

题目描述

告警抑制，是指高优先级告警抑制低优先级告警的规则。高优先级告警产生后，低优先级告警不再产生。请根据原始告警列表和告警抑制关系，给出实际产生的告警列表。

- 不会出现循环抑制的情况。
- 告警不会传递，比如A->B,B->C，这种情况下A不会直接抑制C。但被抑制的告警仍然可以抑制其他低优先级告警。

输入描述

第一行为数字N，表示告警抑制关系个数， $0 \leq N \leq 120$

接下来N行，每行是由空格分隔的两个告警ID，例如: id1 id2，表示id1抑制id2，告警ID的格式为：

大写字母+0个或者1个数字

最后一行为告警产生列表，**列表长度** [1,100]

输出描述

真实产生的告警列表

备注

告警ID之间以单个空格分隔

用例

输入	2 A B B C A B C D E
输出	A D E
说明	A抑制了B, B抑制了C, 最后实际的告警为A D E

输入	4 F G C B A G A0 A A B C D E
输出	A C D E
说明	无

题目解析

我们基于用例2来说明下这题。

有一个告警列表alertList，A B C D E，我们逐一检查对应的告警是否可以正常发生：

- 第一个告警A，能够抑制它的只有A0，而当前告警列表alertList中没有A0，因此告警A可以正常发生
- 第二个告警B，能够抑制它的只有C，而当前告警列表alertList中有C，因此告警B被抑制，不可以发生
- 第三个告警C，没有能抑制它的告警，因此正常发生
- 第四个告警D，没有能抑制它的告警，因此正常发生
- 第五个告警E，没有能抑制它的告警，因此正常发生

因此，本题的解题思路很简单，只需要记录每一个告警id2的所有抑制它的告警集合id1s，然后遍历告警列表alertList，遍历每一个告警id2：

- 如果没有可以抑制id2的更高级的告警，则id2正常发生
- 如果有可以抑制id2的更告警的告警id1，但是id1没有出现在alertList中，则id2正常发生
- 其他情况，id2不可以发生

```
1 import java.util.*;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int n = Integer.parseInt(sc.nextLine());
8
9         HashMap<String, HashSet<String>> fa = new HashMap<>();
10        for (int i = 0; i < n; i++) {
11            String[] tmp = sc.nextLine().split(" ");
12            // id1抑制id2
13            String id1 = tmp[0], id2 = tmp[1];
14            fa.putIfAbsent(id2, new HashSet<>());
15            // fa用于记录抑制id2的所有id1的集合
16            fa.get(id2).add(id1);
17        }
18
19        String[] alertList = sc.nextLine().split(" ");
20
21        System.out.println(getResult(fa, alertList));
22    }
23
24    public static String getResult(HashMap<String, HashSet<String>> fa, String[]
alertList) {
```

```
25 HashSet<String> alertSet = new HashSet<>(Arrays.asList(alertList));  
26  
27 StringJoiner sj = new StringJoiner(" ");  
28  
29 for (String id2 : alertList) {  
30     // 如果没有抑制id2的更高级的告警，或者有抑制id2的更高级的告警，但是此高级告警没有出现在  
alertList列表中  
31     if (!fa.containsKey(id2) || Collections.disjoint(fa.get(id2), alertSet)) {  
32         // 此时id2就可以正常告警  
33         sj.add(id2);  
34     }  
35 }  
36  
37 return sj.toString();  
38 }  
39 }
```