

E6-D题

题目描述

给定仅包含小写英文字母的字符串 S 与 T 。

初始时，你有一个空字符串 A 。每次你可以对 A 进行以下操作：

- 选择一个 S 的前缀 S' ，将 S' 连接在 A 后。

你需要判断是否存在一种操作方案使得 $A=T$ 。

输入

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 10$)，表示数据组数。

对于每组数据：

第一行，一个仅包含小写英文字母的字符串 S ($1 \leq |S| \leq 1 \times 10^5$)。

第二行，一个仅包含小写英文字母的字符串 T ($1 \leq |T| \leq 1 \times 10^5$)。

输出

对于每组数据：

一行，若存在一种操作方案使得 $A=T$ 则输出 `owo`，否则输出 `ono`。

输入样例

```
1 2
2 buaa
3 bubuaabu
4 mips
5 riscv
```

输出样例

```
1 owo
2 ono
```

解题思路

原题可以等价于 能否将 T 分割成若干个 S 的前缀；

可以联想到KMP算法和next数组；

如果每次尝试匹配 S 和 T 后，匹配长度 j 都不为0，那么字符串 S 就可以划分为若干个 S 的前缀，即存在一种操作方案使得 $A=T$

参考代码

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  // 求next数组
6  vector<int> pi(string s) {
7      int n = (int) s.length();
8      vector<int> pi(n);
9      for (int i = 1; i < n; i++) {
10         int j = pi[i - 1];
11         while (j > 0 && s[i] != s[j]) j = pi[j - 1];
12         if (s[i] == s[j]) j++;
13         pi[i] = j;
14     }
15     return pi;
16 }
17
18 int main() {
19     ios::sync_with_stdio(false);
20     cin.tie(0);
21     int tt;
22     cin >> tt;
23     while (tt--) {
24         string s, t;
25         cin >> s >> t;
26         int n = (int) s.length();
27         vector<int> p = pi(s);
28         int j = 0;
29         bool ok = true;
30         // 不断进行匹配
31         for (int i = 0; i < (int) t.length(); i++) {
32             if (s[j] == t[i]) {
33                 j = (j + 1) % n;
34                 continue;
35             }
36             while (j > 0 && s[j] != t[i]) j = p[j - 1];
37             if (s[j] != t[i]) {
38                 ok = false;
39                 break;
40             }
41             j = (j + 1) % n;
42         }
```

```
43         cout << (ok ? "owo" : "ono") << '\n';  
44     }  
45     return 0;  
46 }
```