本题的思想比较新颖，在很多面试题中会出现！！！本题实现是借鉴别人的，自己没有实现，这个题目感觉不难，就是技巧性。自己的方法也“貌似“实现了，但是结果超时。跨越的时间比较久，一直没有时间去搞算法！！  
C:\Users\Michael\AppData\Roaming\Tencent\QQ\Temp\WD$XN4L~WJZC)IAL[{`V5VN.pngC:\Users\Michael\AppData\Roaming\Tencent\QQ\Temp\WD$XN4L~WJZC)IAL[{`V5VN.pngC:\Users\Michael\AppData\Roaming\Tencent\QQ\Temp\WD$XN4L~WJZC)IAL[{`V5VN.png

（借鉴地址：<http://blog.csdn.net/u011860731/article/details/48373423>）

(一个不错有关的项目，统计有多少个单词：

<https://github.com/Michael2397/TrieTreeDemo>)

思路：

输入:

5

babaab

babbbaaaa

abba

aaaaabaa

babaababb

5

babb

baabaaa

bab

bb

bbabbaab

输出：

1

0

3

0

0

1、创建Trie数结构：

Trie{

Trie[] next = new Trie[26];//存放当前结点的子节点，相当于都要26个子节点，具体使用哪个，视情况而定。

Trie count;//记录经过了多少次当前结点，这个创新来源于原题目中

}

2、建Trie树

从根节点开始，以此找当前结点的子节点p.[ch-‘a’],判断子结点是否有值

如果没有值，则创建子节点，并且给count赋值1，修改当前结点；

如果有值，则count++，移动下一节点。

第三层

第二层

25

2

0

0

1

第一层

25

2

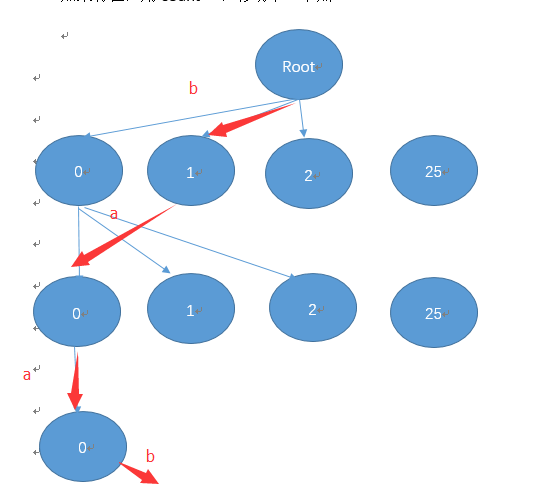
1

0

Root

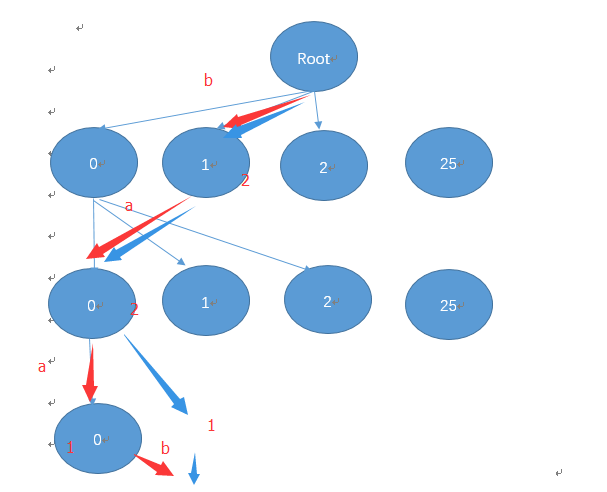
如上图所示，除了root节点以外，其他节点都有26个子节点（以空间换时间）子节点的编号从0-25，刚好是26个英文字母，第0号对应a,第1号对应b,第2号对应c,以此类推。如果进过某一个子节点，则判断该子节点是否存在，如果不存在创建，如果存在count值+1。

如第一个babaab,建树流程（红色）:



因为是第一次，所以，经过的节点上的count值是1，其他都是null.

第二个:babbbaaa(绿色):

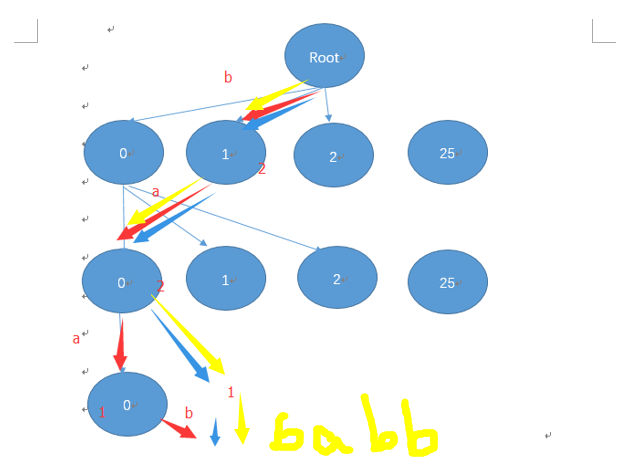


3、查询

这个方案也是题目中给的提示。

按建树的方法遍历树，p=p->next,当到达该字符串的结尾的时候，指针p也位于对应的结尾处（注意，此时p下面还有可能有子节点）。打印出当前p.count就可以得到多少个相同前缀

第一个（黄色）：babb



Babb遍历完了，此时的指针位于第四层，则第四层的值count就是有多少个单词是以babb为前缀。