Module 2

- Theory
- Problems
  1) Questions
  2) Logic Building
  3) Dry Run
  4) Coding

⇒ Sorting (arranging elements in a particular order)

(algorithms)

- Bubble sort
- Selection sort
- Insertion sort

$O(N^2)$

where N is the size of array

⇒ Bubble sort

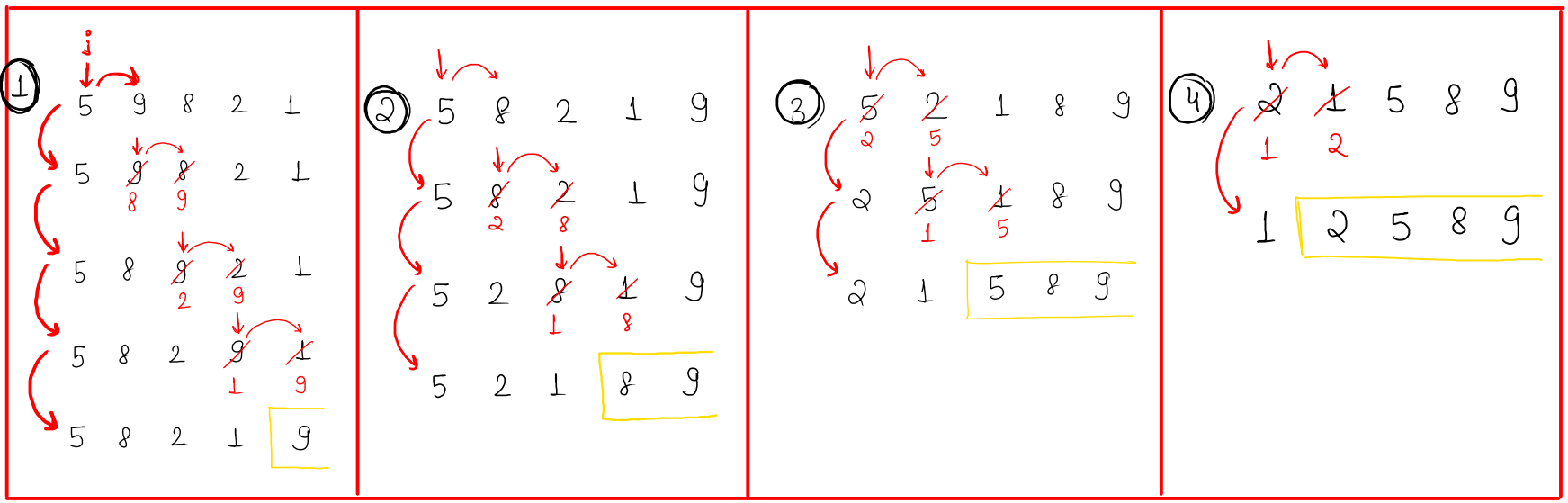
↳ Pick the largest element and place it to the rightmost side of unsorted part of array

Dry run

arr =

0	1	2	3	4
5	9	8	2	1

n = 5



note:- n = 5

i = 0 → 4
i = 1 → 3
i = 2 → 2
i = 3 → 1

} j

j = n − i − 1

Psudo code

n = 5

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

j ⟶ myself

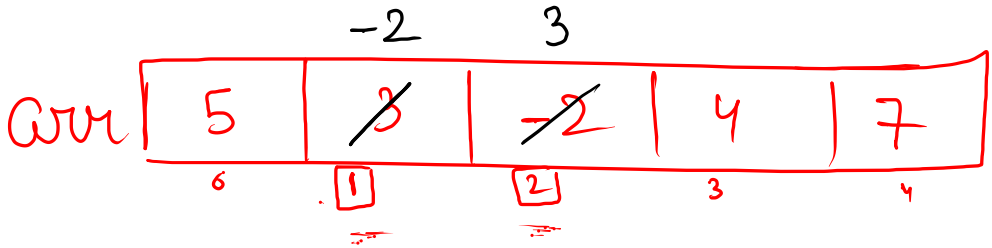
(j+1) ⟶ other

Permutation

```java
public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

1 2

temp = 3




```
int temp = arr[x];
arr[x] = arr[y]
arr[y] = temp;
```

a, b

```
int temp = a;
a = b;
b = temp;
```

Code

```java
public static void main(String[] args) {
    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();
    }

    bubbleSort(arr, n);
}
public static void bubbleSort(int[] arr, int n) {
    //logic
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if ( arr[j] > arr[j + 1] ) {
                swap(arr, j, j + 1);
            }
        }
    }
    //printing
    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}

public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

⇒ Insertion sort

(pick the first element of unsorted array and place it at the correct position)

arr = | 0: 5 | 1: 9 | 2: 8 | 3: 2 | 4: 1 | , n = 5