1+n-2+1+n-3+\ldots+1+1$

$=(\underbrace{1+1+\ldots+1}_{n\text{ time}})+(n+n-1+n-2+\ldots+1)$

$= n + \frac{n(n+1)}{2}$

$$2n + \frac{n^2+n}{2}$$

$$= \frac{3n+n^2}{2}$$

$\approx O(n^2)$ worst case.

Rahul

Bubble Sort

```
Void Bubble sort (int arr [], int n)
{ ~~p~~ int i, j;
        for (i=0; i<n-1; i++)
        { for (j=0; j< n-i-1; j++)
          { if (arr[j] > arr[j+1])
            swap[&arr[j], &arr[j+1]]
          }
        }
```

Worst case :→

		inner loop
When	$i = 0$	n times
	$i = 1$	n-1 times
	$i = 2$	n-2 times
	⋮	⋮
	$i = n-2$	2 times

Total $= n + n-1 + n-2 + n-3 + \dots + 2$

$$= \frac{n(n+1)}{2} - 1$$

Rahul

$$\approx O(n^2)$$

Algorithm name	Cases	Time
Quick Sort	Best Average	n log n
	Worst	n^2
Merge Sort	Best Average Worst	n log n

Algorithm name.	Cases	Time
Quick Sort	Best Averge	$n \log n$
	Worst	n^2
Merge Sort	Best Average Worst	$n \log n$
Insertion Sort	Best	n
	Average Worst	n^2
Bubble Sort	Best	n
	Average Worst	n^2

Rahul.