Subarray with 0 sum

Easy Accuracy: 39.79% Submissions: 176K+ Points: 2

Given an array of positive and negative numbers. Find if there is a **subarray** (of size at-least one) with **0 sum**.

Example 1:

Input:
5
4 2 -3 1 6

Output:
Yes

Explanation:
2, -3, 1 is the subarray with sum 0.

C++ (g++ 5.4) Average Time: 20m Start Timer

```cpp
// } Driver Code Ends
class Solution{
    public:
    //Complete this function
    //Function to check whether there is a subarray present with 0-sum or not.
    bool subArrayExists(int arr[], int n)
    {
        //Your code
        unordered_set<int>s;
        int sum = 0;
        for(int i = 0;i<n;i++){
            sum += arr[i];
            if (s.find(sum) != s.end())
                return true;
            if(sum == 0)
                return true;
            s.insert(sum);
        }
        return false;
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Largest subarray with 0 sum

Easy Accuracy: 41.84% Submissions: 182K+ Points: 2

Given an array having both positive and negative integers. The task is to compute the length of the largest subarray with sum 0.

Example 1:

Input:
N = 8
A[] = {15,-2,2,-8,1,7,10,23}
Output: 5
Explanation: The largest subarray with sum 0 will be -2 2 -8 1 7.

Your Task:
You just have to complete the function **maxLen()** which takes two arguments an array **A** and **n,** where n is the size of the array A and returns the length of the largest subarray with 0 sum


```
1  // } Driver Code Ends
9  /*You are required to complete this function*/
10
11 class Solution{
12     public:
13     int maxLen(vector<int>&a, int n)
14     {
15         unordered_map <int,int> mp;
16         int cnt=0,sum=0;
17         for(int i=0;i<n;i++){
18             sum+=a[i];
19             if(sum==0){
20                 cnt=i+1;
21             }else if(mp[sum]>0){
22                 cnt=max(cnt,i-mp[sum]+1);
23             }else{
24                 mp[sum]=i+1;
25             }
26         }
27         return cnt;
28     }
29 };
30
31 // } Driver Code Ends
```

Longest consecutive subsequence

Medium Accuracy: 33.0% Submissions: 219K+ Points: 4

Given an array of positive integers. Find the length of the longest sub-sequence such that elements in the subsequence are consecutive integers, the **consecutive numbers can be in any order.**

Example 1:

Input:
N = 7
a[] = {2,6,1,9,4,5,3}
Output:
6
Explanation:
The consecutive numbers here are 1, 2, 3, 4, 5, 6. These 6


```cpp
// } Driver Code Ends
class Solution{
  public:
    // arr[] : the input array
    // N : size of the array arr[]

    //Function to return length of longest subsequence of consecutive integers.
    int findLongestConseqSubseq(int arr[], int n)
    {
      //Your code here
      sort (arr,arr+n);
          int cnt = 1 , res = 1;

          if(n==0){
              return 0;
          }
          for(int i = 1 ; i<n ; i++){
              if(arr[i] == arr[i-1]+1){
                  cnt++;
              }else if(arr[i]==arr[i-1]){
                  continue;
              }else{
                  cnt=1;
              }
              res = max(res,cnt);
          }
          return res;
    }
};
// } Driver Code Ends
```