

题目回顾

H 妮妮与自动机

时间限制：3000ms 内存限制：65536kb

通过率：16/25 (64.00%) 正确率：16/112 (14.29%)

题目描述

妮妮最近学习了一个叫做字符串匹配自动机的东西，妮妮觉得他十分高级，但是很可惜她不会实现。

由字符串 S 构建的字符串匹配自动机共有 $n + 1$ 个状态，其中 n 为字符串 S 的长度。第 i 个状态表示接收到的字符串的最后 $i - 1$ 个字符（长度为 $i - 1$ 的后缀）恰能匹配 S 的前 $i - 1$ 个字符，第 $n + 1$ 个状态表示已经匹配上字符串 S ，为字符串匹配自动机的终态。

例如：对于字符串 aab ，它的字符串匹配自动机有 4 个状态（记为状态 $0, 1, 2, 3$ ），分别表示当前匹配上了 \emptyset, a, aa, aab 。则：

状态 0 接收到字符 a 时转移到状态 1 ，接收到其他字符时维持状态 0 。

状态 1 接收到字符 a 时转移到状态 2 ，接收到其他字符时转移到状态 0 。

状态 2 接收到字符 a 时维持状态 2 ，接收到字符 b 时转移到状态 3 ，接收到其他字符时转移到状态 0 。

作为一个自动机，它需要构建一个转移函数 δ ，其中 δ 为 $n \times \Sigma$ 大小的矩阵， Σ 表示字符集（本题中字符集为 a 至 j 共 10 个字符）。

$\delta_{i,j}$ 表示在第 i 个状态时接收字符 j 转移到的最大状态，妮妮想要知道这个转移矩阵 δ 具体是什么。

j 从 $0 \sim 9$ 分别代表小写字母 $a \sim j$ 。

KMP!

转移矩阵大小为 $10^6 \times 10$

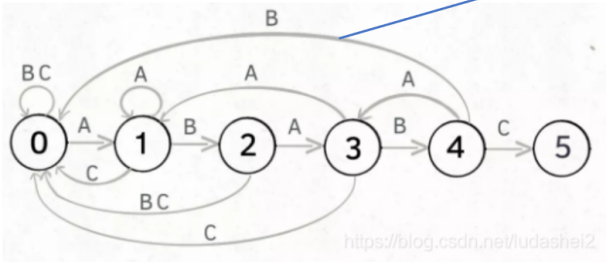
第一行一个字符串 s ，字符串长度为 n ($1 \leq n \leq 10^6$)

自动机

有限自动机是一个处理信息的简单机器，通过对文本字符串进行扫描，找出模式的所有出现位置。

例如，若想查找字符串ABABC

KMP!!!



```
vector<int> getf(char p[]){
    int lenp = strlen(p);
    vector<int> f(lenp);
    for(int i=1;i<lenp;i++){
        int k = f[i-1];
        while(k>0&&p[i]!=p[k]){
            k = f[k-1];
        }
        if(p[i]==p[k]){
            k++;
        }
        f[i]=k;
    }
    return f;
}
```

动态规划

设 $result[i][j]$ 表示在位置 i 上输入字符' $a+j$ '的结果，则易得：
 若字符恰好匹配，即 $s[i] == 'a'+j$ ， $result[i][j] = i+1$
 否则回退，即 $result[i][j] = result[f[i-1]][j]$

```
void getkun(int position){
    if(position==0){
        for(int i=0;i<10;i++){
            result[position][i] = 0;
        }
        result[position][s[0]-'a']=1;
        return ;
    }
    int k = f[position-1];
    for(int i=0;i<10;i++){
        result[position][i] = result[k][i];
    }
    result[position][s[position]-'a']=position+1;
    return ;
}
```

添加副

```
int main(){
    scanf("%s", s);
    int lens = strlen(s);
    f = getf(s);
    for(int i=0;i<lens;i++){
        getkun(i);
        for(int j=0;j<10;j++){
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

详细代码：

```
#include <iostream>
#include <vector>
#include <string.h>

using namespace std;

char s[1001000];
int result[1001000][10];
vector<int> f;

vector<int> getf(char p[]){
    int lenp = strlen(p);
    vector<int> f(lenp);
    for(int i=1;i<lenp;i++){
        int k = f[i-1];
        while(k>0&&p[i]!=p[k]){
            k = f[k-1];
        }
        if(p[i]==p[k]){
            k++;
        }
        f[i]=k;
    }
    return f;
}

void getkun(int position){
    if(position==0){
```

```
        for(int i=0;i<10;i++){
            result[position][i] = 0;
        }
        result[position][s[0]-'a']=1;
        return ;
    }
    int k = f[position-1];
    for(int i=0;i<10;i++){
        result[position][i] = result[k][i];
    }
    result[position][s[position]-'a']=position+1;
    return ;
}

int main(){
    scanf("%s", s);
    int lens = strlen(s);
    f = getf(s);
    for(int i=0;i<lens;i++){
        getkun(i);
        for(int j=0;j<10;j++){
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```