Topics

Lecture 2 : Arrays & Strings

Deadline

Sep 28, 2020, 11:59 PM

- Code : Find duplicate 0/40
- Code : Merge two sorted arrays 0/80
- Code : Maximise the sum 0/80
- Introduction to char arrays
- cstring header file functions
- Code : Check Permutation 0/40
- Code : Reverse each word 0/80
- Code : Compress the string 0/80
- Introduction to 2D arrays
- Find an element
- Code : Rotate matrix 0/80
- Vectors
- Vectors 0/10
- Insert element 0/10
- What is the output 0/10
- What is the output 0/10
- What is the output 0/10
- What is the output 0/10
- string class

Assignment

0.0%

Score 0/800

- Sort 0 1 2 0/80
- Push Zeros to end 0/80

Your Code | Main Code | Code Pair | C++ (g++ 5

Problem | Result

Code : Maximise the sum

Send Feedback

Given 2 sorted arrays (in increasing order), find a path through the intersections that produces maximum sum and return the maximum sum.

That is, we can switch from one array to another array only at common elements.

If no intersection element is present, we need to take sum of all elements from the array with greater sum.

Input Format :

Line 1 : An integer M i.e. size of first array
Line 2 : M integers which are elements of first array, separated by spaces
Line 3 : An integer N i.e. size of second array
Line 4 : N integers which are elements of second array, separated by spaces

Output Format :

Maximum sum value

Constraints :

1 <= M, N <= 10^6

Sample Input :

```
6
1 5 10 15 20 25
5
2 4 5 9 15
```

Sample Output :

```
81
```

Explanation :

We start from array 2 and take sum till 5 (sum = 11). Then we'll switch to array at element 10 and take till 15. So sum = 36. Now, no elements left in array after 15, so we'll continue in array 1. Hence sum is 81

```
long maxPathSum(i
{
    /* Don't writ
     Don't read i
     Return outpu
     Taking input
     */


}
```

PREVIOUS | NEXT | CUSTOM INPUT | SUBMIT SOLUTI