Experiment-3

Aim - WAP to analyse the time complexity of selection sort and bubble sort.

Software Used :- Turbo C

Theory :-

Algorithm :-

Selection Sort.

```
for i → 1 to n-1                          - C1
    m = a[i]                              - C2
    p=i
    for j → i+1 to n                      - C3
        if (m > a[j])                     - C4
            m = a[j]                      - C5
            p = j
        end if
    end for
    set a[p] = a[i]                       - C6
        a[i] = m
end for
```

Worst case:

$Tn = C_1(n) + C_2(n) + \sum_{i=1}^{n} C_3(n-i) + \sum_{i=1}^{n} C_4(n-i) + C_5 \sum_{i=1}^{n} (n-i)$
$+ C_6(n)$

$Tn = ((C_1 + C_2 + C_6)(n) + \sum_{i=1}^{n} (n-i) ((C_3 + C_4 + C_5)$

$= ((C_1 + C_2 + C_6)n + \frac{n(n+1)}{2} ((C_3 + C_4 + C_5)$

$= ((C_1 + C_2 + C_6)(n) + \frac{n^2+n}{2} ((C_3 + C_4 + C_5)$

Tn can also be written as
$Tn = O(n^2)$

Best case:

$Tn = ((C_1 + C_2 + C_6)n + \sum_{i=1}^{n} C_3(n-i) + C_4(1) + C_5(1)$

$= ((C_1 + C_2 + C_6)n + \frac{n(n+1)}{2} C_3$

$= ((C_1 + C_2 + C_6)n + \left(\frac{n^2+n}{2}\right) C_3$

$Tn = O(n^2)$

Bubble Sort

```
for i→1 to n                                   – $C_1$
    for j→1 to n-i-1                           – $C_2$
        if (a[j] > a[j+1])                     – $C_3$
            swap (a[j] and a[j+1])             – $C_4$
        endif
    end for
end for
```

Worst case:

$$Tn = C_1(n) + (C_2+C_3+C_4) \sum_{j=1}^{n} (n-j-1)$$

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

$$= C_1(n) + (C_2+C_3+C_4) \left(\frac{n^2-n}{2}\right)$$

Since, highest degree of n is 2
Using the Big O notation
time complexity $Tn = O(n^2)$

the trade union has been establishm

Best Case:

$T_n = O(n)$

as we are using a flag to skip extra iterations i.e we would skip the iterations of inner loop, i.e only one inner loop per outer loop. Provided, if any swap took place in previous iteration.

Result:-

The time complexity of selection sort and bubble sort studied and analyzed.