

Remember the Word

一. 考察内容:

动态规划 字符串 字典树

二. 题目分析:

[题目大意]

给一个长度为 n ($n \leq 3e5$) 的大字符串, 和 m ($m \leq 4000$) 个长度不超过100的小字符串, 求出用小字符串正好拼出大字符串的方案数。

[写题思路]

考虑用动态规划解决, 设状态 $f[i]$ 为小字符串拼成大字符串后 i 个字符的方案数。首先我们把小字符串放入字典树, 转移时逆序枚举 i , 每次我们可以用大字符串的第 i 位开始在字典树上查询 (查询长度不会超过 $i+100$ 位), 途中如果碰到一个长度为 j 的小字符串, 则会增加 $f[i+j]$ 个方案。

最后 $f[1]$ 就是整体的答案。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 09 09
*文件类型: 源代码文件
*题目来源: UVALive
*当前状态: 已通过
*备忘录: 字符串 字典树
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 300010
#define MOD 20071027
int n, f[MAXN], cnt, T;
char s[MAXN];
struct NODE
{
    int son[26], v;
    void clear(){for(int i=0; i<26; ++i) son[i]=0; v=0;}
}o[5000000];
struct Trie
{
    int root;
    void clear(){cnt=0, root=++cnt; o[root].clear();}
    void push(char s[])
    {
        int len=strlen(s), T=root;
        for(int i=0; i<len; ++i)
        {
            char c=s[i]-'a';
            if(!o[T].son[c])
            {
```

```

        o[T].son[c]=++cnt;
        o[o[T].son[c]].clear();
    }
    T=o[T].son[c];
}
o[T].v=1;
}
int query(char s[],int start)
{
    int ans=0,len=n,T=root;
    for(int i=start;i<len;++i)
    {
        // printf("vis:%d %d %c %d\n",T,i,s[i],o[T].v);
        char c=s[i]-'a';
        ans=(ans+f[i]*o[T].v%MOD)%MOD;
        if(!o[T].son[c])return ans;
        T=o[T].son[c];
    }
    return (o[T].v+ans)%MOD;
}
}
}trie;
/* Variable explain:

*/
void read()
{
    char ls1[1000];
    trie.clear();
    if(scanf("%s%d",s,&n)==-1)exit(0);
    int ls2=strlen(s);
    for(int i=1;i<=n;++i)scanf("%s",ls1),trie.push(ls1);
    n=ls2;
    return;
}
void dp()
{
    for(int i=n-1;i>=0;--i)
    {
        f[i]=trie.query(s,i);
        // printf("%d\n",f[i]);
    }
    printf("Case %d: %d\n",++T,f[0]);
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    while(1)
    {
        read();
        dp();
    }
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)