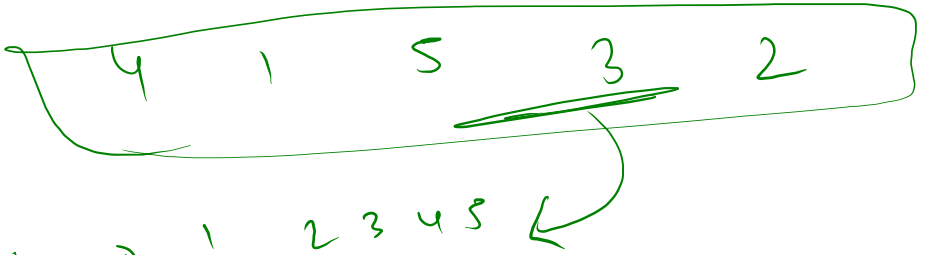
Sorting → arranging elements in some order.



increasing → 1 2 3 4 5

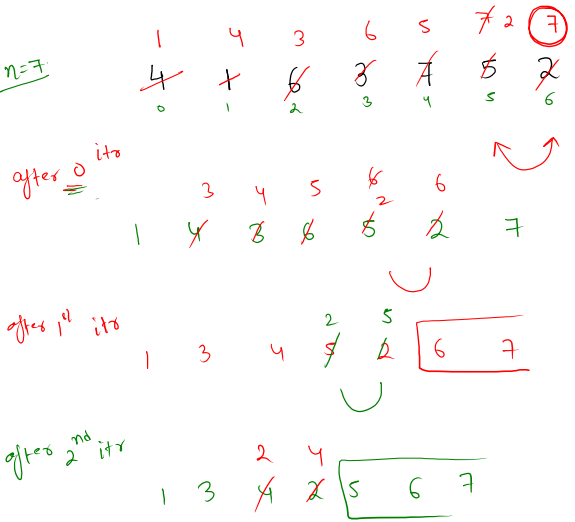
decreasing → 5 4 3 2 1

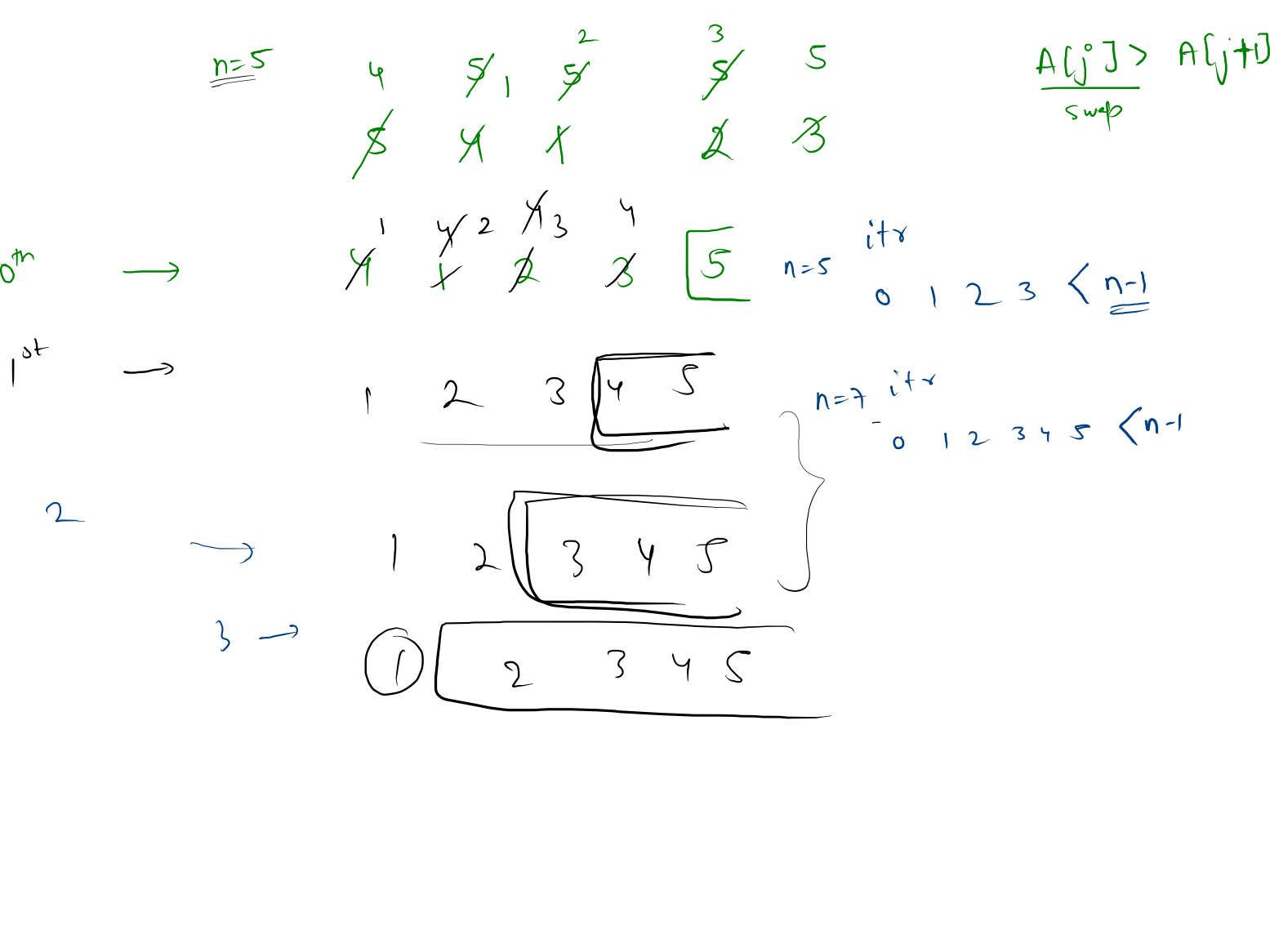
odd... even... 1 5 3 4 2

Bubble Sorting.

"1 itr largest number will be at its correct position."
one

n=7




n=5
A[j] > A[j+1]
swap
0th
1st
2
3
n=5 itr
0 1 2 3 < n-1
n=7 itr
0 1 2 3 4 5 < n-1

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }


        for(int i = 0; i < n-1; i++){
            for(int j = 0; j < n-1-i ; j++){
                if(A[j] > A[j+1]){
                    int tmp = A[j];
                    A[j] = A[j+1];
                    A[j+1] = tmp;
                }
            }
        }

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

```

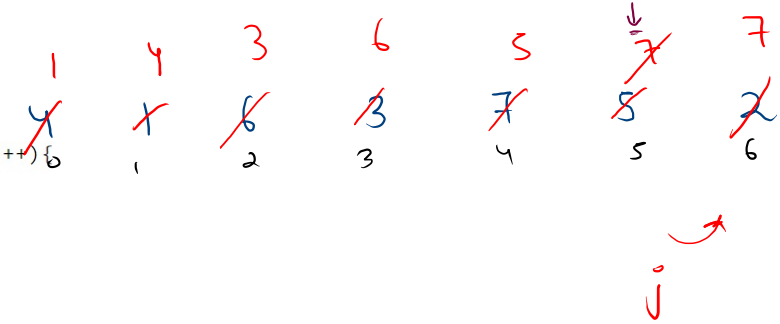
$j < n-i-1$

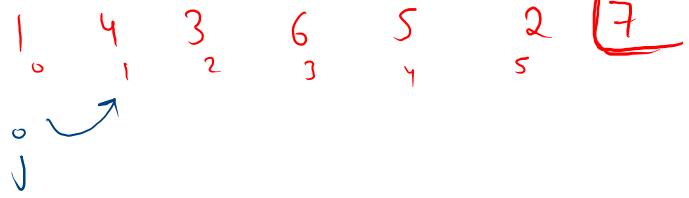
$j < n-1$

n=7

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

i=0





i=1

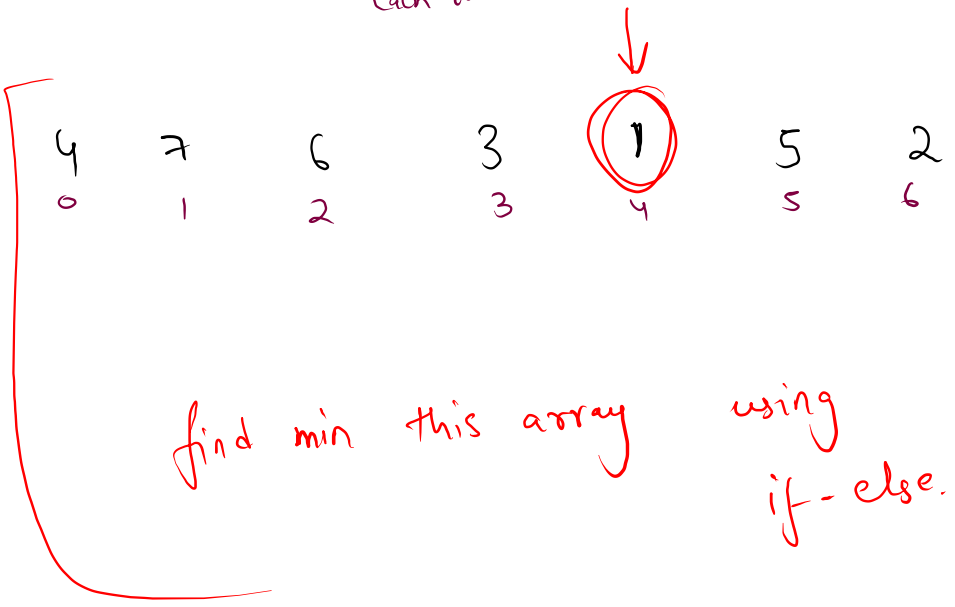
Selection Sort

"smallest ele. at correct position"

each itr

asc

inc.



itr

↓

n-1

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }

        //logic: selection sort
        for(int i = 0; i < n-1; i++){   //i = 0
            int minIdx = i;

            for(int j = i + 1; j < n; j++){
                if(A[j] < A[minIdx]){
                    minIdx = j;
                }
            }

            int tmp = A[i];
            A[i] = A[minIdx];
            A[minIdx] = tmp;

        }

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

    }
}
```

```
14      //logic: selection sort
15      for(int i = 0; i < n-1; i++){   //i = 0
16          int minIdx = i;
17
18          for(int j = i + 1; j < n; j++){
19              if(A[j] < A[minIdx]){
20                  minIdx = j;
21              }
22          }
23
24          int tmp = A[i];
25          A[i] = A[minIdx];
26          A[minIdx] = tmp;
27
28      }
29
```

Swap (i, mI)

n = 6

2	5	1	4	6	3
0	1	2	3	4	5

i = 0

mI = 2

A[3] < A[2]

4 < 1 ✗

A[5] < A[2]

3 < 1

A[4] < A[2]

6 < 1

Default behaviour → Java → increasing order

sort → inbuilt () → Java.

```java
//import java.util.*;
import java.util.Arrays;
public class Main
{
    public static void main(String[] args) {
        int [] A = {4,7,6,3,1,5,2};


        Arrays.sort(A);



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