

trie树的指针实现

1471：【例题1】Phone List

【题目描述】 原题来自：POJ 3630 给定 n 个长度不超过 10 的数字串，问其中是否存在两个数字串 S, T ，使得 S 是 T 的前缀，多组数据。【输入】 第一行一个整数 T ，表示数据组数。对于每组数据，第一行一个数 n ，接下来 n 行输入 n 个数字串。【输出】 对于每组数据，若存在两个数字串 S, T ，使得 S 是 T 的前缀，则输出 NO，否则输出 YES。请注意此处结果与输出的对应关系！【输入样例】 2 3 911 97625999 91125426 5 113 12340 123440 12345 98346 【输出样例】 NO YES 【提示】 数据范围：对于 100% 的数据， $1 \leq T \leq 40, 1 \leq n \leq 104$ 。【来源】 http://ybt.ssoier.cn:8088/problem_show.php?pid=1471

思路

- (1) 先构建一棵trie树，这是一棵10叉树,如下面的程序insert，每次单词的结束会涂上一个颜色
- (2) 按照每个单词的路径，去查询一次find，在查询的路径中有2个以上涂过颜色的点，那么就返回true了
- (3) 因为有多组测试数据，所以每组数据都建立的一棵树，用完以后要把树删除del_tree

源代码

```
1 #include<iostream>
2 #include<cstdio>
3 #include<cstring>
4 using namespace std;
5 int t,n;
6 bool ans;
7 string s1,a[10005];
8 struct Node
9 {
10     bool is_col;
11     Node* ch[10];
12 };
13 Node *root,*nil;
14 void insert()
15 {
16     int len=s1.size();
17     Node *p=root;
18     for(int i=0;i<len;i++)
19     {
20         int bh=s1[i]-'0';
21         if(p->ch[bh]==nil)
22         {
23             p->ch[bh]=new Node;
24             p->ch[bh]->is_col=false;
25             for(int j=0;j<10;j++)
26             {
27                 p->ch[bh]->ch[j]=nil;
28             }
29         }
30         p=p->ch[bh];
31     }
32     p->is_col=true;
33 }
34 int find(string s){
```

```

35     int len=s.size();
36     int ans=0;
37     Node * p=root;
38     for(int i=0;i<len;i++){
39         int bh=s[i]-'0';
40         if(p->ch[bh]->is_col==true) ans++;
41         p=p->ch[bh];
42     }
43     return ans;
44 }
45 bool search(){
46     for(int i=1;i<=n;i++){
47         int cnt=find(a[i]);
48         if(cnt>=2) return true;
49     }
50     return false;
51 }
52 void solve()
53 {
54     root=new Node;
55     nil=new Node;
56     root->is_col=false;
57     for(int i=0;i<10;i++)
58     {
59         root->ch[i]=nil;
60     }
61     ans=false;
62     cin>>n;
63     for(int i=1;i<=n;i++)
64     {
65         cin>>s1;
66         a[i]=s1;
67         insert();
68     }
69     if(search())
70     {
71         printf("NO\n");
72     }
73     else
74     printf("YES\n");
75 }
76 void del_tree(Node * p){
77     if(p==nil) return;
78     for(int i=0;i<10;i++)
79     {
80         del_tree(p->ch[i]);
81     }
82     delete p;
83 }
84 int main()
85 {
86     scanf("%d",&t);
87     for(int i=1;i<=t;i++)
88     {
89         del_tree(root);
90         solve();
91     }
92     return 0;
93 }

```

