Practice Compete Jobs Rank Leaderboard

Dashboard > Algorithms > Search > Ice Cream Parlor

Badge Progress (Details)

Points: 1811 Rank: 13933

You have successfully solved Ice Cream Parlor Share Tweet

Try the Next Challenge | Try a Random Challenge

Ice Cream Parlor

by dheeraj

Problem Submissions Leaderboard Discussions Editorial Topics

Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

Complete the function icecreamParlor below to return an array containing the indexes of the prices of the two flavors they buy. The returned array must be sorted ascending.

Input Format

The first line contains an integer, t, denoting the number of trips to the ice cream parlor. The next t sets of lines each describe a visit. Each trip is described as follows:

1. The integer m, the amount of money they have pooled.
2. The integer n, the number of flavors offered at the time.
3. n space-separated integers denoting the cost of each flavor: $cost[c_1, c_2, \ldots, c_n]$.

Note: The index within the cost array represents the flavor of the ice cream purchased.

Constraints

- $1 \leq t \leq 50$
- $2 \leq m \leq 10^4$
- $2 \leq n \leq 10^4$
- $1 \leq c_i \leq 10^4, \forall\, i \in [1, n]$
- There will always be a unique solution.

Output Format

For each test case, print two space-separated integers denoting the indexes of the two flavors purchased, in ascending order.

Sample Input

```
2
4
5
1 4 5 3 2
4
4
2 2 4 3
```

Sample Output

```
1 4
1 2
```

Explanation

Sunny and Johnny make the following two trips to the parlor:

1. The first time, they pool together $m = 4$ dollars. Of the five flavors available that day, flavors 1 and 4 have a total cost of $1 + 3 = 4$.
2. The second time, they pool together $m = 4$ dollars. TOf the four flavors available that day, flavors 1 and 2 have a total cost of $2 + 2 = 4$.

Easy

Submitted 51412 times
Max Score **30**

Need Help?

View Discussions

View Editorial Solution

View Top Submissions

Rate This Challenge:

☆☆☆☆☆

Resources

Binary Search

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable)

C++14

```cpp
#include <vector>
#include <map>
#include <iostream>
#include <algorithm>
typedef unsigned long int uint32;

int main()
{
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(0);
    std::cout.tie(0);

    uint32 T; std::cin >> T;
    for (size_t t = 0; t < T; ++t)
    {
        uint32 m = 0, n = 0;
        std::multimap<uint32, uint32> myMap;
        std::vector<uint32> vecValues;

        std::cin >> m >> n;

        for (size_t idx = 1; idx <= n; ++idx)
        {
            uint32 tempFirst = 0; std::cin >> tempFirst;
            if (tempFirst <= m)
            {
                myMap.emplace(tempFirst, idx);
                vecValues.push_back(tempFirst);
            }
        }

        for (auto vec = vecValues.cbegin(); vec != vecValues.cend(); ++vec)
        {
            std::multimap<uint32, uint32>::const_iterator current = myMap.find(*vec);
            uint32 diff       = m-(current->first);
            uint32 curr_value = current->first;
            uint32 id         = current->second;
            //cout << "current value :" << curr << " id :" << id << " difference2find :" << diff << endl;
            myMap.erase(current);

            std::multimap<uint32, uint32>::const_iterator gett_id2 = myMap.find(diff);
            if ((gett_id2 != myMap.cend()) && (gett_id2->first + curr_value == m) )
            {
                std::cout << id << " " << gett_id2->second << std::endl;
                break;
            }
        }
    }
    return 0;
}

```

Line: 51 Col: 1

Upload Code as File ☐ Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

Challenge your friends:

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

Next Challenge

Contest Calendar|Blog|Scoring|Environment|FAQ|About Us|Support|Careers|Terms Of Service|Privacy Policy|Request a Feature