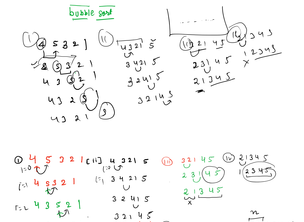
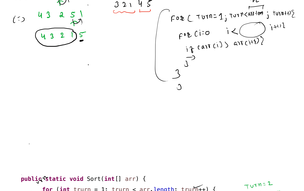
Lec-9
01/06/24 6:00 PM

Sorting

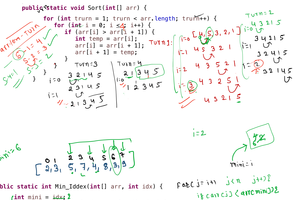
- (i) bubble Sort
- (ii) Selection Sort
- (iii) Insertion Sort
- (iv) DNF
- (v) Merge Sort
- (vi) Quick Sort
- (vii) Heap Sort
- (viii) Counting Sort

bubble Sort

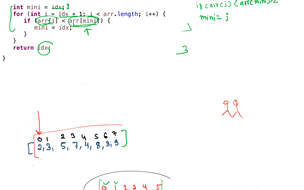




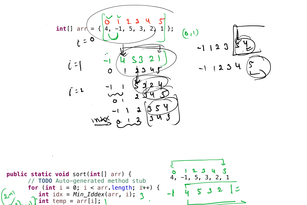
```
for (int trum = 1; trum < arr.length; trum++) {
    for (int i = 0; i<n-1; i++) {
        if (arr[i] > arr[i+1]) {
            int temp = arr[i];
            arr[i] = arr[i+1];
            arr[i+1] = temp;
        }
    }
}
```



```
public static void Sort(int[] arr) {
    for (int trum = 1; trum < arr.length; trum++) {
        for (int i = 0; i<n-1; i++) {
            if (arr[i] > arr[i+1]) {
                int temp = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = temp;
            }
        }
    }
}
```

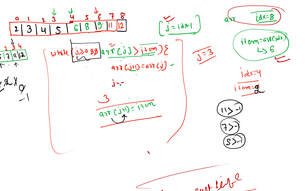


```
public static int Min_Index(int[] arr, int idx) {
    int mini = idx;
    for (int i = idx+1; i < arr.length; i++) {
        if (arr[i] < arr[mini]) {
            mini = i;
        }
    }
    return idx;
}
```

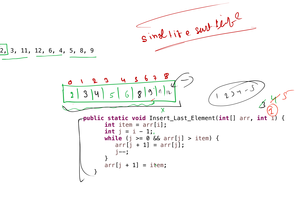


```
int[] arr = {4, -1, 5, 3, 3, 2, 1};
```

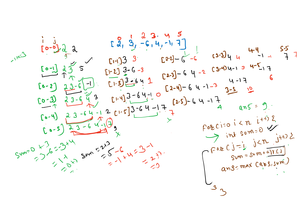
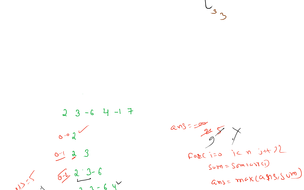
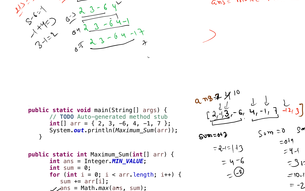
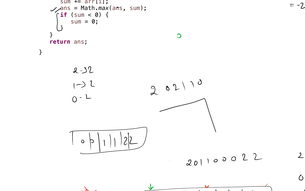
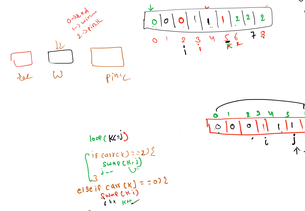
```
public static void sort(int[] arr) {
    // TODO Auto-generated method stub
    for (int i = 0; i < arr.length; i++) {
        int idx = Min_Index(arr, i);
        int temp = arr[i];
        arr[i] = arr[idx];
        arr[idx] = temp;
    }
}
```



```
while (j>=0 && arr[j] > item) {
    arr[j+1] = arr[j];
    j--;
}
arr[j+1] = item;
```



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



```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int[] arr = {2, 3, -6, 4, -1, 7};
    System.out.println(Maximum_Sum(arr));
}

public static int Maximum_Sum(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);
        if (sum < 0) {
            sum = 0;
        }
    }
    return ans;
}
```