

整数二分

给定一个按照升序排列的长度为 n 的整数数组，以及 q 个查询。

对于每个查询，返回一个元素 k 的起始位置和终止位置（位置从 0 开始计数）。

如果数组中不存在该元素，则返回 `-1 -1`。

输入格式

第一行包含整数 n 和 q ，表示数组长度和询问个数。

第二行包含 n 个整数（均在 $1 \sim 10000$ 范围内），表示完整数组。

接下来 q 行，每行包含一个整数 k ，表示一个询问元素。

输出格式

共 q 行，每行包含两个整数，表示所求元素的起始位置和终止位置。

如果数组中不存在该元素，则返回 `-1 -1`。

数据范围

$$1 \leq n \leq 100000$$

$$1 \leq q \leq 10000$$

$$1 \leq k \leq 10000$$

输入样例：

```
6 3
1 2 2 3 3 4
3
4
5
```

输出样例：

```
3 4
5 5
-1 -1
```

① $mid = \frac{l+r+1}{2}$

$if (check(mid))$

$\left\{ \begin{array}{l} true \quad [mid, r] : l = mid \\ false \quad [l, mid-1] : r = mid-1 \end{array} \right.$

② $mid = \frac{l+r}{2}$

$if (check(mid))$

$\left\{ \begin{array}{l} true \quad [l, mid] : r = mid \\ false \quad [mid+1, r] : l = mid+1 \end{array} \right.$

找最小的使用第二种，找最大的使用第一种。

```
#include <iostream>

using namespace std;

const int N = 100010;

int n, m;
int q[N];

int main()
{
    scanf("%d%d", &n, &m);
    for (int i = 0; i < n; i ++ ) scanf("%d", &q[i]);

    while (m -- )
    {
        int x;
        scanf("%d", &x);

        int l = 0, r = n - 1;
        while (l < r)
        {
            int mid = l + r >> 1;
            if (q[mid] >= x) r = mid;    //注意加等于
            else l = mid + 1;
        }

        if (q[l] != x) cout << "-1 -1" << endl;
        else
        {
            cout << l << ' ';

            int l = 0, r = n - 1;
            while (l < r)
            {
                int mid = l + r + 1 >> 1;
                if (q[mid] <= x) l = mid;    //注意加等于
                else r = mid - 1;
            }

            cout << l << endl;
        }
    }

    return 0;
}
```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/39787/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

