[<<  树状数组，线段树简介](http://kinslovertec.blogbus.com/logs/43784719.html) | [首页](http://kinslovertec.blogbus.com) | [滚动数组  >>](http://kinslovertec.blogbus.com/logs/43786080.html)

* **后缀树学习 - [**[**ACM**](http://kinslovertec.blogbus.com/c2708662/)**]**

2009-08-09

[版权声明](http://creativecommons.org/licenses/by/3.0/deed.zh)：转载时请以超链接形式标明文章原始出处和作者信息及[本声明](http://bangzhuzhongxin.blogbus.com/logs/11205960.html)  
<http://kinslovertec.blogbus.com/logs/43784965.html>

后缀树是一种特殊的字典树，但是从抽象程度，应用面，使用，构造的难度上都比字典树高的多了，它的应用我熟悉的有三种，据说还有更强大的功能：

1.查找一个字符串s是否包含字串t：【字典树同法】

后缀树中包含所有s的后缀，则若t是s的一个子串，则它必是每个后缀分支的前缀，我们按照字典树的查找方式，查是否存在一个后缀包含这个t就可以了。

2.统计s中子串t出现的次数：【找叶子数】

要查找t出现的次数，我们需选择一条前缀是t的后缀路径，归纳说明如下：扩展插入后缀串时，当出现第一个t前缀时，我们有一个后缀串ts，这时我们将此后 缀ts插入，则此串在树中的前端必然是t，之后ts再扩展，开头前缀不为t，则插入时对ts这条路径t部分以后无影响，这时扩展到一定时间又出现了t前缀 的tss串，这时我们插入，一开始必然沿着t插入，但过了t后就不一定了，这个后缀串tss必然不可能和ts是一个串（长度就不可能等），所以最终会分叉 成两个，下面归纳：在任意一个时刻插入一个以t为前缀的后缀串，其t必为新出现的，树中不含的，且插入后含t为前缀的后缀路径在t以后必会多出一个子路 径，即多出一个叶子。由于这两个过程是同步的，我们只需要从t向后遍历找到t以后的叶子数目就是t在s中出现的数目了。

3.找出s中最长的重复子串，重复字串指出现2次以上的子串：

【暂时不理解】