23/9/23

Sorting

Date: 24

1 -> Bubble Sort

Sorting -> generally means Ascending Order

0	1	2	3	4
5	4	5	2	1

↳ Sort →

0	1	2	3	4
1	2	3	4	5

→ Swap the Adjacents if needed till we get all the array sorted.

Ex

i	i+1			
0	1	2	3	4
5	4	5	2	1

swap

4 Comparison

→ 4 | 5 | 3 | 2 | 1 (swap)

→ 4 | 3 | 5 | 2 | 1

→ 4 | 3 | 2 | 5 | 1

→ 4 | 3 | 2 | 1 | 5

1st Iteration

4 | 3 | 2 | 1 | 5 ⟸

↳ 1st Largest At its position

2 -> 2nd Largest At its position

4 | 3 | 2 | 1 | 5 → 3 | 4 | 2 | 1 | 5 → 3 | 2 | 4 | 1 | 5 → 3 | 2 | 1 | 4 | 5

3 Comparison

23/9/22

APCO
Date: 9/9

(3) => 3rd Largest At its position

3 | 2 | 1 | 4 | 5

2 | 3 | 1 | 4 | 5 <- 2 Comparison

2 | 1 | 3 | 4 | 5

(4) => 4th Largest At its position

2 | 1 | 3 | 4 | 5 <- 1 Comparison

1 | 2 | 3 | 4 | 5

N = 5

Iteration = N - 1 = 4

Iteration	Comparisons
i = 0 I	4 (n - i - 1)
i = 1 II	3
i = 2 III	2
i = 3 IV	1

Generalize =>

N = n

1 -> (n - 1) Comp
2 -> (n - 2) Comp
3 -> (n - 3) Comp
.
.
.
(n - 1) -> 1 Comp

=> SUM = $\frac{n(n-1)}{2} = \frac{n^2 - n}{2}$

23/9/23

Date: APCO 250

$O(n) = O\left(\frac{n^2}{2} - \frac{n}{2}\right)$

T.C = $O(n^2)$
Space = $O(n)$

Code →

```
for(int i=0; i<n; i++){
    for(int j=0; j<n-i-1; j++){
        if(v[j] > v[j+1]){
            swap(v[j], v[j+1]);
        }
    }
}
```

DRY Run → 5|4|3|2|1 i=0

j=0 n-0-1 = 4

4|5|3|2|1

4|3|5|2|1

4|3|2|5|1 → 4|3|2|1|5

i=1

j=0 n-1-1 = 3

4|3|2|1|5

3|4|2|1|5 → 3|2|4|1|5

APCO

Date: 2 5 1

3	2	1	4	5

$i = 0$, $L = 2$

$j = 0$, $n - 2 - 1 = 2$

3 2 1 4 5 → 2 3 1 4 5 → 2 1 3 4 5

$i = 1$, $n - 1 - 1 = 3$

$j = 0$, $n - 3 - 1 = 1$

2 1 3 4 5

↓

1 2 3 4 5