

原创

后缀trie树 & 后缀自动机

2019-09-13 11:23:25 _Y_-Y_- 阅读数 16 更多

编辑

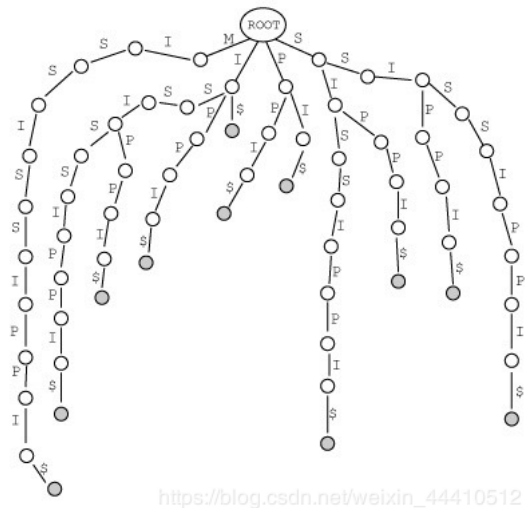
版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/weixin_44410512/article/details/100799519

后缀trie树

后缀trie树 就是在一个字典树里有且只有一个单词的所有后缀
列出字符串 *MISSISSIPPI* 的所有后缀

1	MISSISSIPPI
2	ISSISSIPPI
3	SSISSIPPI
4	SISSIPPI
5	ISSIPPI
6	SSIPPI
7	SIPPI
8	IPPI
9	PPI
10	PI
11	I

后缀trie树就是把所有后缀全部存进字典树里面



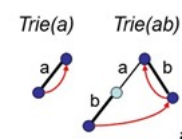
接下来是构造后缀trie树
第一种很简单面，就是按照定义将所有后缀依次插入到字典树里面
第二种就是一个很神奇的操作
比如说 *abaab*

1	abaab
2	baab
3	aab
4	ab
5	b

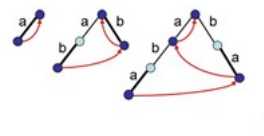
先把所有后缀右对齐，然后从上到下开始操作
先开始把第一个字符插进去



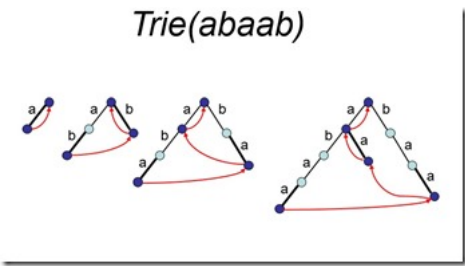
红色箭头的指向的节点和子叶节点都是下一个字符要插入的点



然后在该节点下插入该字符，并且让红色箭头指向刚才插入的那个节点



要是红色箭头下面有要插入的字符，那就不必新建节点，直接指向该节点

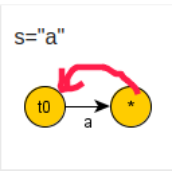


后缀自动机 (SAM)

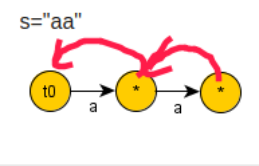
后缀自动机是一张有向无环图，其中顶点是状态，而边代表了状态之间的转移。

初始状态被记作 t_0 ，终止状态用星号(*)标记

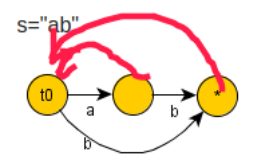
失配指针fail指向的节点 是 当前节点 (top) 所表示的字符串的最大后缀



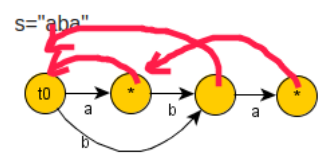
从起点开始构建，第一个字符直接插入在根节点后面，失配指针指向根节点



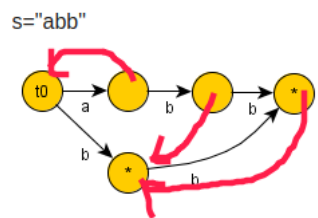
因为第二个节有最大后缀 a 所以不需要在建一个新的节点



要是没有最大后缀，那就只能从根节点出发，连接该节点

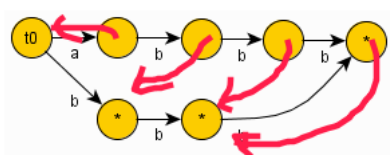


你会发现从根节点出发到标(*)节点的所有路径表示的就是这个字符串的所有后缀



https://blog.csdn.net/weixin_44410512

现在就是用到fail的时候了
要是第二次因为同一条边而fail到同一个节点，那么这个节点就得往下新建一个节点
新建的这个节点就有了第一次访问这个节点的性质，
并且让当前节点指向和第一次节点的fail指向新建的节点
完善这个新建的节点



fail指针主要还是用来完善新建节点的，并且加速图的建立

构造后缀自动机思路：
从当前字符(end)的前一个(last)开始跑fail，要是fail指向的节点不存在这个字符，那么将两个节点相连
跑到结束的时候会有两种情况
第一种是跑的了根节点(0)所指的fail，那么直接将根节点和最后节点(end)相连
第二种就是fail指向的节点下有这个字符，那么需要判断这个节点所表示的字符串是不是与(end)所表示的字符串相同
要是不相同，就要新建一个节点(np)，新建的这个节点将保存先前访问过这个节点的信息
然后继续跑fail 要是fail指向的节点存在这个这个字符，那么连接节点(np)
目的就是找最长后缀相等的前缀

code

```
1  /*
2  Glass Beads
3  POJ - 1509
4  https://cn.vjudge.net/problem/POJ-1509#author=hzo12017_gyz
5  题意：给一个字符串S，每次可以将它的第一个字符移到最后面，求这样能得到的字典序最小的字符串。 如BBAAB，最小的就是AABBB
6  解法：后缀自动机
7  */
8  #include <iostream>
9  #include <algorithm>
10 #include <cstring>
11 #define maxn 100000
12 using namespace std;
13 struct node{
14     int len,fail;
15     int next[26];
16     void init(){
17         fail=-1;
18         len=0;
19         memset(next,-1,sizeof next);
```

```
20     }
21 }a[maxn];
22 int top,last;
23 char s[maxn];
24 void init(){
25     top=last=0;
26     a[top++].init();
27 }
28 int newnode(){
29     a[top].init();
30     return top++;
31 }
32 void add(int c){
33     int end=newnode();
34     int now=last;
35     a[end].len=a[last].len+1;
36     for(;now!=-1&&a[now].next[c]==-1; now=a[now].fail){
37         a[now].next[c]=end;
38     }
39     if(now==-1) a[end].fail=0;
40     else{
41         int next=a[now].next[c];
42         if(a[now].len+1==a[next].len) a[end].fail=next;
43         else{
44             int np = newnode();
45             a[np]=a[next];
46             a[np].len=a[now].len+1;
47             a[end].fail=a[next].fail = np;
48             for(;now!=-1&&a[now].next[c]==next;now=a[now].fail){
49                 a[now].next[c] = np;
50             }
51         }
52     }
53     last = end;
54 }
55 int main(){
56     int T;
57     cin>>T;
58     while(T--){
59         init();
60         cin>>s;
61         int len=strlen(s);
62         for(int i=0;i<len*2;i++){
63             add(s[i%len]-'a');
64         }
65         int ans=0;
66         for(int i=0;i<len;i++){
67             for(int j=0;j<26;j++){
68                 if(a[ans].next[j]!=-1){
69                     ans=a[ans].next[j];
70                     break;
71                 }
72             }
73         }
74         cout<<a[ans].len-len+1<<endl;
75     }
76 }
```

更多例题[在这里](#)

有 0 个人打赏

文章最后发布于: 2019-09-13 13:33:54

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客

猿衣酷

专属于程序员的卫衣

关闭