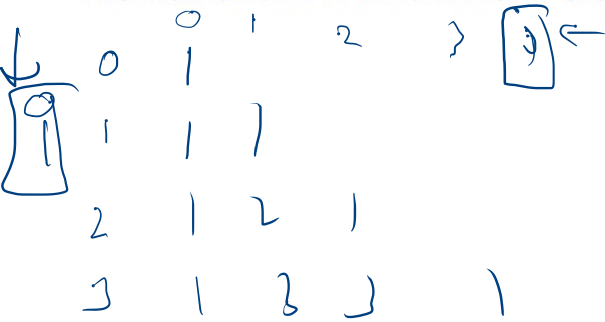


Recursion Binomial Pattern (Day 28)



0	1	1	2	3
1	1	1		
2	1	2	1	
3	1	3	3	1

$$0C_0$$

$$1C_1$$

$$1C_2$$

$$2C_0$$

$$2C_1$$

$$2C_2$$

$$3C_0$$

$$3C_1$$

$$3C_2$$

$$3C_3$$

$$nCr = \frac{n!}{(n-r)! \times r!} = nC_{r-1} \times \frac{n-r+1}{r}$$

$$2C_1 = \cancel{2C_0}^1 \times \frac{2}{1} = 2$$

$$\eta_{C_2} = \eta_{C_{2-1}} \times \frac{n-2+1}{2}$$

$$i_{cj} = i_{cj-1} + \frac{i - j + 1}{n}$$

$$b(i, j) = b(i, j-1) * \frac{(i-j+1)}{j}$$

HW \Rightarrow Revisions

for (part 2)
 given
 scatter ()

→ recursive relation

\rightarrow 0 1 2 3
 \rightarrow 1 1 1
 \rightarrow 2 1 2 1 \nearrow cum
 \rightarrow 3 1 3 (3) 1



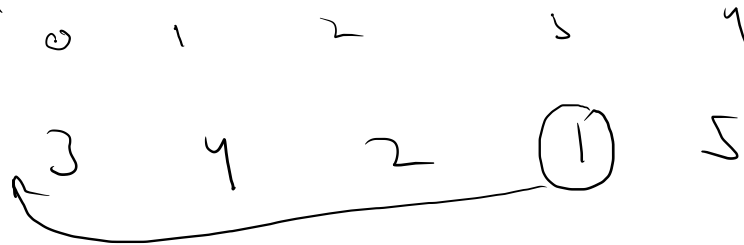
$$f(i, j) = f(i, j-1) * \frac{(i-j+1)}{j}$$

Sorting

arrange ele in order $\left\{ \begin{array}{l} \rightarrow \text{Asc} \\ \leftarrow \text{Desc} \end{array} \right.$

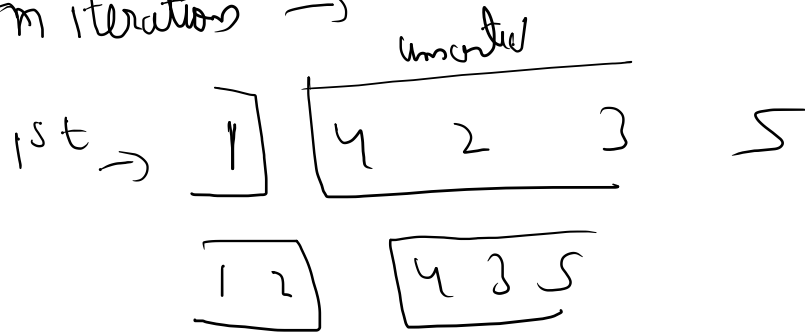
Selection sort algorithm (Day 30) (ASC)

First
2



5 → elements → n

n iterations →

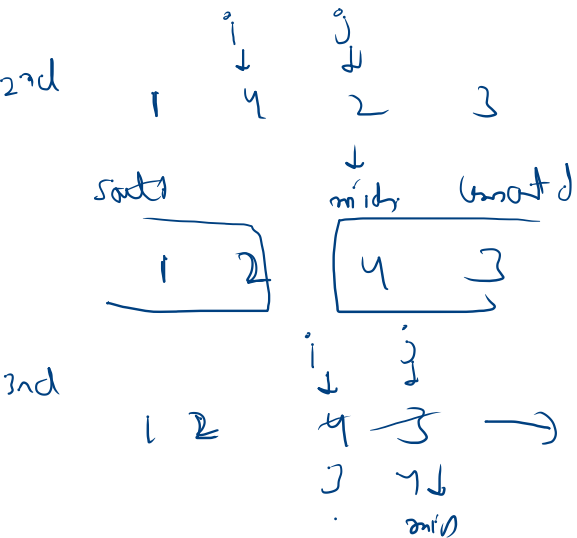
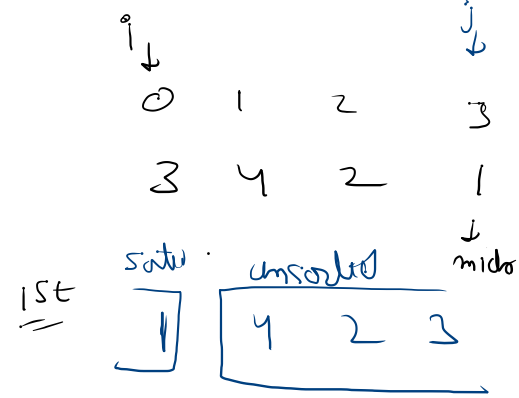


①

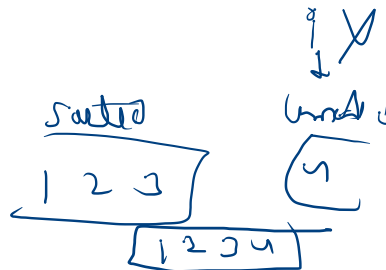
$n-1$ iterations → no. of elements

②

Sort from first
& put one element in
correct position



In placing ✓
not
new
arr ✗



```
public static void solution(int[] arr) {

    int n = arr.length;

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

```
public static void solution(int[] arr) {
```

```
    int n = arr.length;
```

```
    for (int i = 0; i < n - 1; i++) {  $(n-1)$ 
```

```
        int minIdx = i;
```

```
        for (int j = i + 1; j < n; j++) {  $(n-1)$ 
```

```
            if (arr[minIdx] > arr[j]) {
```

```
                minIdx = j;
```

```
            }
```

```
        }
```

```
        swap(arr, i, minIdx);
```

```
    }
```

```
}
```

Time $\Rightarrow O(n^2)$

Space $\Rightarrow O(1)$
 \hookrightarrow mp100

Bubble Sort Algorithm (Day 30)

1st

Unsorted

0	1	2	3
4	3	5	7
3	4	1	5
			2

Sorted

4
2
5

n-1 iterations

Largest elem

3	4	1	2
---	---	---	---

5

2nd

3

4

1

L

1

4

2

L

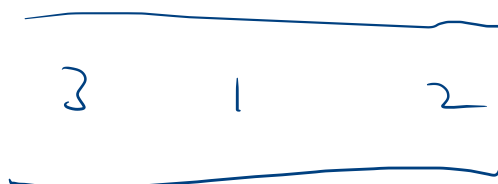
2

4

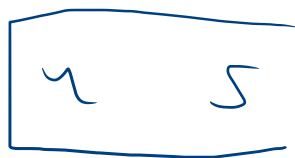
end of
unsorted

5

end 2nd
=)



sorted



ans

8

1

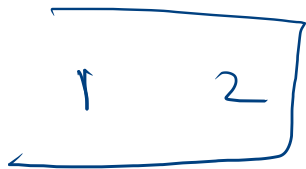
2

4

8

2

unsorted

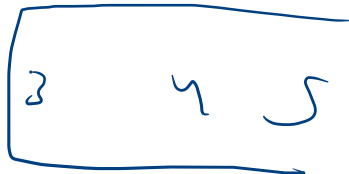


2

4

3

sorted



4

5

ans →

4 (1)

2
1
0

2
2
1

3

4

5

Unsorted

Sorted

and

1

2 3 4 5

In Place algo

$$T(n) = O(n^2)$$

$$S(n) = O(1)$$

Select Sort

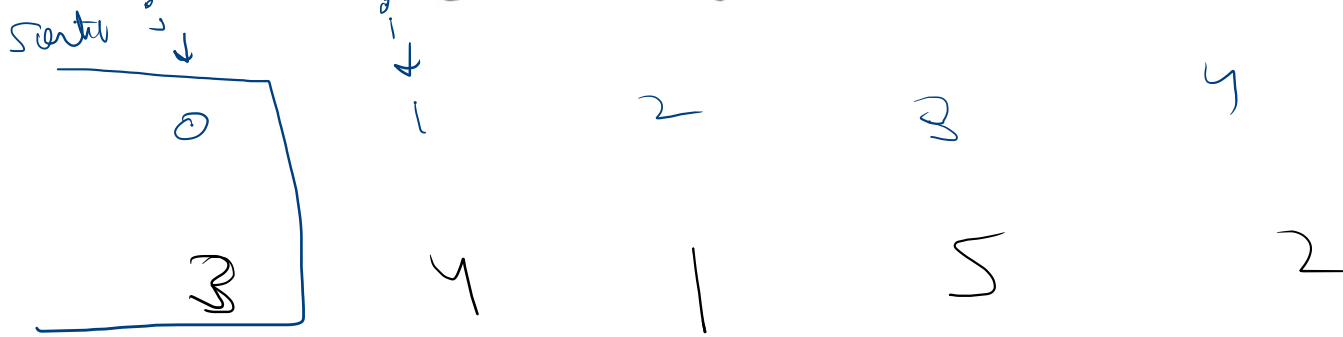
```
public static void solution(int[] arr) {  
  
    int n = arr.length;  
  
    for (int i = 0; i < n - 1; i++) {  
        int minIdx = i;  
        for (int j = i + 1; j < n; j++) {  
            if (arr[minIdx] > arr[j]) {  
                minIdx = j;  
            }  
        }  
        swap(arr, i, minIdx);  
    }  
}
```

3 1 2 4 5

Bubble Sort

```
public static void solution(int[] arr) {  
  
    int n = arr.length;  
  
    for (int iter = 1; iter < n; iter++) {  
        for (int j = 0; j < n - iter; j++) {  
            if (arr[j] > arr[j + 1]) {  
                swap(arr, j, j + 1);  
            }  
        }  
    }  
}
```

Insertion sort algorithm (Day 30)

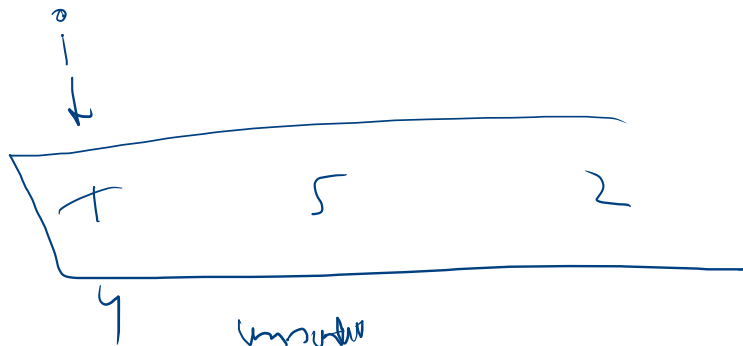
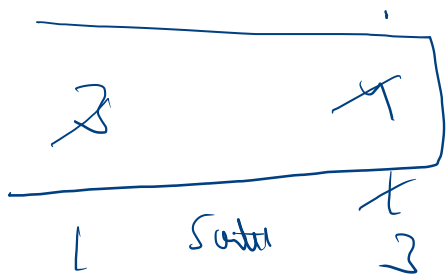


(n-1)

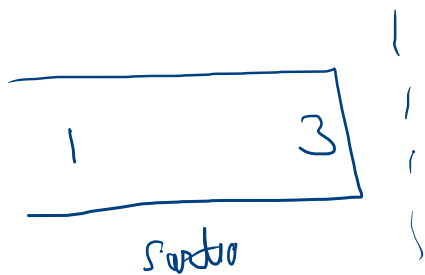
Insert

- ① Inserting a element in your sorted region
- ② After inserting validate whole sorted region

2nd



3rd



3rd

0
1
2
↓

1

3

4

5

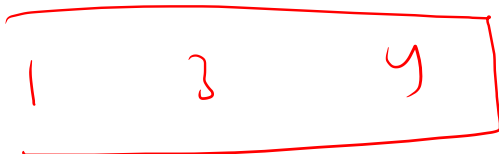
2

sorted

unsorted

new sorted

new unsorted



sorted



unsorted

2nd

9A

1

1

3
2

4
X
2

5
X
4

0
1
X
5

sorted

unsorted

sorted

new

unsorted

and

1

2

3

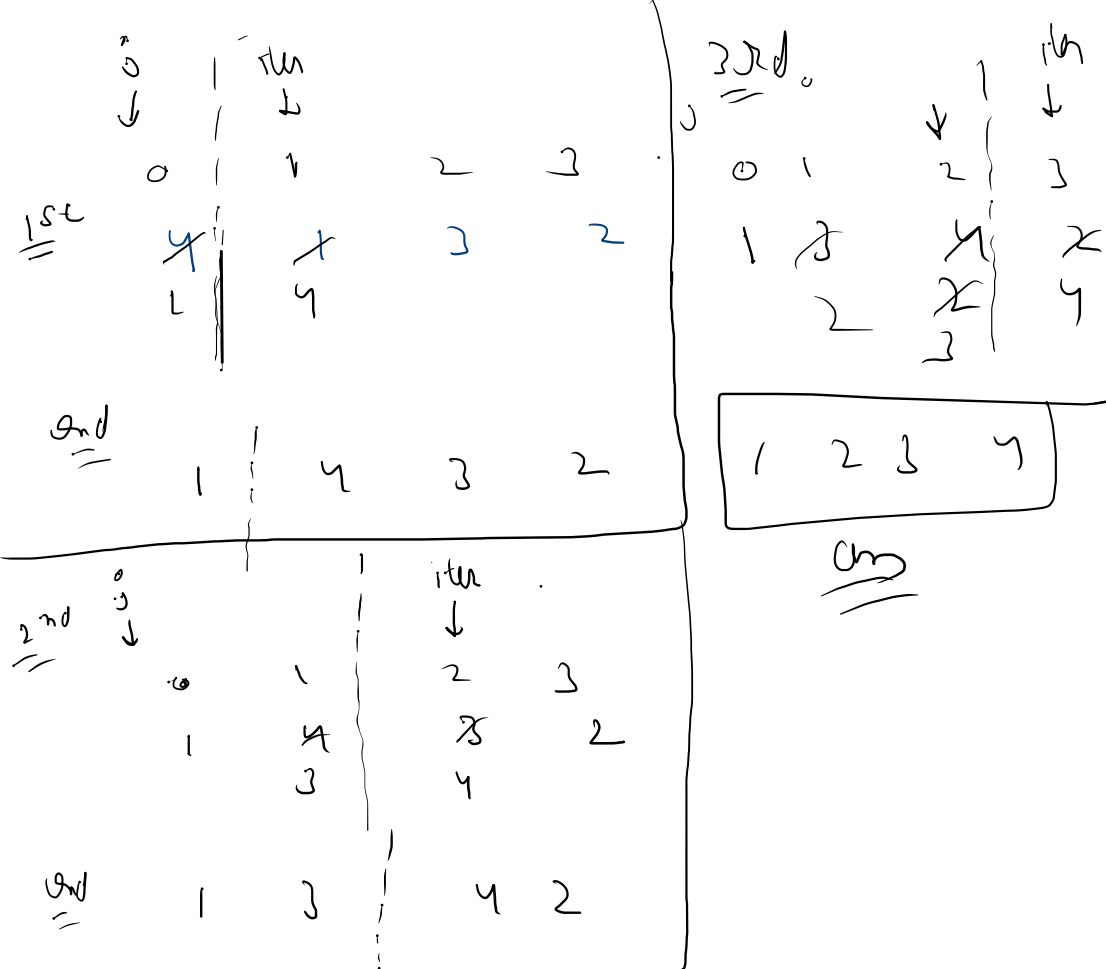
4

5 \Rightarrow ans

$$Time(n) = O(n^2)$$

$$Space(n) = O(1)$$

$$Inplace = \checkmark$$



```
public static void solution(int[] arr) {

    int n = arr.length;

    for (int iter = 1; iter < n; iter++) {

        // validation of inserted element
        for (int j = iter - 1; j ≥ 0; j--) {

            if (arr[j] > arr[j + 1]) {
                swap(arr, j, j + 1);
            } else { // optimisation
                break;
            }
        }
    }
}
```


Sorted

0

~~3~~

1

unsorted

2

1

2

3

4

4

2

~~4~~
3

5

1st pass

min_idx = 1

for (j = i + 1; j < n) {
 update min_idx

}

swap(i, min_idx)

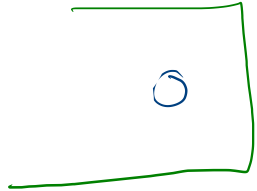
i = 0 to n-1)

j = i+1 to n)

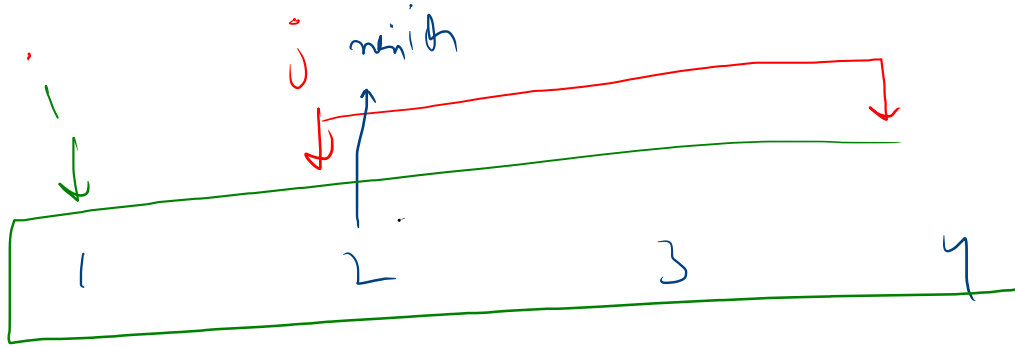
2nd iter

start

sorted



~~3~~
↑



~~4~~
2

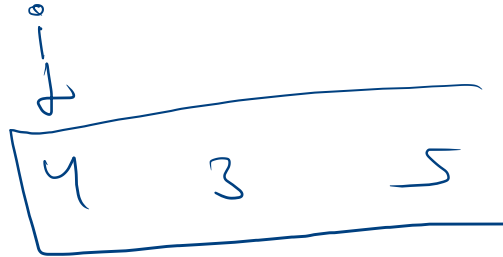
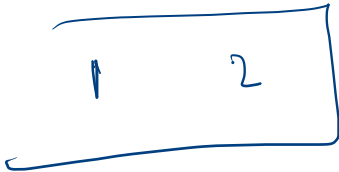
~~2~~
4

~~3~~
3

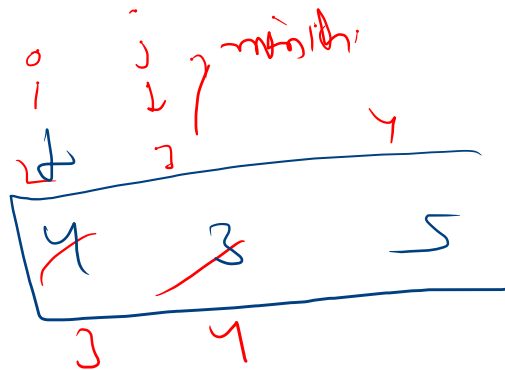
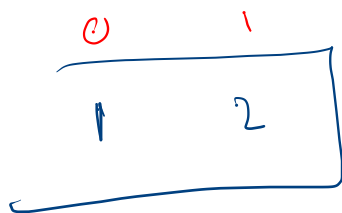
5

$\text{min index} = i = 1$

and



3rd iter



$i = n-1$

min id = 2

end

