```
for (int i = Count_zero + Count_one; i<n; i++){
        arr[i] = 2;
}
```

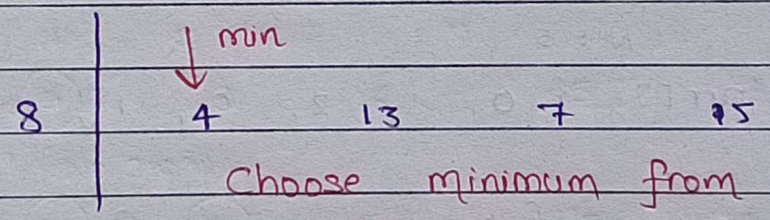
(Lecture - 10)

(Sorting : Basic)

* Selection Sort :

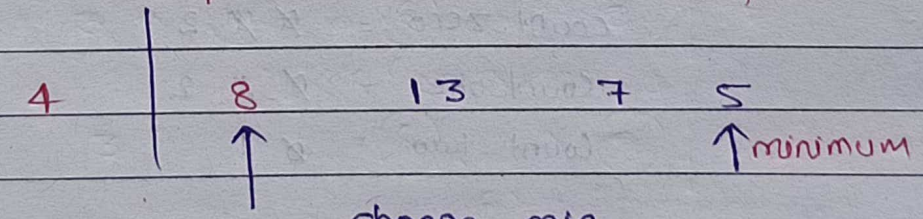
arr : {8, 4, 13, 7, 5}

output : 4 5 7 8 13



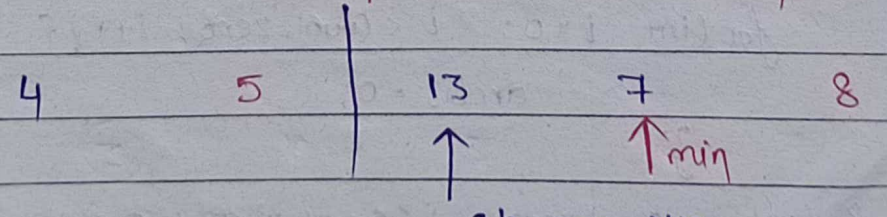
Choose minimum from this side

Swap this with 1st element



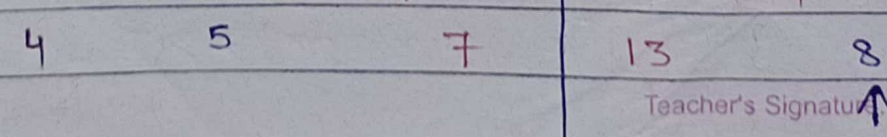
choose min

Swap with 2nd element.



Choose Min

Swap with 3rd element.



Teacher's Signatu

chose minimum

swap with 4th element.

4 5 7 8 13

↳ final result.

Code:

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

* Bubble sort

arr[]: { 7 2 4 3 6 5 }

In this we compare two value and swap it

In every pass one element goes at its right place.

The last element goes at right place

We have do (n-1) pass to complete this.

Teacher's Signature.....

Page No.:
Date : / /

pass 1:

(7) (2) 4 3 6 5

7 > 2 (swap)

2 (7) (4) 3 6 5

7 > 4 (swap)

2 4 (7) (3) 6 5

7 > 3 (swap)

2 4 3 (7) (6) 5

7 > 6 (swap)

2 4 3 6 (7) (5)

7 > 5 (swap)

2 4 3 6 5 | 7

↳ got its right place.

pass 2:

(2) (4) 3 6 5 | 7

2 < 4 (No swap)

2 (4) (3) 6 5 | 7

4 > 3 (swap)

2 3 (4) (6) 5 | 7

4 < 6 (No swap)

2 3 4 (6) (5) | 7

6 > 5 (swap)

2 3 4 5 | 6 7

pass 3:

(2) (3) 4 5 | 6 7

2 < 3 (No swap)

2 (3) (4) 5 | 6 7

3 < 4 (No swap)

Teacher's Signature.....

2 3 (4) (5) | 6 7

4 < 5 (No swap)

pass 4:

(2) (3) 4 | 5 6 7

2 < 3 (No swap)

2 (3) (4) | 5 6 7

3 < 4 (No swap)

pass 5:

(2) (3) | 4 5 6 7

2 < 3 (No swap)

2 3 4 5 6 7

↳ sorted

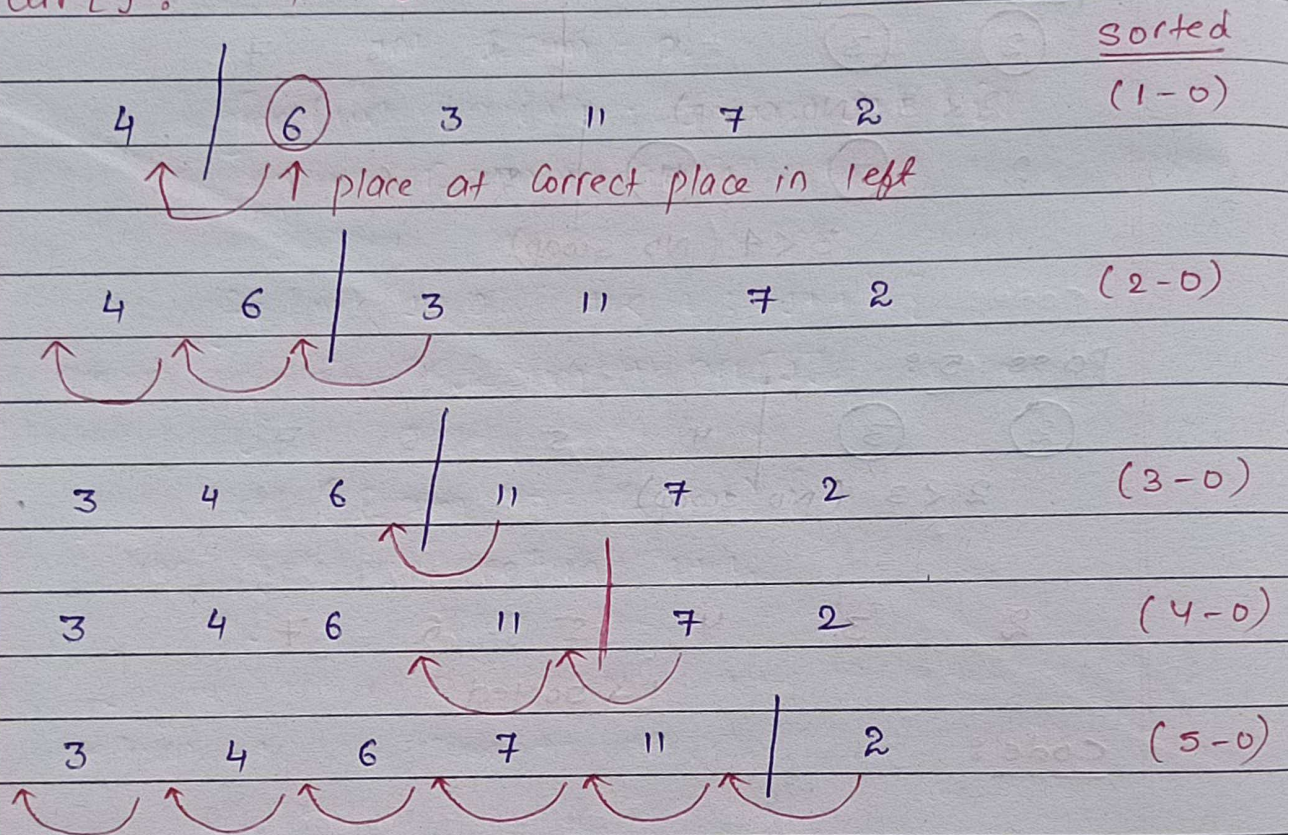
Code:

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

Teacher's Signature.....

* Insertion Sort :

arr[] : 4 6 3 11 7 2



2 3 4 6 7 11

↳ sorted.

Code :

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

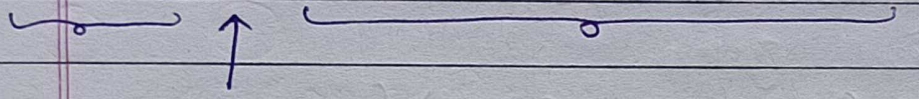
* Problem : Equilibrium point

The left side subarray and right side subarray of an element is equal, then the element is known as equilibrium point:

Example:

→

1	2	3	4	5
1	3	5	2	2



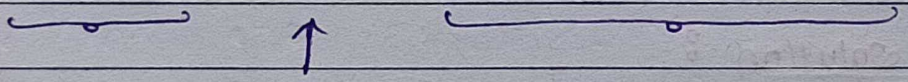
l_sum = 0 right = 3 + 5 + 2 + 2 = 12

l_sum != right

1 is not Equilibrium point

→

1	3	5	2	2



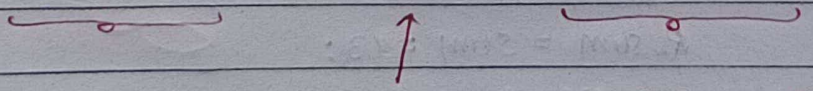
l_sum = 1 right = 5 + 2 + 2 = 9

l_sum != right

3 is not Equilibrium point

→

1	3	5	2	2



l_sum = 1 + 3 = 4 right = 2 + 2 = 4

l_sum = right

5 is the Equilibrium point.

Teacher's Signature.....

Code : (Brute force) T.C : $O(N^2)$

```
for (int i = 0;  i < n;  i++) {
        int r_sum = 0;
        int l_sum = 0;
    for (int j = i+1 ; j < n; j++ ) {
            r_sum += arr[j];
    }
    for (int j = 0;  j < i ; j++) {
            l_sum += arr[j];
    }
    if (l_sum == r_sum) {
            return i+1;
    }
}
return -1;
```

Optimal solution :

1	3	5	2	2

calculate sum

sum = 13

R_sum = sum = 13;

→

1	3	5	2	2
↑				

R_sum = 13 - 1 = 12

L_sum = 0

> Not Equal.

Teacher's Signature.....

1 3 5 2 2

↑

R_sum = 12 - 3 = 9
L_sum = 1
> Not Equal

1 3 5 2 2

↑

R_sum = 9 - 5 = 4
L_sum = 4
> Equal

Equillibrium point.

Time complexity : O(N)

Code :

```
if(n==1)
    return 1;
long totalSum = 0;
int res = -1;
long sum = 0;
for(int i=0; i<n; i++)
    totalSum += arr[i];

for(int i=1; i<n; i++) {
    sum += arr[i-1];

    if(totalSum - sum - arr[i] == sum) {
        res = i;
        break;
    }
}
return res == -1 ? -1 : res+1;
}
```