

P1678 烦恼的高考志愿

题目描述

现有 m ($m \leq 100000$) 所学校，每所学校预计分数线是 a_i ($a_i \leq 10^6$)。有 n ($n \leq 100000$) 位学生，估分分别为 b_i ($b_i \leq 10^6$)。

根据 n 位学生的估分情况，分别给每位学生推荐一所学校，要求学校的预计分数线和学生的估分相差最小（可高可低，毕竟是估分嘛），这个最小值为不满意度。求所有学生不满意度和的最小值。

输入格式

第一行读入两个整数 m, n 。 m 表示学校数， n 表示学生数。第二行共有 m 个数，表示 m 个学校的预计录取分数。第三行有 n 个数，表示 n 个学生的估分成绩。

输出格式

一行，为最小的不满度之和。

输入样例

```
4 3
513 598 567 689
500 600 550
```

输出样例

```
32
```

解析

将题意翻译一下：查找最接近的数字。那么直接套用模板即可。

编码

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

using namespace std;

//学校的录取分数
int school[100001];
//学生考的分数
int scores[100001];
//学校数量
int m;
```

```

//学生数量
int n;

//二分查找法查找最接近的数值
int BinSearch(int x) {
    int left = 0, right = m - 1;
    int middle;
    //先判断目标数字是否存在于左右两个边界
    //此处的代码必须保留，因为下面的搜索最终结果会导致left>right，导致最终计算错误
    if (x <= school[left]) {
        return abs(x - school[left]);
    }
    if (x >= school[right]) {
        return abs(x - school[right]);
    }

    //当数字不存在于数组的时候，即left大于right的时候，
    //此时我们将在nums[left]和nums[right]中比较哪个离得更近
    while (left <= right) {
        middle = left + (right - left) / 2;
        //直接找到目标
        if (school[middle] == x) {
            return 0;
        }
        //中间值大于目标，在左侧进行查找
        else if (school[middle] > x) {
            right = middle - 1;
        }
        //中间值小于目标，在右侧进行查找
        else if (school[middle] < x) {
            left = middle + 1;
        }
    }

    //找不到的时候作比较一下到底哪个离得更近;
    int minLeft = abs(school[left] - x);
    int minRight = abs(x - school[right]);
    //返回最小值
    return min(minLeft, minRight);
}

int main(int argc, char *argv[]) {
    //读入基础数据
    cin >> m >> n;
    int sum = 0;
    for (int i = 0; i < m; ++i) {
        cin >> school[i];
    }
}

```

```
//将待查找的数组按照升序进行排列
sort(school, school + m);
//使用二分搜索查找最接近的数字
for (int i = 0; i < n; ++i) {
    cin >> scores[i];
    int value = BinSearch(scores[i]);
    sum += value;
}
cout << sum;
return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。



昔誤