

## 统计匹配的二元组个数

知识点数组

时间限制：1s 空间限制：32MB 限定语言：不限

### 题目描述：

给定两个数组A和B，若数组A的某个元素A[i]与数组B中的某个元素B[j]满足  $A[i] == B[j]$ ，则寻找到一个值匹配的二元组(i, j)。请统计在这两个数组A和B中，一共存在多少个这样的二元组。

### 输入描述：

第一行输入数组A的长度M；第二行输入数组B的长度N；第三行输入数组A的值；第四行输入数组B的值。

$1 \leq M, N \leq 100000$

A, B数组中数值的取值均小于100000；

### 输出描述：

输出匹配的二元组个数。

### 补充说明：

若不存在相等的值，则输出0。所采用的算法复杂度需小于 $O(N^2)$ ，否则会超时。输入数组中允许出现重复数字，一个数字可以匹配多次。

## 示例1

输入：

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

输出：

```
4
```

说明：

若下标从0开始，则匹配的二元组分别为(0, 3), (1, 2), (2, 1), (3, 0), 共计4个。

## 示例2

输入：

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

输出：

```
4
```

说明：

若下标从0开始，则匹配的二元组分别为(0, 0), (1, 1), (4, 1), (5, 0) 共计4个。

## 解题思路：

1. 使用map来记录A的元素及其个数
2. 如果map中的key在B中存在，则二元数组个数为其map的value值

```
public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int M = sc.nextInt();
        int N = sc.nextInt();
        sc.nextLine();

        String[] A = sc.nextLine().split(" ");
        String[] B = sc.nextLine().split(" ");

        Map<String, Integer> map = new HashMap<>();
        for(int i=0; i<M; i++){
            String a = A[i];
            map.put( a, map.getOrDefault( a, 0) + 1);
        }

        int res = 0;
        for(int i=0; i<N; i++){
            String b = B[i];
            if(map.containsKey(b)){
                res += map.get(b);
            }
        }

        System.out.println(res);
    }

}
```