Game of Two Stacks

Submitted a minute ago • Score: 60.00 Status: Accepted

Success 0.03s

✔ 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	

Submitted Code

nguage: C++ Open in editor

```cpp
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

/*
 * Complete the 'twoStacks' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER maxSum
 *  2. INTEGER_ARRAY a
 *  3. INTEGER_ARRAY b
 */

int twoStacks(int maxSum, vector<int> a, vector<int> b) {
    int a_id = 0;
    int b_id = 0;
    int sum = 0;
    int count = 0;

    while (b_id < (int)b.size() && sum + b[b_id] <= maxSum){
        sum += b[b_id];
        b_id++;
    }
    count = b_id;
    b_id--;
```

```
    while (a_id < (int)a.size() && b_id < (int)b.size()){
        sum += a[a_id];
        if (sum > maxSum){
            while(b_id >= 0){
                sum -= b[b_id];
                b_id --;
                if (sum <= maxSum){
                    break;
                }
            }
            if (sum > maxSum && b_id < 0){
                a_id --;
                break;
            }
        }
        count = max(a_id+b_id+2, count);
        a_id ++;
    }

    return count;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string g_temp;
    getline(cin, g_temp);

    int g = stoi(ltrim(rtrim(g_temp)));

    for (int g_itr = 0; g_itr < g; g_itr++) {
        string first_multiple_input_temp;
        getline(cin, first_multiple_input_temp);

        vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));

        int n = stoi(first_multiple_input[0]);

        int m = stoi(first_multiple_input[1]);

        int maxSum = stoi(first_multiple_input[2]);

        string a_temp_temp;
        getline(cin, a_temp_temp);

        vector<string> a_temp = split(rtrim(a_temp_temp));

        vector<int> a(n);

        for (int i = 0; i < n; i++) {
            int a_item = stoi(a_temp[i]);

            a[i] = a_item;
        }

        string b_temp_temp;
        getline(cin, b_temp_temp);

        vector<string> b_temp = split(rtrim(b_temp_temp));

        vector<int> b(m);

        for (int i = 0; i < m; i++) {
            int b_item = stoi(b_temp[i]);
```

```
            b[i] = b_item;
        }

        int result = twoStacks(maxSum, a, b);

        fout << result << "\n";
    }

    fout.close();

    return 0;
}

string ltrim(const string &str) {
    string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) {
    string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
        s.end()
    );

    return s;
}

vector<string> split(const string &str) {
    vector<string> tokens;

    string::size_type start = 0;
    string::size_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));

        start = end + 1;
    }

    tokens.push_back(str.substr(start));

    return tokens;
}

```