Second Largest | Practice | Geek

geeksforgeeks.org/batch/gfg-160-problems/track/arrays-gfg-160/problem/second-largest3735

Courses | Placement | IBM Data Science | GATE | Search...

Problem | Editorial | Submissions

Given an array of **positive** integers **arr[]**, return the **second largest** element from the array. If the second largest element doesn't exist then return **-1.**

Note: The second largest element should not be equal to the largest element.

Examples:

Input: arr[] = [12, 35, 1, 10, 34, 1]
Output: 34
Explanation: The largest element of the array is 35 and the second largest element is 34.

Input: arr[] = [10, 5, 10]
Output: 5
Explanation: The largest element of the array is 10 and the second largest element is 5.

Input: arr[] = [10, 10, 10]
Output: -1
Explanation: The largest element of the array is 10 and the second

C++ (12) | Start Timer

```cpp
class Solution {
  public:
    int getSecondLargest(vector<int> &arr) {
        // code here
        int max=-1;
        int sec= -1;
        int len= arr.size();

        for(int i=0;i<len;i++)
        {
            if(arr[i]>max)
            {
                sec=max;
                max=arr[i];
            }
            else if (arr[i]>sec && arr[i]<max)
            {
                sec=arr[i];
            }
        }

        return sec;
    }
};
```

Custom Input | Compile & Run | Submit

Problem | Editorial | Submissions

Reverse an Array

Difficulty: **Easy** Accuracy: **55.32%** Submissions: **225K+** Points: **2**

Average Time: **5m**

You are given an array of integers **arr[]**. You have to **reverse** the given array.

Note: Modify the array in place.

Examples:

Input: arr = [1, 4, 3, 2, 6, 5]
Output: [5, 6, 2, 3, 4, 1]
Explanation: The elements of the array are [1, 4, 3, 2, 6, 5]. After reversing the array, the first element goes to the last position, the second element goes to the second last position and so on. Hence, the answer is [5, 6, 2, 3, 4, 1].

Input: arr = [4, 5, 2]
Output: [2, 5, 4]
Explanation: The elements of the array are [4, 5, 2]. The reversed

C++ (12) Start Timer

```cpp
class Solution {
  public:
    void reverseArray(vector<int> &arr) {
        // code here
        int len=arr.size();
        for(int i=0;i<(len/2);i++)
        {
            int temp=arr[i];
            arr[i]=arr[len-1-i];
            arr[len-1-i]=temp;
        }
    }
};
```

Custom Input Compile & Run Submit

Move All Zeroes to End

Difficulty: **Easy** Accuracy: **45.51%** Submissions: **378K+** Points: **2**

Average Time: **15m**

You are given an array **arr[]** of non-negative integers. You have to move all the zeros in the array to the right end while maintaining the relative order of the non-zero elements. The operation must be performed **in place**, meaning you should not use extra space for another array.

Examples:

Input: arr[] = [1, 2, 0, 4, 3, 0, 5, 0]
Output: [1, 2, 4, 3, 5, 0, 0, 0]
Explanation: There are three 0s that are moved to the end.

Input: arr[] = [10, 20, 30]
Output: [10, 20, 30]
Explanation: No change in array as there are no 0s.

C++ (12) Start Timer

```cpp
class Solution {
  public:
    void pushZerosToEnd(vector<int>& arr) {
        // code here
        int j=0;
        int len=arr.size();
        int i=0;
        while(i<len)
        {
            if(arr[i]!=0)
            {
                arr[j]=arr[i];
                j++;
            }
            i++;

        }
        while(j<len)
            {
                arr[j]=0;
                j++;
            }
    }
};
```

Custom Input Compile & Run Submit