

# 矩形

## 描述

给定两个矩阵，判断第二个矩阵在第一个矩阵的哪些位置出现过。

## 输入

输入的第一行包含四个正整数 $a, b, c, d$ ，表示第一个矩阵大小为 $a \times b$ ，第二个矩阵的大小为 $c \times d$ 。

接下来是一个 $a \times b$ 的矩阵。

再接下来是一个 $c \times d$ 的矩阵。

保证矩阵中每个数字都为正整数且不超过100。

## 输出

若第二个矩阵在第一个矩阵的 $(i, j)$ 位置出现（即出现位置的左上角），输出 $i$ 和 $j$ 。若有多个位置，按字典序从小到大的顺序依次输出。

字典序：对于两个位置 $(a, b), (c, d)$ ，若 $a < c$ 则 $(a, b)$ 比 $(c, d)$ 小，若 $a > c$ 则 $(a, b)$ 比 $(c, d)$ 大，若 $a = c$ 则再像前边一样比较 $b$ 和 $d$ 。

## 样例1输入

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

## 样例1输出

```
1 1
1 3
3 2
```

## 样例1解释

矩阵2在矩阵1的 $(1, 1)$ 、 $(1, 3)$ 、 $(3, 2)$ 这些位置出现了。

## 样例2

请查看下发文件 (attachment/8785/87857d0f6284f999593b8b05066e4055683f9e8c.zip)内的sample2\_input.txt和sample2\_output.txt。

## 限制

对于50%的数据,  $a, b, c, d \leq 50$ ;

对于100%的数据,  $a, b, c, d \leq 1000$ 。

时间: 4 sec

空间: 512 MB

## 提示

[对于长度为 $L$ 的数列 $S$ ,  $S$ 中最大的元素为 $K$ , 我们设他的 $hash$ 值 $H(S) = S_1C^0 + S_1C^1 + \dots + S_LC^{L-1}$ , 其中 $C$ 为任意大于 $K$ 的常数]

[对于不同的字符串 $A, B$ ,  $H(A) \neq H(B)$ ]

[我们先来看看一维的情况, 给定两个字符串 $A, B$ , 我们怎么判断 $A$ 在 $B$ 中出现? 显然我们可以用 $hash$ 来判断。但是 $hash$ 值太大了怎么办? 取模呀! 找一个比较好的质数 $p$ , 对于字符串 $A, B$ , 若 $A = B$ 则显然 $H(A) \bmod p = H(B) \bmod p$ ; 若 $A \neq B$ ,  $H(A) \bmod p$ 有一定概率会和 $H(B) \bmod p$ 相同。怎么办呢? 我们再看第二个质数 $q$ , 再来验证 $H(A) \bmod q$ 和 $H(B) \bmod q$ ! 可以证明, 这样基本上是不会再出错了的。]

[拓展到二维。]

[对于第一个矩阵: ]

[我们可以对每一个元素求向左长度为 $d$ 的矩阵元素的 $hash$ 值, 得到一个矩阵。]

[然后我们再用新矩阵, 再做一次 $hash$ , 就是向上长度为 $c$ 的矩阵元素的 $hash$ 值, 得到新矩阵 $X$ 。]

[接着我们将第二个字符串的 $hash$ 值求出来, 就是先每一行求一个 $hash$ 值, 再将这 $c$ 个 $hash$ 值再 $hash$ 一次变成一个数字, 然后我们就去 $X$ 矩阵中找这个数字, 找到多少个就说明第二个矩阵在第一个矩阵中出现了多少次。]

[时间复杂度 $O(n^2)$ ]

另外, 为了帮助大家完成题目, 我们提供了只包含了输入输出功能的程序模板, 也提供了含有算法的大部分实现细节的程序。

你可以根据自己的实际情况, 在这些程序的基础上进行作答, 或不参考这些程序, 这将与你的得分无关。

这些程序可以从【[这里](#) (attachment/b23f/b23f7b079eeb5ef7c08e37c3652bc3497c11d31e.zip)】下载。

---

UI powered by Twitter Bootstrap (<http://getbootstrap.com/>).

Tsinghua Online Judge is designed and coded by Li Ruizhe.

For all suggestions and bug reports, contact [oj\[at\]liruizhe\[dot\]org](mailto:oj[at]liruizhe[dot]org).