

G. Anthem of Berland(Codeforces 808)

DP

KMP

给定一个含小写字母和?的s串和一个含小写字母的t串，其中?可以匹配任意字符，求s串最多匹配t串多少次。

链接

[Codeforces 808-G](#)

题解

算法思路

使用DP

设状态 $d(i,j)$ 表示s的前i个字符，在其后缀匹配t前缀的最大长度为j的情况下，对t的最多匹配次数。

则状态转移方程为

$$d(i+1,k) = \max\{d(i,j) + (ch(j,c) == m) \mid 0 \leq j \leq m \ \&\& \ k = ch(j,c)\}$$

其中 $ch(j,c)$ 为t串的前j个字符接上字符c后，匹配t串前缀的最大匹配长度，而 $m = \text{len}(t)$ 。

$ch(j,c)$ 使用KMP算法中的失配函数计算。先依次接上t中的字符，最大匹配长度即为该字符序号。接上任意字符时通过失配函数转移到之前已经计算过的状态即得最大匹配长度。

复杂度分析

1. 空间复杂度

s和t长度小于 10^5 ，因此空间复杂度为 $O(10^5)$ ，占用内存 $k \cdot 10\text{MB}$ ， $k < 10$ 。

2. 时间复杂度

设 $n = \text{len}(s)$ ， $m = \text{len}(t)$ ，状态数 $O(mn)$ 个，决策小于26个，总时间复杂度为 $O(mn)$ 。

示例

```
??c???? //s串
```

```
abccab //t串
```

```
fail: //失配函数fail
```

```
0 0 //初始状态
```

```
1 0 //状态1，表示前1个字符匹配
```

```
2 0
```

```
3 0
```

```
4 1 //状态4，表示前4个字符匹配，正在匹配第5个字符。失配则回到状态1
```

```
5 2
```

```
ch:
```

```
0 0 1 //已匹配t的前0个字符，当接上字符0(即a)，匹配t的最大长度为1
```

```
0 1 0
```

```
0 2 0
```

```
0 3 0
```

```
0 4 0
```

```
0 5 0
```

```
0 6 0
```

```
0 7 0
```

```
0 8 0
```

```
0 9 0
```

```
0 10 0
```

```
0 11 0
```

```
0 12 0
```

```
0 13 0
```

```
0 14 0
```

```
0 15 0
```

```
0 16 0
```

```
0 17 0
```

```
0 18 0
```

```
0 19 0
```

```
0 20 0
```

```
0 21 0
```

```
0 22 0
```

```
0 23 0
```

```
0 24 0
```

```
0 25 0
```

```
1 0 1
```

```
1 1 2 //已匹配t的前1个字符，当接上字符1(即b)，匹配t的最大长度为2
```

```
1 2 0
```

```
1 3 0
```

```
1 4 0
```

```
1 5 0
```

```
1 6 0
```

```
1 7 0
```

```
1 8 0
```

```
1 9 0
```

```
1 10 0
```

```
1 11 0
```

```
1 12 0
```

```
1 13 0
```

```
1 14 0
```

1 15 0
1 16 0
1 17 0
1 18 0
1 19 0
1 20 0
1 21 0
1 22 0
1 23 0
1 24 0
1 25 0
2 0 1
2 1 0
2 2 3
2 3 0
2 4 0
2 5 0
2 6 0
2 7 0
2 8 0
2 9 0
2 10 0
2 11 0
2 12 0
2 13 0
2 14 0
2 15 0
2 16 0
2 17 0
2 18 0
2 19 0
2 20 0
2 21 0
2 22 0
2 23 0
2 24 0
2 25 0
3 0 4
3 1 0
3 2 0
3 3 0
3 4 0
3 5 0
3 6 0
3 7 0
3 8 0
3 9 0
3 10 0
3 11 0
3 12 0
3 13 0
3 14 0

3 15 0

```
3 16 0
3 17 0
3 18 0
3 19 0
3 20 0
3 21 0
3 22 0
3 23 0
3 24 0
3 25 0
4 0 1
4 1 5
4 2 0
4 3 0
4 4 0
4 5 0
4 6 0
4 7 0
4 8 0
4 9 0
4 10 0
4 11 0
4 12 0
4 13 0
4 14 0
4 15 0
4 16 0
4 17 0
4 18 0
4 19 0
4 20 0
4 21 0
4 22 0
4 23 0
4 24 0
4 25 0
5 0 1
5 1 0
5 2 3 //已匹配abcab，当接上字符c，匹配t的最大长度为3
5 3 0
5 4 0
5 5 0
5 6 0
5 7 0
5 8 0
5 9 0
5 10 0
5 11 0
5 12 0
5 13 0
5 14 0
5 15 0

5 16 0
```

```
5 17 0
5 18 0
5 19 0
5 20 0
5 21 0
5 22 0
5 23 0
5 24 0
5 25 0
```

d:

```
0 0 0
0 1 -1
0 2 -1
0 3 -1
0 4 -1
0 5 -1
1 0 0
1 1 0
1 2 -1
1 3 -1
1 4 -1
1 5 -1
2 0 0
2 1 0
2 2 0
2 3 -1
2 4 -1
2 5 -1
3 0 0
3 1 -1
3 2 -1
3 3 0
3 4 -1
3 5 -1
4 0 0
4 1 0
4 2 -1
4 3 -1
4 4 0
4 5 -1
5 0 0
5 1 0
5 2 0
5 3 -1
5 4 -1
5 5 1 //s的前5个字符??c??, 其后缀匹配t前缀的最大长度为5时（即填abcab），最大匹配次数为1
6 0 1
6 1 1
6 2 0
6 3 1
6 4 -1
6 5 -1
```

```
7 0 1
7 1 1
7 2 1
7 3 0
7 4 1
7 5 -1
8 0 1
8 1 1
8 2 1
8 3 1
8 4 0
8 5 2
2 //最终答案
```