

Hat's Words

一. 考察内容:

字符串 字典树 Trie

二. 题目分析:

[题目大意]

给出一些单词，求出哪些单词可以被其他两个单词正好拼接而成。

[写题思路]

这是一道字典树模板题，首先建两棵字典树，第一个字典树正序存所有单词，第二个字典树倒叙存所有单词，即“apple”在第一个字典树中存为“apple”，在第二个字典树中存为“elppa”。

对于每个字符串，分别进行查询，由于我们知道该字符串的长度，我们先在第二棵字典树中找到该字符串，随后每次让第一棵字典树向下找一个节点，第二棵字典树向上找一个节点，如果某时刻两个节点都有对应的字符串，则该单词可以被这两个字符串拼成。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 08
*文件类型: 源代码文件
*题目来源: HDU
*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 cnt, s;
string s[50001];
struct NODE
{
    int son[26], fa, v;
    char c;
    void clear(int T, int value)
    {
        for(int i=0; i<=25; ++i) son[i]=0;
        fa=T;
        v=0;
        c=value;
    }
}o[200000];
struct Trie
{
    int root;
    void clear(){root=++cnt;}
    void push(char s[], int v)
    {
        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,s[i]);
            }
            T=o[T].son[c];
        }
        o[T].v=v;
    }
}trie1,trie2;
/* Variable explain:
*/
bool query(char ls1[])
{
    char ls2[100];
    int len=strlen(ls1),T1=trie1.root,T2=trie2.root;
    for(int i=0;i<len;++i)ls2[i]=ls1[len-i-1];
    for(int i=0;i<len;++i)
    {
        int c=ls2[i]-'a';
        T2=o[T2].son[c];
    }
    // if(o[T2].v)ans=1;
    for(int i=0;i<len-1;++i)
    {
        int c=ls1[i]-'a';
        T1=o[T1].son[c];
        T2=o[T2].fa;
        // printf("%d %d %d %c %c\n",i,T1,T2,o[T1].c,o[T2].c);
        if(o[T1].v&&o[T2].v) return true;
    }
    return false;
}
void read()
{
    char ls1[100];
    trie1.clear();
    trie2.clear();
    while(scanf("%s",ls1)==1)
    {
        s[++_s]=ls1;
        trie1.push(ls1,1);
        int len=strlen(ls1);
        for(int i=0;i<len>>1;++i)swap(ls1[i],ls1[len-i-1]);
        trie2.push(ls1,1);
    }
    return;
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    char ls1[100];
    sort(s+1,s+1+_s);
    for(int i=1;i<=_s;++i)
    {
        int len=s[i].length();
        for(int j=0;j<len;++j)ls1[j]=s[i][j];
        ls1[len]=0;
        if(query(ls1))printf("%s\n",ls1);
    }

    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)