

1st LGE Code Jam 2019 - Problem A (English)

1st LGE Code Jam 2019 - Problem B

1st LGE Code Jam 2019 - Problem B (English)

1st LGE Code Jam 2019 - Problem C

1st LGE Code Jam 2019 - Problem C (English)

1st LGE Code Jam 2019 - Problem D

1st LGE Code Jam 2019 - Problem D (English)

1st LGE Code Jam 2019 - Problem E

1st LGE Code Jam 2019 - Problem E (English)

1st LGE Code Jam 2019 - Problem F

1st LGE Code Jam 2019 - Problem F (English)

2019 Online Code Jam 2 - Editorial

코드잼 준비하기

2019년 활동

2018년 활동

2016년 활동

2015년 활동

2014년 활동

2013년 활동

2012년 활동

코딩전문가와 함께하는 코딩 도장

Space tools

1st LGE Code Jam 2019 - Problem C(English)

Created by 박선현 sunhyun.park on 2019/05/15

[Solution]

In easiest way, we can find the closest element through comparing all element of B.

In this case, time complexity would be O(m x n), we can't solve this problem in time limit.

We could use binary search to find out the closest value after sorting the array.

In this way, time complexity could be reduced about to O(m x log n).

```
#include <iostream>
#include <algorithm>

using namespace std;

int A[1000000];
int B[1000000];

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int t, n, m;
    cin >> t;
    while (t--) {
        cin >> n >> m;
        for (int i = 0; i < n; i++) {
            cin >> A[i];
        }
        for (int i = 0; i < m; i++) {
            cin >> B[i];
        }
        sort(B, B + m);
        long long ret = 0;
        for (int i = 0; i < n; i++) {
            int idx = lower_bound(B, B + m, A[i]) - B;
            if (idx == 0 || (idx < m && B[idx] - A[i] < A[i] - B[idx - 1])) {
                ret += B[idx];
            } else {
                ret += B[idx - 1];
            }
        }
        cout << ret << endl;
    }
    return 0;
}
```

Like

Be the first to like this

No labels

Write a comment...