

Lec-27 06/02/25 2:59 PM

```
int[] arr1 = { 2, 3, 5, 7, 8 };
int[] arr2 = { 1, 3, 5, 8, 10, 11, 13 };
```

Handwritten array:  $\{1, 2, 3, 3, 5, 5, 7, 8, 8, \dots, 11, 13\}$

```
public static int[] Merge_Two_Array(int[] arr1, int[] arr2) {
    int n = arr1.length;
    int m = arr2.length;
    int[] ans = new int[n + m];
    int i = 0, j = 0, k = 0;
    while (i < n && j < m) {
        if (arr1[i] < arr2[j]) {
            ans[k] = arr1[i];
            k++;
            i++;
        } else {
            ans[k] = arr2[j];
            k++;
            j++;
        }
    }
}
```

Handwritten diagrams showing merging of two sorted arrays [2, 3, 5, 7, 8] and [1, 3, 5, 8, 10, 11, 13] into a result array.

```
int[] arr = { 5, 7, 2, 1, 8, 3, 4 };
```

Handwritten notes:  $12, 3, 4, 5, 7, 8$  with  $10 \log n$  and  $\log \leq 4 < \text{right}$ . Includes a tree diagram for binary search.

Handwritten diagrams showing recursive steps for finding the maximum element in an array, including a tree structure for the recursive calls.

```
public static int[] Sort(int[] arr, int si, int ei) {
    if (si == ei) {
        int[] bs = new int[1];
        bs[0] = arr[si];
        return bs;
    }
    int mid = (si + ei) / 2;
    int[] a = Sort(arr, si, mid);
    int[] b = Sort(arr, mid + 1, ei);
    return Merge_Two_Array(a, b);
}
```

Handwritten diagrams showing the recursive sorting process and a table for the recursive steps.

Handwritten notes:  $2 \log \leq 4 < \text{right}$  and  $\log \leq 4 < \text{right}$ . Includes a tree diagram for binary search.

```
public static void Sort(int[] arr, int si, int ei) {
    // TODO Auto-generated method stub
    if (si >= ei) {
        return;
    }
    int idx = Partition(arr, si, ei);
    Sort(arr, si, idx - 1);
    Sort(arr, idx + 1, ei);
}
```

Handwritten diagrams showing the recursive sorting process and a tree structure for the recursive calls.

Handwritten notes:  $N \rightarrow \text{prime } x$  and  $N = a \times b$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .

sieve of eratosthenes

Handwritten diagram showing the sieve of Eratosthenes algorithm and a table for the recursive steps.

Handwritten notes:  $N = 10^5$  and  $\log \log N$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .

```
public static int PrimeSieve(int n) {
    int[] ans = new int[n];
    ans[0] = ans[1] = 1;
    for (int i = 2; i * i <= ans.length; i++) {
        if (ans[i] == 0) {
            for (int j = 2; i * j <= ans.length; j++) {
                ans[i * j] = 1;
            }
        }
    }
}
```

Handwritten diagrams showing the prime sieve algorithm and a table for the recursive steps.

Handwritten notes:  $0 \leq n \leq 10^5$  and  $2 \rightarrow \text{sub } 8 \text{ min}$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .

Handwritten notes:  $0 \leq n \leq 10^5$  and  $2 \rightarrow \text{sub } 8 \text{ min}$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .

Handwritten notes:  $0 \leq n \leq 10^5$  and  $2 \rightarrow \text{sub } 8 \text{ min}$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .

Handwritten notes:  $0 \leq n \leq 10^5$  and  $2 \rightarrow \text{sub } 8 \text{ min}$ . Includes a diagram showing the relationship between  $N$ ,  $a$ , and  $b$ .