Hash Tables: Ice Cream Parlor ☆

You have successfully solved Hash Tables: Ice Cream Parlor Share Tweet

Try the next challenge

Problem Submissions Leaderboard Editorial

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Each time Sunny and Johnny take a trip to the Ice Cream Parlor, they pool their money to buy ice cream. On any given day, the parlor offers a line of flavors. Each flavor has a cost associated with it.

Given the value of $money$ and the $cost$ of each flavor for t trips to the Ice Cream Parlor, help Sunny and Johnny choose two distinct flavors such that they spend their entire pool of money during each visit. ID numbers are the 1- based index number associated with a $cost$. For each trip to the parlor, print the ID numbers for the two types of ice cream that Sunny and Johnny purchase as two space-separated integers on a new line. You must print the smaller ID first and the larger ID second.

For example, there are $n = 5$ flavors having $cost = [2, 1, 3, 5, 6]$. Together they have $money = 5$ to spend. They would purchase flavor ID's 1 and 3 for a cost of $2 + 3 = 5$. Use 1 based indexing for your response.

Note:

- Two ice creams having unique IDs i and j may have the same cost (i.e., $cost[i] \equiv cost[j]$).
- There will always be a unique solution.

Function Description

Complete the function whatFlavors in the editor below. It must determine the two flavors they will purchase and print them as two space-separated integers on a line.

whatFlavors has the following parameter(s):

- cost: an array of integers representing price for a flavor
- money: an integer representing the amount of money they have to spend

Input Format

The first line contains an integer, t, the number of trips to the ice cream parlor.

Each of the next t sets of 3 lines is as follows:

- The first line contains $money$.
- The second line contains an integer, n, the size of the array $cost$.
- The third line contains n space-separated integers denoting the $cost[i]$.

Constraints

- $1 \leq t \leq 50$
- $2 \leq money \leq 10^9$

- $2 \leq n \leq 5 * 10^4$
- $1 \leq cost[i] \leq 10^9$

Output Format

Print two space-separated integers denoting the respective indices for the two distinct flavors they choose to purchase in ascending order. Recall that each ice cream flavor has a unique ID number in the inclusive range from 1 to $|cost|$.

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 $money = 4$ dollars. There are five flavors available that day and flavors 1 and 4 have a total cost of $1 + 3 = 4$.
2. The second time, they pool together $money = 4$ dollars. There are four flavors available that day and flavors 1 and 2 have a total cost of $2 + 2 = 4$.

Change Theme C++

```
//https://www.youtube.com/watch?v=OtSf56T6Q40&ab_channel=BinaryBeast
//I was able to develop the logic myself but was making minor mistakes.
// Complete the whatFlavors function below.
void whatFlavors(vector<int> cost, int money) {
    ll n = cost.size();
    unordered_map<ll,ll> value;
        unordered_map<ll,vector<ll>> pos;
        for(ll i=0;i<n;i++)
        {
            value[cost[i]]++;
            pos[cost[i]].push_back(i+1);
        }
         if(money%2==0)
        {
            int temp = money/2;
            if(value[temp] > 1)
            {
                ll a1 = pos[temp][0];
                ll a2 = pos[temp][1];
                cout<<min(a1,a2)<<" "<<max(a1,a2)<<endl;
            }
            else{
                for(ll i=0;i<n;i++)
                {
                    ll a = pos[cost[i]][0];
                    value[cost[i]]--;
                    ll b;
                    if(value[money - cost[i]]!=0)
                    {
                        b = pos[money - cost[i]][0];
                        cout<<min(a,b)<<" "<<max(a,b)<<endl;
                        break;
                    }
                }
            }
        }
        else{
             for(ll i=0;i<n;i++)
                {
                    ll a = pos[cost[i]][0];
                    value[cost[i]]--;
                    ll b;
                    if(value[money - cost[i]]!=0)
                    {
                        b = pos[money - cost[i]][0];
                        cout<<min(a,b)<<" "<<max(a,b)<<endl;
                        break;
                    }

                }
        }
}
```

Line: 40 Col: 18

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- ✓ Test case 0
- ✓ Test case 1
- ✓ Test case 2
- ✓ Test case 3
- ✓ Test case 4
- ✓ Test case 5
- ✓ Test case 6

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock

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