* Selection Sort Algorithm:

1. Scan the array to find the smallest element swapping with first element.
2. Starting from the 2nd element scan the array to find the smallest element swapping with 2nd element.
3. This process is repeated until least is sorted.

70, 30, 20, 50, 60, 10, 40

10

10, 30, 20, 50, 60, 70, 40

20

10, 20, 30, 50, 60, 70, 40

40

10, 20, 30, 40, 60, 70, 50

50

10, 20, 30, 40, 50, 70, 60

60

10, 20, 30, 40, 50, 60, 70

- Algorithm:-

```
for i=1 to n-1 do
    min j ← i;
    min x ← A[i]
for j = i+1 to n do
    if A[j] < min x then
        min x ← j;
        min x ← A[j];
```

end
end
A[minj] ← A[i];
A[i] ← min x;
end

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 5 | 2 | 1 | 4 | 3 |

e.g:

i = 1 to 4
min j = 1
min x = A[1]
min x = 5
j = 2 to 5
A[2] < 5
2 < 5 True
min j = 2
min x = A[2]
min x = 2
j = 3 to 5
A[3] < 2
1 < 2 True
min j = 3
min x = A[3]
min x = 1
j = 4 to 5
A[4] < 1
4 < 1 false
j = 5 to 5
A[5] < 1
3 < 1 false

A[3] = A[1]
A[3] = 5
A[1] = min x
A[1] = 1

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 1 | 2 | 5 | 4 | 3 |

i = 2 to 4
min j = 2
min x = A[2]
min x = 2
j = 3 to 5
A[3] < 2
5 < 2 false
A[4] < 2
4 < 2 false
A[5] < 2 2
3 < 2 false
No Swapping

i = 3 to 4
min j = 3
min x = A[3]
min x = 5
j = 4 to 5
A[4] < 5
4 < 5 True
min j = 4
min x = A[4]
min x = 4
A[5] ≠ 4
3 < 4 True.
min j = 5
min x = A[5]
min x = 3
A[5] = A[3]
A[5] = 5
A[3] = 3
1 2 3 4 5

i = 4 to 4
min j = 4
min x = A[4]
min x = 4
j = 5 to 5
A[5] < 4
5 < 4 false
No Swapping
1 2 3 4 5

- Time Complexity :-

In Selection Sort algorithm after each iteration one element get Sorted.
So problem size reduced by one.

$$T(n) = T(n-1).$$

Inner for loop excutes for n no. of time for

$$T(n) = T(n-1) + n \quad \text{— (I)}$$

Replace n by n-1

$$T(n-1) = T(n-2) + (n-1) \quad \text{— (II)}$$

put $T(n-1)$ in eqn (i)

$T(n) = 2T(n-2) + (n-1) + n$ — (iii)

Replace n by n-2 in eqn (i) or n by n-1 in eqn (ii)

$T(n-2) = T(n-3) + (n-2)$

put $T(n-2)$ in eqn (iii)

$T(n) = T(n-3) + (n-2) + (n-1) + n$

$\vdots$

$T(n) = T(n-k) + (n-k+1) + (n-k+2) + \cdots + n$

put $n = k$

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

$T(0) = 0$

$T(n) = 1 + 2 + 3 + \cdots + n$

$= \Sigma n$

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

Complexity $= O(n^2)$ [Best, Worst & average]

Q.1] Sort the following array by using Selection Sort Algorithm

50, 40, 5, 9, 45, 90, 65, 25, 75.

5

5, 40, 50, 9, 45, 90, 65, 25, 75

9

5, 9, 50, 40, 45, 90, 65, 25, 75

25

5, 9, 25, 40, 45, 90, 65, 50, 75

45

5, 9, 25, 40, 45, 90, 65, 50, 75

50

5, 9, 25, 40, 15, 50, 65, 10, 75

15

5, 9, 25, 40, 15, 50, 65, 75, 90

Q.2] Scan the following word by using Selection Sort Algorithm.

EXAMPLE

E X A M P L E

A

A X E M P L E

E

A E X M P L E

E

A E E M P L X

L

A E E L P M X

M

A E E L M P X

| A | E | E | L | M | P | X |
|---|---|---|---|---|---|---|

Date 29/10/25
Page

* Insertion Sort:-

30, 70, 20, 50, 40, 10, 60

| 30 | 70 20 50 40 10 60 |
|---|---|
| 0 | Sorted / Unsorted |

| 30 | 70 | 20 50 40 10 60 |
|---|---|---|
| 0 | 1 | Unsorted |

| 20 | 30 | 70 | 50 40 10 60 |
|---|---|---|---|
| 0 | 1 | 2 | Unsorted |

| 20 | 30 | 50 | 70 | 40 10 60 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | Unsorted |

| 20 | 30 | 40 | 50 | 70 | 10 60 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | Unsorted |

| 10 | 20 | 30 | 40 | 50 | 70 | 60 |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | Unsorted |

| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

- Algorithm:-

```
for j = 2 to A.length
    key = A[j]
    i = j - 1
while i>0 and A[i] > key
    A[i+1] = A[i]
    i = i-1
end
    A[i+1] = key
end
```

- example :-

| 2 | 13 | 5 | 18 | 14 |
|---|---|---|---|---|
| 1 | 2 | 3 | 5 | 4 |

j = 2 to 5
key = A[2]
= 13
i = 1
1>0 and 2>13 -- false
No Swapping

j = 3 to 5
key = A[3]
= 5
i = 2
1>0 and 13 > 5 -- True
1>0 and 2>13 false

A[3] = A[2]
A[3] = 13
i = 1
A[2] = key

A[2] = 5

| 2 | 5 | 13 | 18 | 14 |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

j = 4 to 5

key = A[4]

key = 18

i = j-1 = 3

3>0 and A[3]>key

3>0 and 13>18 -- false

No swapping

J = 5

key = A[5]

key = 14

i = 4

4>0 and A[4]>key

4>0 and 18>14 -- true

A[i+1] = A[i]

A[5] = A[4]

A[5] = 18

i = 3

3>0 and A[3]>14

3>0 and 13>14 -- false

A[i+1] = key

A[4] = 14

| 2 | 3 | 13 | 14 | 18 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

* Find Time Complexity of insertion sort.

—>

1. Best Case :-

When list is already sorted, so condition in the while loop is never executed so $t_j = 1$

So the recurrance relation is,

$$T(n) = C_1 n + C_2(n-1) + C_3 \cdot (n-1) + C_4 \sum_{j=2}^{n} t_j + C_5 \cdot \sum_{j=2}^{n} t_{j-1} + C_6 \sum_{j=1}^{n} t_{j-1} + C_7 (n-1)$$

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} 1 = (1+1+\cdots+1) = n-1 \text{ times}$$

$$\sum_{j=2}^{n} t_j - 1 = \sum_{j=2}^{n} 1 - 1 = 0$$

$$T(n) = C_1 \cdot n + C_2(n-1) + C_3(n-1) + C_4(n-1) + C_5 \times 0 + C_6 \times 0 + C_7(n-1)$$

$$= [C_1 + C_2 + C_3 + C_4 + C_7]n - [C_2 + C_3 + C_4 + C_7]$$

$$T(n) = an - b$$

Complexity $= O(n)$

2. Worst Case :-

In the worst case, complexity, elements are arranged in reverse order, so the condition in while loop is executed, so $t_j = j$.

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} j = 2 + 3 + 4 + \cdots + n \quad \text{---(1)}$$

Add & Sub 1 in eqn (1),

$$\sum t_j = (1+2+3+\cdots+n)-1$$
$$= \sum n-1$$
$$= \frac{n(n+1)}{2} - 1$$

$$\sum_{j=2}^{n} t_j - 1 = \sum_{j=2}^{n} j-1 = (2-1)+(3-1)+\cdots+(n-1)$$
$$= 1+2+3+\cdots+n-1$$
$$= \sum n-1$$
$$= \frac{n(n-1)}{2}$$

Put $\sum t_j$ & $\sum t_j - 1$ in eqn (1)

$$T(n) = C_1 n + C_2(n-1) + C_3(n-1) + C_4\left(n(n-1)-1\right) + C_5\left(\frac{n(n-1)}{2}\right) + C_6\left(\frac{n(n-1)}{2}\right) + C_7(n-1)$$

$$= \left[\frac{C_4}{2} + \frac{C_5}{2} + \frac{C_6}{2}\right] n^2 + \left[C_1 + C_2 + C_3 + \frac{C_4}{2} - \frac{C_5}{2} - \frac{C_6}{2}\right] n$$

$$- \left[C_2 + C_3 + C_4 + C_7\right]$$

$$= an^2 + bn + c$$

Complexity $= O(n^2)$

3. Average Case:-

- In average case Complexity, Some of the elements are Sorted and remaining elements are unsorted, So $t_j = j-1$

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} \frac{j-1}{2} = \frac{1}{2} \sum_{j=2}^{n} j-1 = \frac{1}{2} \frac{n(n-1)}{2}$$

$$\sum_{j=2}^{n} t_j - 1 = \sum_{j=2}^{n} \frac{j-1}{2} - 1 = \sum \frac{j-3}{2}$$

$$= \frac{n(n-3)}{2 \times 2}$$

put $\sum t_j$ & $\sum t_j - 1$ in eqn (1).

$$T(n) = C_1 \cdot n + C_2(n-1) + C_3(n-1) + \frac{C_4}{2} \frac{n(n-1)}{2} +$$

$$(C_5 + C_6) \times \frac{n(n-3)}{2 \times 2} + C_7(n-1)$$

$$T(n) = an^2 + bn - c$$

Complexity $= O(n^2)$

* Sort the following word by using insertion sort

E X A M P L E