**Find Maximum Equal sum of Three Stacks :-**

Medium Accuracy: 48.71% Submissions: 26K+ Points: 4

Given three stacks S1, S2 & S3 of size N1, N2 & N3 respectively, having only Positive Integers. The task is to find the possible equal maximum sum of the stacks with the removal of top elements allowed. Stacks are represented as an array, and the first index of the array represents the top element of the stack.

Example 1:

Input:

N1 = 3, N2 = 4, N3 = 2

S1 = {4,2,3}

S2 = {1,1,2,3}

S3= {1,4}

Output:

5

Explanation:

We can pop 1 element from the 1st stack, and 2

elements from the 2nd stack. Now remaining elements

yield the equal sum of the three stacks, that is 5.

Example 2:

Input:

N1 =2, N2 = 1, N3 = 3

S1 = {4,7}

S2 = {10}

S3 = {1,2,3}

Output:

0

Explanation:

We will never get an equal sum after popping

some elements, so the answer will be 0.

Your Task:  
You don't need to read input or print anything. Your task is to complete the function maxEqualSum() which takes the arrays S1[], S2[], and S3[] and their sizes N1, N2, and N3 as inputs and returns the maximum equal sum we can obtain.

Expected Time Complexity: O(N1+N2+N3)  
Expected Auxiliary Space: O(1)

Constraints:  
1 <= N1, N2, N3 <= 105  
1 <= S1[i], S2[i], S3[i] <= 103  
The sum, N1+N2+N3 doesn't exceed 106

**Code :-**

//{ Driver Code Starts

//Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function Template for C++

class Solution{

public:

int maxEqualSum(int N1,int N2,int N3,vector<int> &S1,vector<int> &S2,vector<int> &S3){

int ans=INT\_MIN;

int i=0, j=0, k=0, sum1=0, sum2=0, sum3=0;

while(i<N1 || j<N2 || k<N3){

if(i<N1) sum1+=S1[i];

if(j<N2) sum2+=S2[j];

if(k<N3) sum3+=S3[k];

i++; j++; k++;

}

int mini = min(sum1, min(sum2, sum3));

int x=N1-1, y=N2-1, z=N3-1, sumx=S1[x], sumy=S2[y], sumz=S3[z];

while(true){

if(sumx<=mini && sumy<=mini && sumz<=mini){

int small = min(sumx, min(sumy, sumz));

if(small==sumx){

x--;

sumx+=S1[x];

}

else if(small==sumy){

y--;

sumy+=S2[y];

}

else if(small==sumz){

z--;

sumz+=S3[z];

}

if(sumx==sumy && sumy==sumz)

ans = max(ans, sumx);

}

else

break;

}

if(ans==INT\_MIN) return 0;

return ans;

}

};

//{ Driver Code Starts.

int main(){

int t;

cin>>t;

while(t--){

int n1,n2,n3;

cin>>n1>>n2>>n3;

vector<int> s1(n1),s2(n2),s3(n3);

for(int i=0;i<n1;i++){

cin>>s1[i];

}

for(int i=0;i<n2;i++){

cin>>s2[i];

}

for(int i=0;i<n3;i++){

cin>>s3[i];

}

Solution ob;

int ans=ob.maxEqualSum(n1,n2,n3,s1,s2,s3);

cout<<ans<<endl;

}

return 0;

}

// } Driver Code Ends

**Time Complexity :- O(n1+n2+n3)**

**Space Complexity :- O(1)**