https://www.hackerrank.com/challenges/equal-stacks

//---CON BUSQUEDA BINARIA--------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

string[] tokens\_n1 = Console.ReadLine().Split(' ');

int n1 = Convert.ToInt32(tokens\_n1[0]);

int n2 = Convert.ToInt32(tokens\_n1[1]);

int n3 = Convert.ToInt32(tokens\_n1[2]);

string[] h1\_temp = Console.ReadLine().Split(' ');

int[] a = Array.ConvertAll(h1\_temp, Int32.Parse);

string[] h2\_temp = Console.ReadLine().Split(' ');

int[] b = Array.ConvertAll(h2\_temp, Int32.Parse);

string[] h3\_temp = Console.ReadLine().Split(' ');

int[] c = Array.ConvertAll(h3\_temp, Int32.Parse);

//int[] a = { 3, 2, 1, 1, 1 };

//int[] b = { 4, 3, 2 };

//int[] c = { 1, 1, 4, 1 };

List<int> x = new List<int>();

List<int> y = new List<int>();

List<int> z = new List<int>();

int sum\_x = 0;

for (int i = a.Length-1; i >=0; i--)

{

sum\_x += a[i];

x.Add(sum\_x);

}

int sum\_y = 0;

for (int i = b.Length-1; i >=0; i--)

{

sum\_y += b[i];

y.Add(sum\_y);

}

int sum\_z = 0;

for (int i = c.Length-1; i >=0; i--)

{

sum\_z += c[i];

z.Add(sum\_z);

}

int max\_altura = 0;

for (int i = x.Count-1; i >=0; i--)

{

if (y.BinarySearch(x[i]) > -1 && z.BinarySearch(x[i]) > -1)

{

max\_altura = x[i];

break;

}

}

Console.WriteLine(max\_altura);

Console.ReadLine();

}

}

}

//----------CON ARRAY DE BOOL LOS ENCUENTRA EN O(1)--------------

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

class Solution

{

static void Main(String[] args)

{

string[] tokens\_n1 = Console.ReadLine().Split(' ');

int n1 = Convert.ToInt32(tokens\_n1[0]);

int n2 = Convert.ToInt32(tokens\_n1[1]);

int n3 = Convert.ToInt32(tokens\_n1[2]);

string[] h1\_temp = Console.ReadLine().Split(' ');

int[] a = Array.ConvertAll(h1\_temp, Int32.Parse);

string[] h2\_temp = Console.ReadLine().Split(' ');

int[] b = Array.ConvertAll(h2\_temp, Int32.Parse);

string[] h3\_temp = Console.ReadLine().Split(' ');

int[] c = Array.ConvertAll(h3\_temp, Int32.Parse);

//int[] a = { 3, 2, 1, 1, 1 };

//int[] b = { 4, 3, 2 };

//int[] c = { 1, 1, 4, 1 };

List<int> x = new List<int>();

//List<int> y = new List<int>();

//List<int> z = new List<int>();

bool[] y = new bool[10000000]; //10^5\*100

bool[] z = new bool[10000000];

int sum\_x = 0;

for (int i = a.Length - 1; i >= 0; i--)

{

sum\_x += a[i];

x.Add(sum\_x);

}

int sum\_y = 0;

for (int i = b.Length - 1; i >= 0; i--)

{

sum\_y += b[i];

y[sum\_y] = true;

}

int sum\_z = 0;

for (int i = c.Length - 1; i >= 0; i--)

{

sum\_z += c[i];

z[sum\_z] = true;

}

int max\_altura = 0;

// x.Sort();

// y.Sort();

// z.Sort();

for (int i = x.Count - 1; i >= 0; i--)

{

//if (y.BinarySearch(x[i]) > -1 && z.BinarySearch(x[i]) > -1)

//{

// max\_altura = x[i];

// break;

//}

if (y[x[i]] && z[x[i]])

{

max\_altura = x[i];

break;

}

}

Console.WriteLine(max\_altura);

Console.ReadLine();

}

}

//-----------SOLUCION DE LA EDITORIAL POR [**nabila\_ahmed**](https://www.hackerrank.com/nabila_ahmed)-----------

**Set by [nabila\_ahmed](https://www.hackerrank.com/nabila_ahmed" \t "_blank)**

Problem Setter's code :

C++

#include <bits/stdc++.h>

using namespace std;

int arr1[100005];

int arr2[100005];

int arr3[100005];

void solution() {

int n1, n2, n3, h, sum1=0, sum2=0, sum3=0, d1, d2, d3;

cin >> n1 >> n2 >> n3;

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

cin>>arr1[i];

sum1+=arr1[i];

}

d1 = 0;

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

cin>>arr2[i];

sum2+=arr2[i];

}

d2 = 0;

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

cin>>arr3[i];

sum3+=arr3[i];

}

d3 = 0;

while(1) {

if(d1 == n1 || d2 == n2 || d3 == n3) {

h = 0;

break;

}

if(sum1 == sum2 && sum2 == sum3) {

h = sum1;

break;

}

// If height op pile one is highest

if(sum1 >= sum2 && sum1 >= sum3) {

sum1 -= arr1[d1];

d1++;

}

// If height op pile two is highest

else if(sum2 >= sum1 && sum2>=sum3) {

sum2 -= arr2[d2];

d2++;

}

// If height op pile three is highest

else if(sum3 >= sum2 && sum3 >= sum1) {

sum3 -= arr3[d3];

d3++;

}

}

cout << h << endl;

}

int main () {

solution();

return 0;

}

---EN C#-------

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

class Solution

{

static void solution(int n1, int n2, int n3, int[] h1, int[] h2, int[] h3)

{

int h = 0, sum1 = 0, sum2 = 0, sum3 = 0, d1, d2, d3;

//cin >> n1 >> n2 >> n3;

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

{

//cin>>arr1[i];

sum1 += h1[i];

}

d1 = 0;

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

{

// cin>>arr2[i];

sum2 += h2[i];

}

d2 = 0;

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

{

// cin>>arr3[i];

sum3 += h3[i];

}

d3 = 0;

while (true)

{

if (d1 == n1 || d2 == n2 || d3 == n3)

{

h = 0;

break;

}

if (sum1 == sum2 && sum2 == sum3)

{

h = sum1;

break;

}

// If height op pile one is highest

if (sum1 >= sum2 && sum1 >= sum3)

{

sum1 -= h1[d1];

d1++;

}

// If height op pile two is highest

else if (sum2 >= sum1 && sum2 >= sum3)

{

sum2 -= h2[d2];

d2++;

}

// If height op pile three is highest

else if (sum3 >= sum2 && sum3 >= sum1)

{

sum3 -= h3[d3];

d3++;

}

}

Console.WriteLine(h);

}

static void Main(String[] args)

{

string[] tokens\_n1 = Console.ReadLine().Split(' ');

int n1 = Convert.ToInt32(tokens\_n1[0]);

int n2 = Convert.ToInt32(tokens\_n1[1]);

int n3 = Convert.ToInt32(tokens\_n1[2]);

string[] h1\_temp = Console.ReadLine().Split(' ');

int[] a = Array.ConvertAll(h1\_temp, Int32.Parse);

string[] h2\_temp = Console.ReadLine().Split(' ');

int[] b = Array.ConvertAll(h2\_temp, Int32.Parse);

string[] h3\_temp = Console.ReadLine().Split(' ');

int[] c = Array.ConvertAll(h3\_temp, Int32.Parse);

solution(n1, n2, n3, a, b, c);

Console.ReadLine();

}

}

--------------------------------

[**saganus**](https://www.hackerrank.com/saganus)17 days ago

I'm sure this is not the most Pythonic way, but it works.

n1,n2,n3 = input().strip().split(' ')

n1,n2,n3 = [int(n1),int(n2),int(n3)]

h1 = [int(h1\_temp) for h1\_temp in input().strip().split(' ')]

h2 = [int(h2\_temp) for h2\_temp in input().strip().split(' ')]

h3 = [int(h3\_temp) for h3\_temp in input().strip().split(' ')]

h1.reverse()

h2.reverse()

h3.reverse()

sum1 = sum(h1)

sum2 = sum(h2)

sum3 = sum(h3)

while (sum1 != sum2) or (sum1 != sum3) or (sum2 != sum3):

if sum1 > sum2 or sum1 > sum3:

sum1 -= h1.pop()

elif sum2 > sum1 or sum2 > sum3:

sum2 -= h2.pop()

elif sum3 > sum1 or sum3 > sum2:

sum3 -= h3.pop()

print(str(sum1))

----EN C#-------

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

class Solution

{

static void Main(String[] args)

{

string[] tokens\_n1 = Console.ReadLine().Split(' ');

int n1 = Convert.ToInt32(tokens\_n1[0]);

int n2 = Convert.ToInt32(tokens\_n1[1]);

int n3 = Convert.ToInt32(tokens\_n1[2]);

string[] h1\_temp = Console.ReadLine().Split(' ');

int[] a = Array.ConvertAll(h1\_temp, Int32.Parse);

string[] h2\_temp = Console.ReadLine().Split(' ');

int[] b = Array.ConvertAll(h2\_temp, Int32.Parse);

string[] h3\_temp = Console.ReadLine().Split(' ');

int[] c = Array.ConvertAll(h3\_temp, Int32.Parse);

//int[] a = { 3, 2, 1, 1, 1 };

//int[] b = { 4, 3, 2 };

//int[] c = { 1, 1, 4, 1 };

Stack<int> sa = new Stack<int>();

Stack<int> sb = new Stack<int>();

Stack<int> sc = new Stack<int>();

int sum1 = 0, sum2 = 0, sum3 = 0;

for (int i = a.Length - 1; i >= 0; i--)

{

sa.Push(a[i]);

sum1 += a[i];

}

for (int i = b.Length - 1; i >= 0; i--)

{

sb.Push(b[i]);

sum2 += b[i];

}

for (int i = c.Length - 1; i >= 0; i--)

{

sc.Push(c[i]);

sum3 += c[i];

}

while ((sum1 != sum2) || (sum1 != sum3) || (sum2 != sum3))

{

if (sum1 > sum2 || sum1 > sum3)

sum1 -= sa.Pop();

else if (sum2 > sum1 || sum2 > sum3)

sum2 -= sb.Pop();

else if (sum3 > sum1 || sum3 > sum2)

sum3 -= sc.Pop();

}

Console.WriteLine(sum1);

Console.ReadLine();

}

}