

# 秘密信息

## 一. 考察内容:

字符串 字典树

## 二. 题目分析:

[题目大意]

给出n个A类01串和m个B类01串，求出对于每个B类01串，有多少个A类01串可以与他匹配。

注：定义a、b可以匹配，即a、b的前 $\min(\text{len}(a), \text{len}(b))$ 位都是相同的。

[写题思路]

考虑用一棵字典树维护所有的A类01串，对于每个字典树的节点，存两个信息，1.以该节点结尾的01串个数（节点权值），2.包含该节点的01串个数（亦可称为该节点的子树权值之和）。

对于B类01串进行查询，从树根开始向下爬到该串的末尾，途中如果遇到节点权值不为0的，就把这些权值加入答案，爬到末尾后，再加上末尾子树的权值之和，就是该01串能匹配的个数。

## 三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 09 08
*文件类型: 源代码文件
*题目来源: BZOJ
*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;
int n,m,cnt;
struct NODE
{
    int son[2],fa,v,vtree;
    void clear(int f){son[0]=son[1]=vtree=0,fa=f;}
}o[1000000];
struct Trie
{
    int root;
    void clear(){root=++cnt;}
    void push(char s[])
    {
        int len=strlen(s),T=root;
        for(int i=0;i<len;++i)
        {
            int c=s[i]-'0';
            if(!o[T].son[c])
            {
```

```

        o[T].son[c]=++cnt;
        o[o[T].son[c]].clear(T);
    }
    T=o[T].son[c];
}
++o[T].v;
++o[T].vtree;
while(T!=root)
{
    o[o[T].fa].vtree=o[o[o[T].fa].son[0]].vtree+o[o[o[T].fa].son[1]].vtree+o[o[T]
].fa].v;//回溯维护祖先的子树信息
    T=o[T].fa;
}
}
int query(char s[])
{
    int len=strlen(s),T=root,ans=0;
    for(int i=0;i<len;++i)
    {
        int c=s[i]-'0';
        if(!T)return ans;
        ans+=o[T].v;
        T=o[T].son[c];
    }
    return ans+o[T].vtree;
}
}trie;
/* Variable explain:
*/
void read()
{
    char ls1[50005];
    int ls2;
    scanf("%d%d",&n,&m);
    trie.clear();
    for(int i=1;i<=n;++i)
    {
        scanf("%d",&ls2);
        for(int j=0;j<ls2;++j)
            scanf("%s",ls1+j);
        trie.push(ls1);
    }
    return;
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    char ls1[50005];
    int ls2;
    for(int i=1;i<=m;++i)
    {
        scanf("%d",&ls2);
        for(int j=0;j<ls2;++j)
            scanf("%s",ls1+j);
        printf("%d\n",trie.query(ls1));
    }
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)