AC自动机

by 李翔

AC自动机

- 不是accept自动机。

- 是Aho-Corasick 造出来的。所以你懂的

用途

- 字符串的匹配问题
- 多串的匹配问题

- 例如给几个单词 acbs，asf,dsef,
- 再给出一个 很长的文章，acbsdfgeasf
- 问在这个文章中，总共出现了多少个单词，或者是单词出现的总次数。

实现的原理

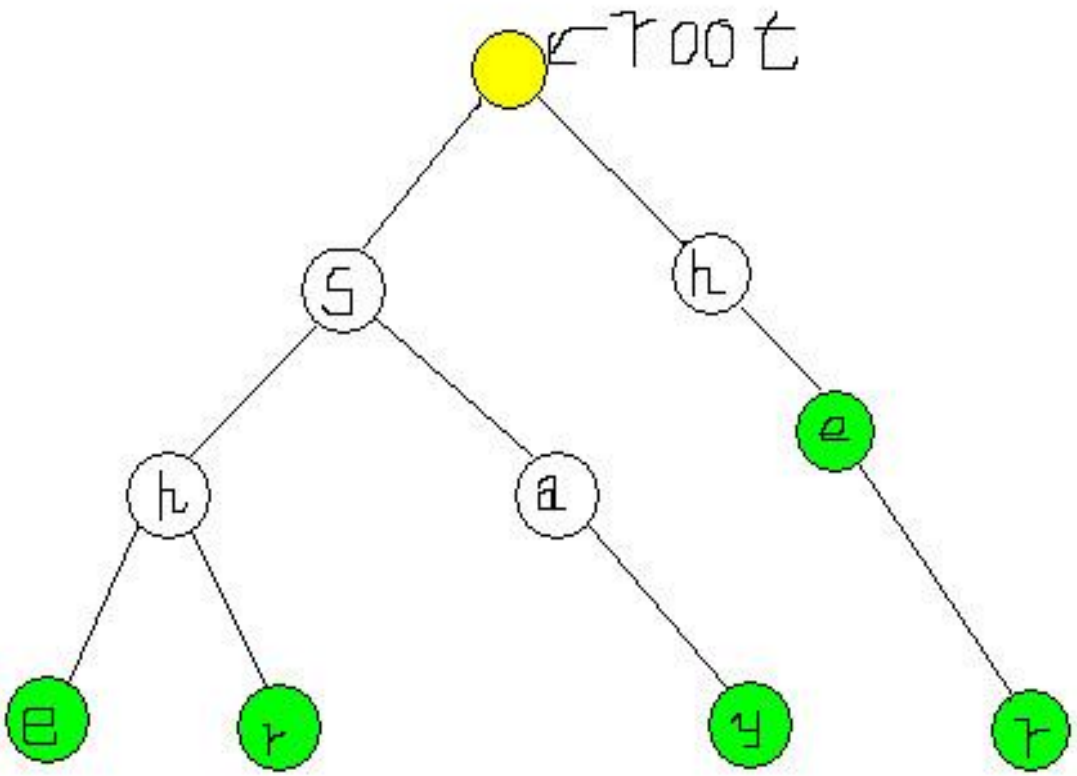
- 形象的说：KMP+trie树（字典树）

  KMP，之前学过的。为什么用到了这个呢？

  因为也是一种没有多余回溯的算法。

  什么是trie树（字典树）？

Trie树



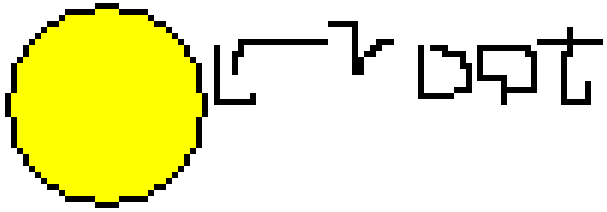
有神马特点

- 有一个空的根节点。

- 对于一般的trie数是解决用英文字母组成文章。所以每个节点会有26个子节点（指针）。

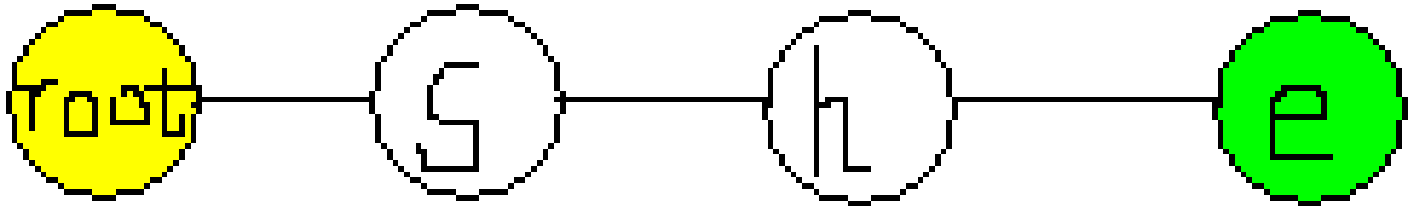
构造的过程

- 开始的时候有一个root



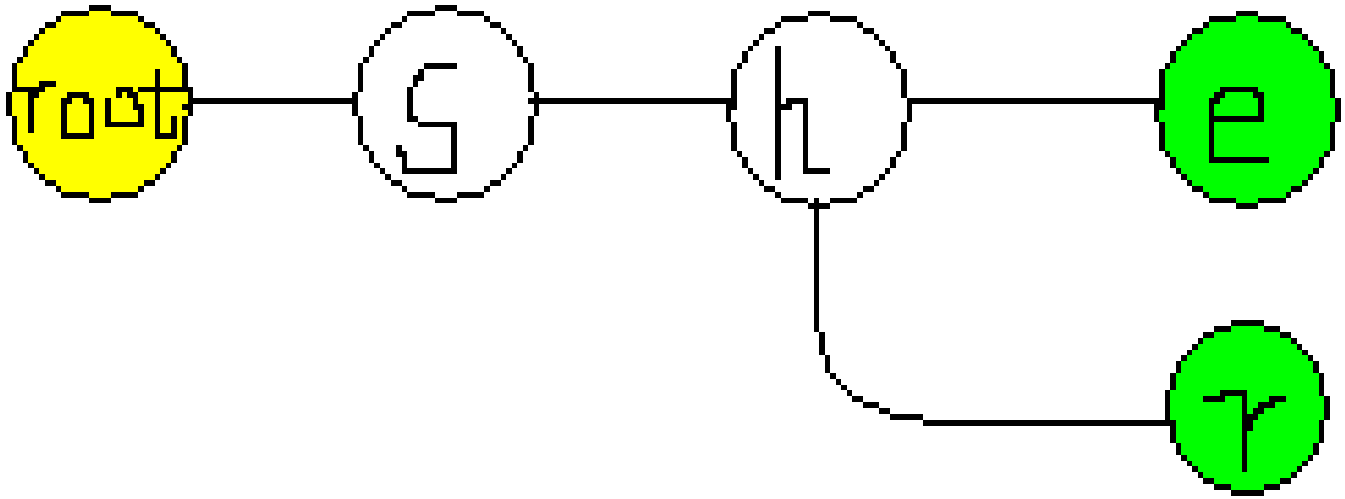
过程

- 要在该树中插入一个单词she



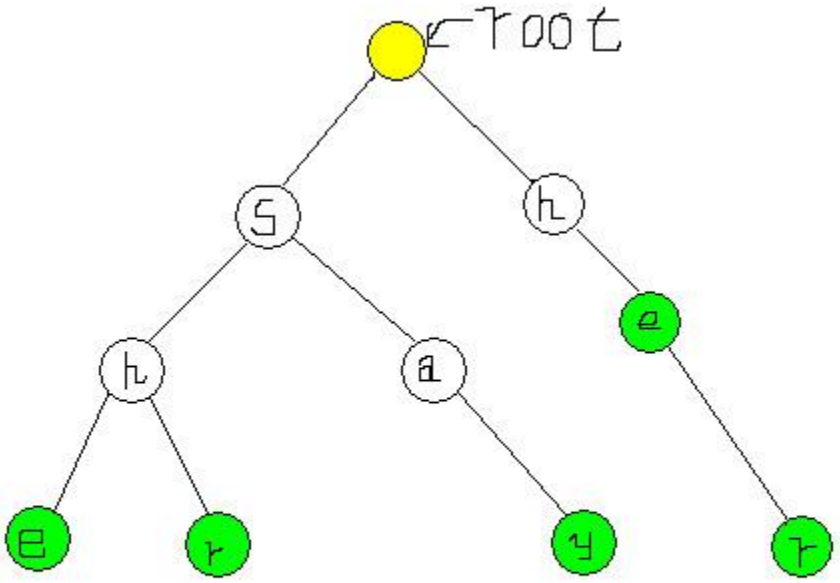
过程

- 再加一个单词shr。



过程

- 再插入say和her等，这样一个trie树就搞定了



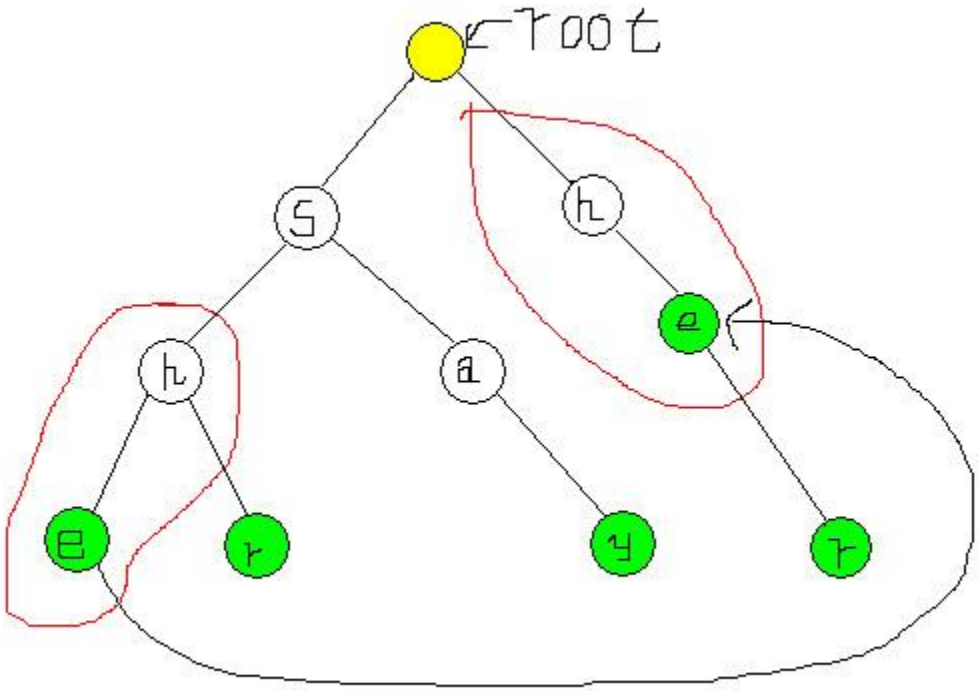
如何与kmp联系在一起？

- 关键是在trie树上 加了一种fail指针。

- Fail指针的用途：就像是kmp中的next的数组。
- 在字符串失配的时候确定转移的节点。

先看到底是什么样的

- 这只显示了e的失配指针。
- 例如匹配文章：sher

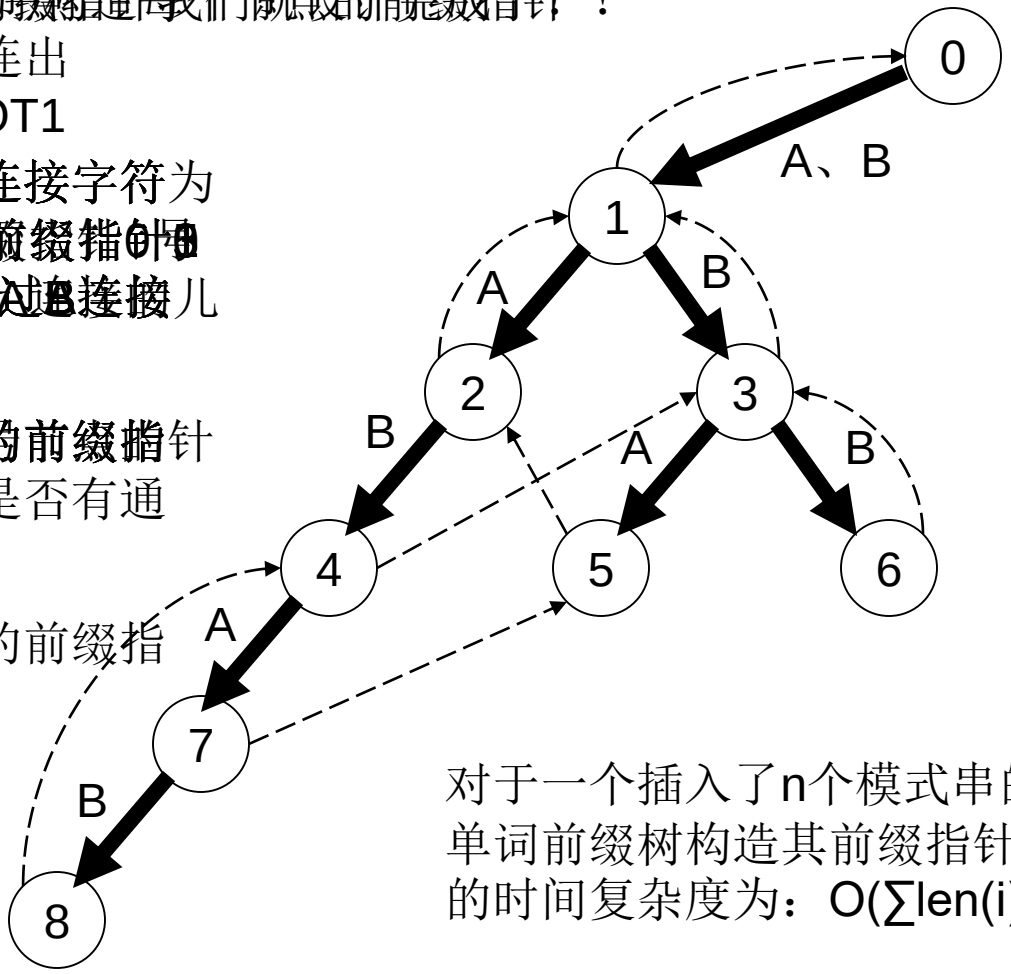


构造fail指针的原理

- 根据父亲节点的fail指针来构造子节点的fail指针。

- 动态的过程：下面是拷贝人家的ppt！编号为0的节点可以忽略之。

如何高效的构造前缀指针



对于一个插入了n个模式串的单词前缀树构造其前缀指针的时间复杂度为：O(∑len(i))

如何解决开始给的问题

在所有的绿色节点上做上标记。一旦访问到这个节点，就记录下。这样就能解决上面的问题。

hdu2222

5 //单词数
she //单词
he
say
shr
her
Yasherhs//文章

问有多少单词在文章中出现

代码实现

```
struct node
{
    int next[26];
    int fail;
    int count;
    void init()
    {
        memset(next, -1, sizeof(next));
        fail = 0;
        count = 0;
    }
}s[500005];
```

在树中插入单词

```
void ins()
{
    int len = strlen(str);
    int i, j, ind;
    for(i = ind = 0; i < len; i++)
    {
        j = str[i] - 'a';
        if(s[ind].next[j] == -1)
        {
            s[sind].init();
            s[ind].next[j] = sind++;
        }
        ind = s[ind].next[j];
    }
    s[ind].count++;
}
```

在这里的操作对于不同的题目一般有3种不同的操作。

1：s[ind].count++;

这个是在解决出现总次数的时候是这样处理的。

2：s[ind],count=1;

这个是在ac自动机上进行dp的时候经常用的。

3.新加一个标记id。

这个是在处理有哪些单词出现过。

```
void make_fail()
{
    qin = qout = 0;
    int i, ind, ind_f;
    for(i = 0; i < 26; i++)  {
        if(s[0].next[i] != -1) {
            q[qin++] = s[0].next[i];
        }
    }
    while(qin != qout)   {
        ind = q[qout++];
        for(i = 0; i < 26; i++)   {   //找之后的子节点
            if(s[ind].next[i] != -1)   {
                q[qin++] = s[ind].next[i];
                ind_f = s[ind].fail;
                while(ind_f > 0 && s[ind_f].next[i] == -1)
                    ind_f = s[ind_f].fail;
                if(s[ind_f].next[i] != -1)
                    ind_f = s[ind_f].next[i];
                s[s[ind].next[i]].fail = ind_f;//子节点的fail根据父节点fail指针的搞定
            }
        }
    }
}
```

```
int fd()  {
    int ct = 0;
    int di, i, ind, p;
    int len = strlen(des);//这个是文章
    for(di = ind = 0; di < len; di++)   {
        i = des[di] - 'a';
        while(ind > 0 && s[ind].next[i] == -1)
            ind = s[ind].fail;

        if(s[ind].next[i] != -1) {//等于-1的时候就已经是找打了根节点。
            ind = s[ind].next[i];
            p = ind;
            while(p > 0 && s[p].count != -1) {//这里是精髓。在找过某个有标记的节点的时候
                ct += s[p].count;  //答案            //会把该位的标记标记为-1，在下次经过有-1
                s[p].count = -1;                    //标记的时候，说明之后的都被计算过，不用
                p = s[p].fail;                      //再重复计算了。
            }
        }
    }
    return ct;
}
```

拓展

在自动机上进行dp

要大家自己去理解

题目

Poj
1204
 4052（题目在4044上下载）
Hdu
2222
3065