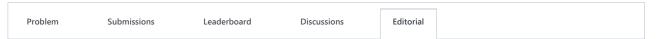


All Domains > Tutorials > Cracking the Coding Interview > Binary Search: Ice Cream Parlor

Binary Search: Ice Cream Parlor ■





Check out the resources on the page's right side to learn more about binary search. The video tutorial is by Gayle Laakmann McDowell, author of the bestselling interview book Cracking the Coding Interview.

Each time Sunny and Johnny take a trip to the Ice Cream Parlor, they pool together m dollars for ice cream. On any given day, the parlor offers a line of n flavors. Each flavor, i, is numbered sequentially with a unique ID number from 1 to n and has a cost, c_i , associated with it.

Given the value of m 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 (m) during each visit. 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.

Note: Two ice creams having unique IDs i and j may have the same cost (i.e., $c_i \equiv c_j$).

Input Format

The first line contains an integer, t, denoting the number of trips to the ice cream parlor. The 3t subsequent lines describe all of Sunny and Johnny's trips to the parlor; each trip is described as follows:

- 1. The first line contains m.
- 2. The second line contains n.
- 3. The third line contains n space-separated integers denoting the cost of each respective flavor. The ith integer corresponds to the cost, ci, for the ice cream with ID number i (where $1 \le i \le n$).

Constraints

- $1 \le t \le 50$
- $2 \le m \le 10^4$
- $2 \le n \le 10^4$
- $1 \le c_i \le 10^4$, where $i \in [1, n]$
- It is guaranteed that there will always be a unique solution.

Output Format

Print two space-separated integers denoting the respective ID numbers for the two distinct flavors they choose to purchase, where the smaller ID is printed first and the larger ID is printed second. Recall that each ice cream flavor has a unique ID number in the inclusive range from 1 to no.

Sample Input

- 1 4 5 3 2
- 2 2 4 3

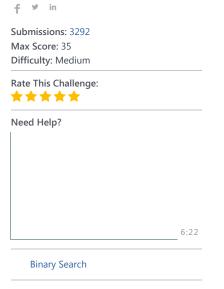
Sample Output

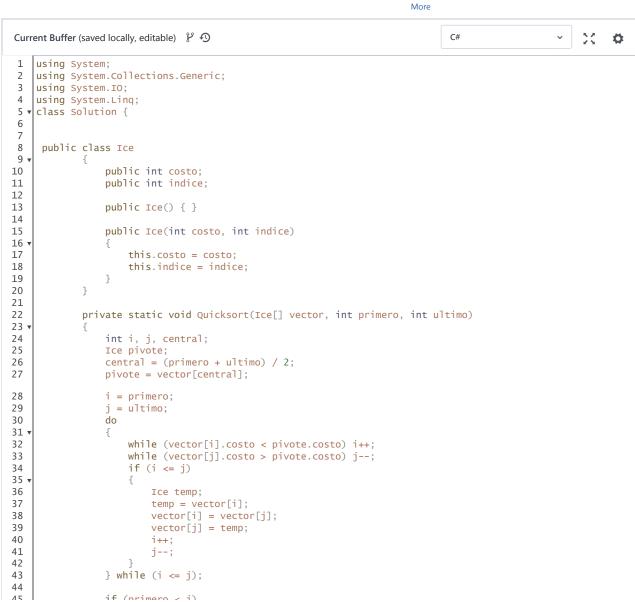
- 1 4
- 1 2

Explanation

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

- 1. The first time, they pool together m=4 dollars. There are five flavors available that day and flavors 1 and 4 have a total cost of 1+3=4. Thus, we print 1 4 on a new line.
- 2. The second time, they pool together m=4 dollars. There are four flavors available that day and flavors 1 and 2 have a total cost of 2+2=4. Thus, we print 1 2 on a new line.





```
46
 47
                       Quicksort(vector, primero, j);
 48
 49
                  if (i < ultimo)</pre>
 50
                  {
                       Quicksort(vector, i, ultimo);
 51
 52
 53
 54
 55
              // A iterative binary search function. It returns location of x in
 56
              // given array arr[]..r] if present, otherwise -1
 57
              static int BinarySearch(Ice[] arr, int 1, int r, int x)
 58
 59
                  //int 7 = 0, r = arr.Length - 1;
 60
 61
                  while (1 \ll r)
 62
 63
                       int m = 1 + (r - 1) / 2;
 64
 65
                       // Check if x is present at mid
 66
                       if (arr[m].costo == x)
 67
 68
                           return m;
 69
 70
 71
                       // If x greater, ignore left half
 72
                       if (arr[m].costo < x)</pre>
 73
 74
                           1 = m + 1;
 75
 76
 77
                       // If x is smaller, ignore right half
 78
                       else
 79
 80
                           r = m - 1;
 81
 82
 83
 84
                  // if we reach here, then element was not present
 85
                  return -1;
 86
 87
 88
 89
              static void Main(string[] args)
 90
 91
                  int t = Convert.ToInt32(Console.ReadLine());
 92
                  for (int a0 = 0; a0 < t; a0++)
 93
                       int m = Convert.ToInt32(Console.ReadLine());
 94
                       int n = Convert.ToInt32(Console.ReadLine());
string[] a_temp = Console.ReadLine().Split(' ');
 95
 96
 97
                       int[] a = Array.ConvertAll(a_temp, e => int.Parse(e));
 98
 99
                       Ice[] iceCream = new Ice[n];
                       for (int i = 0; i < n; i++)
100
101
                           iceCream[i] = new Ice(a[i], i + 1);
102
103
104
                       Quicksort(iceCream, 0, n - 1);
105
106
107
                       for (int i = 0; i < n; i++)
108
                           int complemento = m - iceCream[i].costo;
109
110
111
                           int indiceComplemento = BinarySearch(iceCream, i + 1, iceCream.Length - 1,
     complemento);
112
113
                           if (indiceComplemento != -1)
114
115
                               Console.WriteLine(Math.Min(iceCream[i].indice,
      iceCream[indiceComplemento].indice) + " "
116
                                    + Math.Max(iceCream[i].indice, iceCream[indiceComplemento].indice));
117
                               break;
118
119
120
121
122
123
124
                  // Console.ReadLine();
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

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