

T.C.  $\rightarrow O(n^2)$

S.C.  $\rightarrow O(1)$

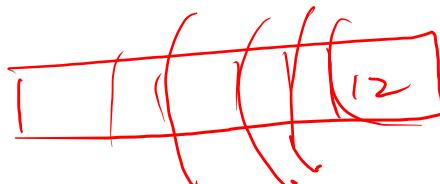
## Searching & Sorting

↳ Bubble sort



$n$  elements

$(n-1)$  iterations



## 2) Selection Sort → ascending order

↳ sort minimum value - element first

$(n-1)$  → ①

$n$  elements  
↳  $(n-1)$  iterations

5	-1	-7	9	3
0	1↑	2↑	3	4

i+ $\tau$ -03

-7	-1	3		
----	----	---	--	--

i+ $\tau$ -01

-7				
----	--	--	--	--

i+ $\tau$ -04

-7	-1	3	5	
----	----	---	---	--

i+ $\tau$ -02

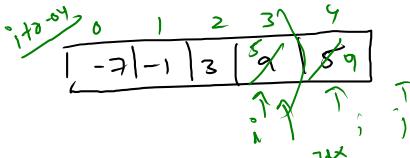
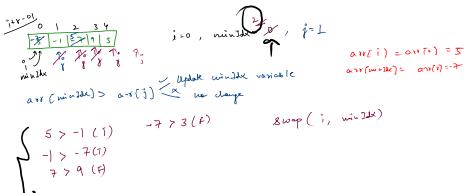
-7	-1		
----	----	--	--

0 1 2 3 4  
i = 1 -7 9 3

$i = \text{which index position decreasing we are taking}$

$j = \text{for iterating over the elements \& always start from } i+1$

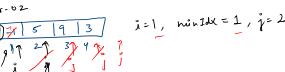
$\min\text{Idx} = \text{in a particular iteration, who is the index of min value element}$



$$\begin{aligned} i &= 3 \\ \text{minIdx} &= 3 \\ j &= 4 \end{aligned}$$

$\Rightarrow n \rightarrow (n-1)$

$$9 > 5 (\text{T})$$

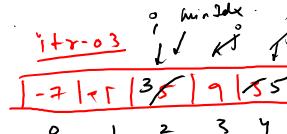


$\text{arr}(\min\text{Idx}) > \text{arr}(i) \Rightarrow \text{arr}(i)$

$$-1 > 5 (\text{F})$$

$$-1 > 9 (\text{F})$$

$$-1 > 3 (\text{F}) \quad \text{swap}(i, \min\text{Idx})$$



$$0 \quad 1 \quad 2 \quad 3 \quad 4$$

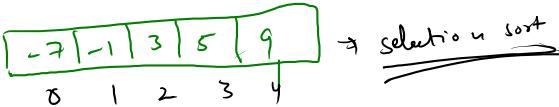
$$i = 2$$

$$\frac{\min\text{Idx}}{j = 3} = 2 \quad 4$$

$$j = 3$$

$$5 > 9 (\text{F})$$

$$5 > 3 (\text{T})$$



$\rightarrow \text{min element first}$

$\rightarrow i, \min\text{Idx}, j$

which index position we are taking

store index of min element in a particular iteration

$\rightarrow \text{swap}(i, \min\text{Idx})$

Tues, Wed, Thurs

→ 1. ~~Imp.~~

Feedback form

→ 2. Tomorrow → Masterclass 8:30 pm

please → 6. assessment

$1 \rightarrow n-1$   
 $2 \rightarrow n-2$   
 $3 \rightarrow n-3$   
 $4 \rightarrow n-4$   
 ...  
 1  
 ...  
 2

```

4 public class Solution {
5   public static void swap(int arr[], int i, int j){
6     int temp = arr[i];
7     arr[i] = arr[j];
8     arr[j] = temp;
9   }
10  public static void selectionSort(int arr[], int n){
11    for(int i=0;i<n-1;i++){
12      int minIdx = i; // Initially
13      for(int j=i+1;j<n;j++){
14        if(arr[minIdx] > arr[j]){
15          minIdx = j;
16        }
17      }
18      swap(arr,i,minIdx);
19    }
20
21  public static void main(String[] args) {
22    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class
23    Scanner scn = new Scanner(System.in);
24    int n = scn.nextInt();
25    int arr[] = new int[n];
26    for(int i=0;i<n;i++){
27      arr[i] = scn.nextInt();
28    }
29
30    selectionSort(arr,n);
31
32
33    for(int i=0;i<n;i++){
34      System.out.print(arr[i] + " ");
35    }
36  }
37}
  
```

$5-6 \text{ min}$   
 Dry run

$$S.C \Rightarrow O(1)$$

$$T.C \Rightarrow O(n^2)$$

$n(n-1) \times n-2 \times n-3 \times \dots \times 2$

$$\Rightarrow \frac{n(n-1)}{2} = O(n^2)$$

### 3) Insertion Sort :-

↳ Arranging deck of cards.

0	1	2	3	4	5
+7	-2	5	70	-9	8



7
---

-1	5	70	-9	8
----	---	----	----	---

-1	7
----	---

5	70	-9	8
---	----	----	---

-1	5	7
----	---	---

70	-9	8
----	----	---

-1	5	7	70
----	---	---	----

-9	8
----	---

-9	-1	5	7	70
----	----	---	---	----

8
---

0	1	2	3	4	5
-9	-1	5	7	8	70



$i=0$

0	1	2	3
3	7	8	-1

$j = 1$

$$anc[i] > anc[i+1]$$

$swap(j, j+1)$

$9 > 7$

$i=1$

0	1	2	3
7	8	9	-1

$j = 1$

$$9 > 8 \text{ (T)}$$

$\leftarrow 7 > 8 \text{ (F)}$

break the loop

$i=2$

9
---

7	8	-1
---	---	----

$i=3$

7	9
---	---

$i=4$

8	-1
---	----

$i=0$

0	1	2	3
-1	7	8	-1

$j = 0$

$$9 > -1 \text{ (T)}$$

$$8 > -1 \text{ (T)}$$

$$7 > -1$$

7	8	9
-1	7	8

$i=1$

-1
----

$i=2$

0
---

$i=3$

-1	7	8	9
----	---	---	---

$n \rightarrow \text{elements}$

$\Rightarrow (n-1) \text{ iterations}$

Array Sort  
 ↗ Java  
 ↗ Insertion Sort  
 ↗ Merge Sort  
 ↗ Quick Sort

```

public class Solution {
    public static void swap(int arr[], int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    public static void insertionSort(int arr[], int n) {
        for(int i=1; i<n; i++) {
            for(int j=i-1; j>=0; j--) {
                if(arr[j] > arr[j+1]) {
                    swap(arr, j, j+1);
                } else {
                    break; // Why breaking the loop? -> Because element already at it's correct
                }
            }
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];
        for(int i=0; i<n; i++) {
            arr[i] = scn.nextInt();
        }

        insertionSort(arr, n);

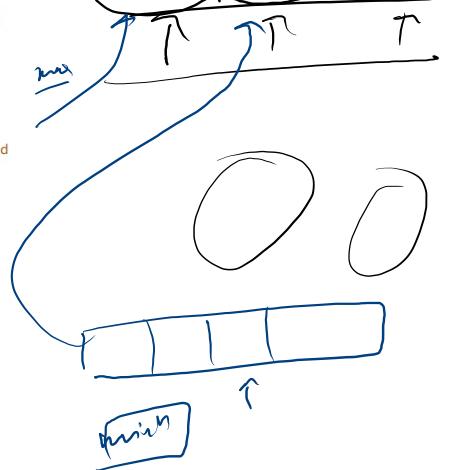
        for(int i=0; i<n; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
  
```

$i = 1 \quad 0^{\text{--}}$   
 $i = 2 \quad 1^{\text{--}}$   
 $i = 3 \quad 2^{\text{--}}$   
 $i = 4 \quad 3^{\text{--}}$   
 $\frac{n(n+1)}{2}$

$S \cdot C = O(1)$

$T \cdot C = O(n^2)$

↳ Bubble Sort / Selection Sort / Insertion Sort



Merge Two Sorted Arrays (Day 31)

Problems Submissions Leaderboard Discussions

- You are given two sorted arrays of integers.
- You have to merge them and form one sorted array.
- You have to do it in linear time complexity.

$n_1 = 4$

0	1	2	3
-2	1	5	9
1	2		

$n_2 = 3$

$b$

4	2	0
0	1	2

$\Rightarrow$

0	1	2	3	4	5	6
-2	1	5	9	8	10	11

Ap-4

-2	5	9	11	4	6	8
----	---	---	----	---	---	---

Bubble  
solution  
method

-2	4	5	6	8	9	11
----	---	---	---	---	---	----

$n$        $i = 0, j = 0$

0	1	2	3
-2	5	9	11

0	1	2
4	6	8

while ( $i < n_1 \text{ and } j < n_2$ )

$f(a[i]) > f(b[j]) \rightarrow \text{False}$

$ans[k] = b[j]$

$j++$ ,  $k++$

$ans[k] = a[i]$

$i++$ ,  $k++$

while ( $j < n_2$ )

$ans(k) = b[j]$

$j++$

while ( $i < n_1$ )

$ans(k) = a[i]$

$i++$

$-2 > 4$  (f)

$5 > 4$  (t)

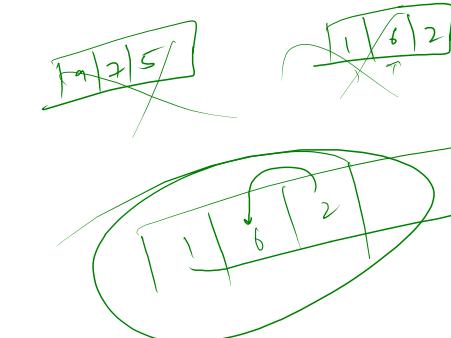
$5 > 6$  (f)

$T.C \geq O(\max(n_1, n_2))$

$S.C \geq O(n_1 + n_2)$

$9 > 6$  (t)

$9 > 8$  (t)



*Friday*

```
public class Solution {
    public static void mergeTwoSortedArrays(int a[],int b[],int n1,int n2){
        int res[] = new int[n1+n2];
        int i=0,j=0,k=0;

        while(i<n1 && j<n2){
            if(a[i] > b[j]){
                res[k] = b[j];
                k++;
                j++;
            }else{
                res[k] = a[i];
                i++;
                k++;
            }
        }

        while(i<n1){
            res[k] = a[i];
            i++;
            k++;
        }

        while(j<n2){
            res[k] = b[j];
            j++;
            k++;
        }

        for(i=0;i<res.length;i++){
            System.out.println(res[i]);
        }
    }
}
```

- ⇒ Classwork
- ⇒ Homework
- ⇒ Fill the feedback from
- ⇒ Masterclass [