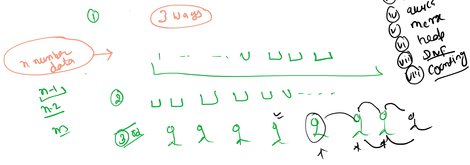
Lec-9
26/3/24 4:46 PM

Sorting → Arrangement → inc or dec

(i) bubble
(ii) selection
(iii) insertion
(iv) quick
(v) merge
(vi) heap
(vii) Dsc counting

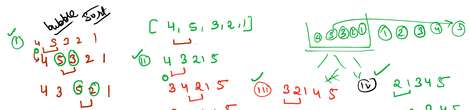
n number data →

n-1
n-2
n



bubble sort

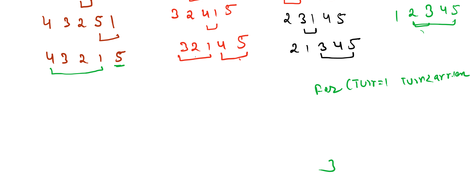
[4, 5, 3, 2, 1]



for (turn=1 turn<arr.lon turn++)

```
public static void sort(int []arr) {
    for (int turn = 1; turn < arr.length; turn++) {
        for(int i=0; i<_ i++) {
            if(arr[i]>arr[i+1]) {
                int temp=arr[i];
                arr[i]=arr[i+1];
                arr[i+1]=temp;
            }
        }
    }
}
```

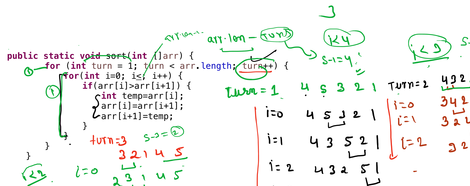
arr.lon − turn



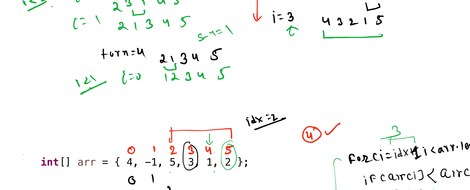
```
int[] arr = { 4, -1, 5, 3, 1, 2};
```

for(i=idx+1 i<arr.lon i++)
if(arr[i] < arr[mini]) {
mini=i
}

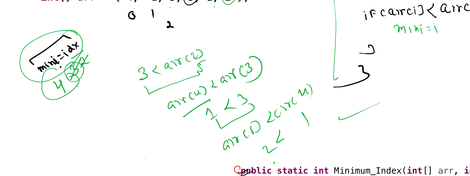
```
public static int Minimum_Index(int[] arr, int idx) {
    int mini = idx;
    for (int i = idx + 1; i < arr.length; i++) {
        if (arr[i] < arr[mini]) {
            mini = i;
        }
    }
    return mini;
}
```



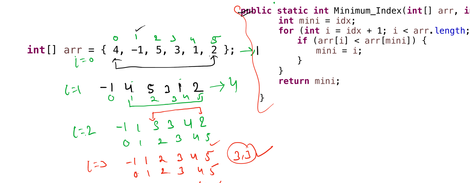
```
public static void sort(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        int idx = Minimum_Index(arr, i);
        int temp = arr[i];
        arr[i] = arr[idx];
        arr[idx] = temp;
    }
}
```



```
public static void Insert_Last_Element(int[] arr, int i) {
    int v = arr[i];
    int j = i - 1;
    while (j >= 0 && arr[j] > v) {
        arr[j + 1] = arr[j];
        j--;
    }
}
```



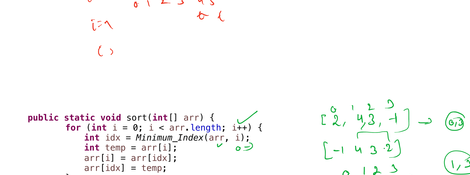
Subarray



for(i=0 i<arr.lon i++)
int sum=0
for(j=i j<arr.lon j++)
sum=sum+arr[j]
ans=max(ans,sum)

```
public static int MaximumSubArraySum(int[] arr) {
    int ans = Integer.MIN_VALUE;
    int sum = 0;
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
        ans = Math.max(ans, sum);
    }
    return ans;
}
```

if(sum<0) {
sum=0
}



int e=arr[0] vote=1

for(i=1 i<arr.lon i++)
if(e==arr[i]) vote++
else vote--
if(vote==0) e=arr[i] vote=1