[**后缀树(Suffix Tree)**](http://kmplayer.javaeye.com/blog/558439)

**文章分类:**[**C++编程**](http://www.javaeye.com/blogs/category/cpp)

1,后缀树（Suffix tree）是一种数据结构，能快速解决很多关于字符串的问题。后缀树提出的目的是用来支持有效的字符串匹配和查询。   
简单点说，后缀树就是将一个给定字符串的所有后缀全部压入一个Trie,然后将只有单个叶子的节点压缩，从而形成的一棵树.   
  
2,后缀树的用途，总结起来大概有如下几种   
(1). 查找字符串o是否在字符串S中。   
方案：用S构造后缀树，按在trie中搜索字串的方法搜索o即可。   
原理：若o在S中，则o必然是S的某个后缀的前缀。   
例如S: leconte，查找o: con是否在S中,则o(con)必然是S(leconte)的后缀之一conte的前缀.有了这个前提，采用trie搜索的方法就不难理解了。   
(2). 指定字符串T在字符串S中的重复次数。   
方案：用S+’$'构造后缀树，搜索T节点下的叶节点数目即为重复次数   
原理：如果T在S中重复了两次，则S应有两个后缀以T为前缀，重复次数就自然统计出来了。   
(3). 字符串S中的最长重复子串   
方案：原理同2，具体做法就是找到最深的非叶节点。   
这个深是指从root所经历过的字符个数，最深非叶节点所经历的字符串起来就是最长重复子串。   
为什么要非叶节点呢?因为既然是要重复，当然叶节点个数要>=2。   
(4). 两个字符串S1，S2的最长公共部分   
方案：将S1#S2$作为字符串压入后缀树，找到最深的非叶节点，且该节点的叶节点既有#也有$(无#)。   
  
3,后缀树的构造:O(n)   
给定一个txt:`mississippi'   
(1)得到txt的所有后缀:   
T1  = mississippi = txt   
T2  = ississippi   
T3  = ssissippi   
T4  = sissippi   
T5  = issippi   
T6  = ssippi   
T7  = sippi   
T8  = ippi   
T9  = ppi   
T10 = pi   
T11 = i   
T12 =               (empty)   
(2)将所有非空后缀进行排序,得到:   
T11 = i   
T8  = ippi   
T5  = issippi   
T2  = ississippi   
T1  = mississippi   
T10 = pi   
T9  = ppi   
T7  = sippi   
T4  = sissippi   
T6  = ssippi   
T3  = ssissippi   
(3)将所有后缀的公有前缀进行合并,即可得到:   
  
           tree                       substrings   
  
tree-->|---mississippi                m .. mississippi   
       |   
       |---i-->|---ssi-->|---ssippi   i .. ississippi   
       |       |         |   
       |       |         |---ppi      issip,issipp,issippi   
       |       |   
       |       |---ppi                ip, ipp, ippi   
       |   
       |---s-->|---si-->|---ssippi    s .. ssissippi   
       |       |        |   
       |       |        |---ppi       ssip, ssipp, ssippi   
       |       |   
       |       |---i-->|---ssippi     si .. sissippi   
       |               |   
       |               |---ppi        sip, sipp, sippi   
       |   
       |---p-->|---pi                 p, pp, ppi   
               |   
               |---i                  p, pi   
  
--- Suffix Tree for "mississippi" ---   
  
4,应用   
  
问题:   
寻找字符串S的最长回文字符串.   
注:所谓“回文”是指当一个字符串正序读和逆序读时都一样。即p=reverse(p)   
  
分析:   
等价于寻找S与reverse(S)的最长公共部分.