Equal Stacks

locked

Submitted a few seconds ago • Score: 1.00 Status: Accepted

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5
✔ Test Case #6	✔ Test Case #7	✔ Test Case #8
✔ Test Case #9	✔ Test Case #10	✔ Test Case #11
✔ Test Case #12	✔ Test Case #13	✔ Test Case #14
✔ Test Case #15	✔ Test Case #16	✔ Test Case #17
✔ Test Case #18	✔ Test Case #19	✔ Test Case #20
✔ Test Case #21	✔ Test Case #22	✔ Test Case #23
✔ Test Case #24	✔ Test Case #25	✔ Test Case #26
✔ Test Case #27	✔ Test Case #28	✔ Test Case #29

Privacy - Terms

✔ Test Case #30

Submitted Code

Language: C++ Open in editor

```cpp
// Sample Input

// 5 3 4
// 3 2 1 1 1
// 4 3 2
// 1 1 4 1
// Sample Output

// 5




#include <bits/stdc++.h>
using namespace std;


int main(){
    int n1;
    int n2;
    int n3;
    cin >> n1 >> n2 >> n3;
    int h1 = 0, h2 = 0, h3 = 0; // heights of the 3 stacks
    vector<int> tower1(n1);

```

Privacy - Terms

```
    for(int i = 0; i < n1; i++){
       cin>>tower1[i];
       h1 += tower1[i];
    }

    vector<int> tower2(n2);
    for(int i = 0; i < n2; i++){
       cin>>tower2[i];
       h2 += tower2[i];
    }

    vector<int> tower3(n3);
    for(int i = 0; i < n3; i++){
       cin>>tower3[i];
       h3 += tower3[i];
    }

    // Use a greedy approach, always remove cylinders from the tallest tower until all towers
    // have the same height.

    bool equalHeight = false;
    if(h1 == h2 && h2 == h3)
    {
        equalHeight = true;
    }
    int r1 = 0, r2 = 0, r3 = 0; // Store the indices of which cylinder to remove

    while(!equalHeight)
    {
        if(h1 >= h2 && h1 >= h3) {
            h1 -= tower1[r1];
            r1++;
        } else if(h2 >= h1 && h2 >= h3) {
            h2 -= tower2[r2];
            r2++;
```

Privacy - Terms

```
61          } else if(h3 >= h1 && h3 >= h2) {
62              h3 -= tower3[r3];
63              r3++;
64          }
65          if((h1 == h2 && h2 == h3) || (h1 == 0 && h2 == 0 && h3 == 0)) {
66              equalHeight = true;
67          }
68      }
69      cout<<h1;
70      return 0;
71  }
```


Privacy - Terms