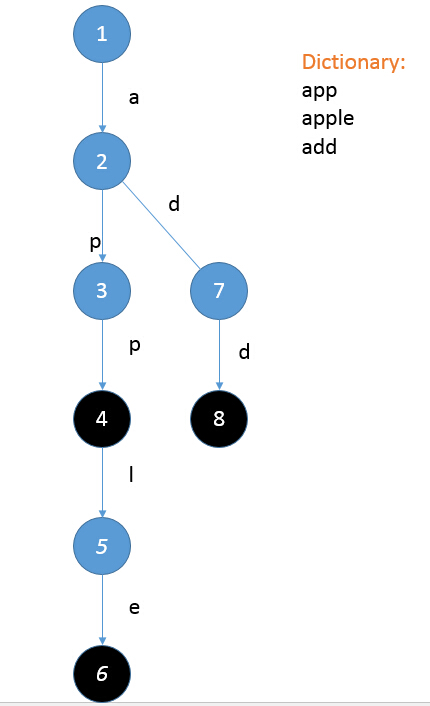
小Hi于是在纸上画了一会，递给小Ho，道：“你看这棵树和这个词典有什么关系？”



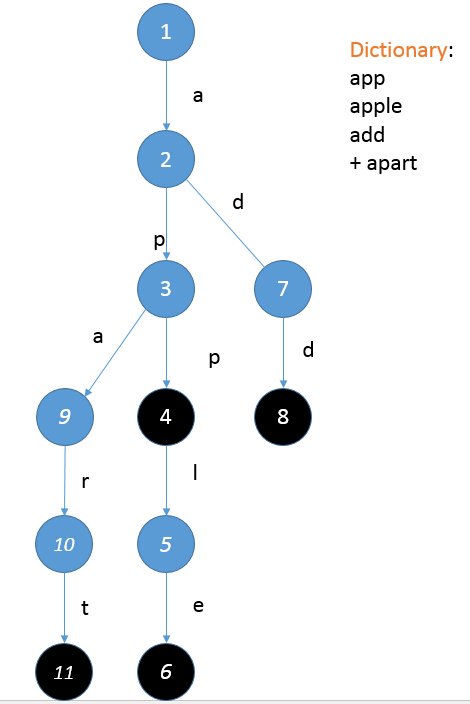
小Ho盯着手里的纸想了一会道：“我知道了！**对于从树的根节点走到每一个黑色节点所经过的路径，如果将路径上的字母都连起来的话，就都对应着词典中的一个单词呢！**”

小Hi说道：“那你知道如何根据一个词典构建这样一棵树么？”

“不造！”

“想你也不知道，我来告诉你吧~”小Hi摆出一副老师的样子，说道：“你先这么想，如果我已经有了这样的一个词典和对应的一棵树，我要添加一个新的单词apart，我应该怎么做？”

“让我想想……”小Ho又开始苦思冥想：“首先我要先看看已经能走到哪一步了对吧？比如我从1号节点走"a"这一条边就可以走到2号节点，然后从2号节点走"p"这一条边可以走到3号节点，然后……就没路可走了！这时候我就需要添加一条从3号节点出发且标记为"p"的边才可以接着往下走……最后就是这样了！然后我把最后到达的这个结点标记为黑色就可以了。”



小Hi说道：“真聪明~那你不妨再算算如果是一个有10W个单词的词典，每个单词的长度不超过10的话，这棵树会有多大？”

小Ho于是掏出笔来，一边画一遍念叨：“假设我已经将前三个单词构成了这样一棵树，那么我要添加一个新的单词的时候，最坏情况是这个单词和之前的三个单词都没有公共前缀，那么这个新的单词的长度如果是5的话，我就至少要添加5个结点到树中才能够继续表示这个词典！”

“而如果每次都是最坏情况的话，这棵树最多也就100W个结点这么大！更何况最坏情况是不可能次次都发生的！毕竟字母表也才26个字母呢！”小Ho继续说道。

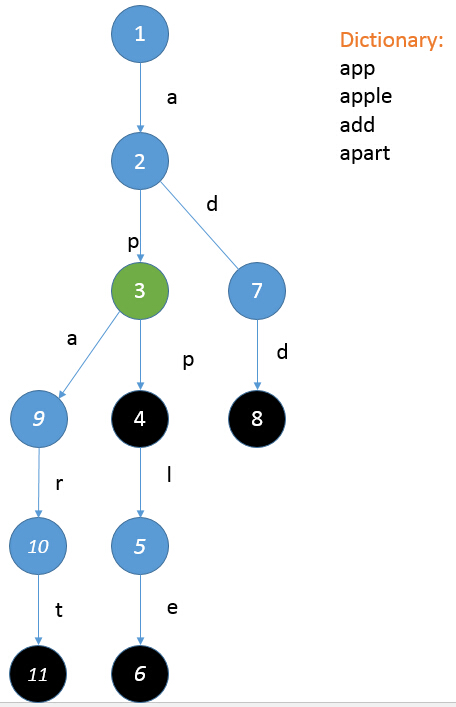
“嗯~这样我们是不是就可以用（单词个数\*单词长度）个结点来表示一个词典了呢？小Hi问道。

“是的呢！”小Ho道：“但是这样一棵树又有什么用呢？”

“可别小看了它，它就是传说中的Trie树哦~至于他有什么用，一会你就知道了！”小Hi笑嘻嘻的回答道。

提示二：如何使用Trie树

小Hi在树上用绿色标出了一个节点，递给小Ho。



“这个结点……是从根节点先走"a"然后走"p"到达的结点呢！哦~~我知道了，以这个结点为根的子树里所有标记结点都是以"ap"为前缀的单词呢！而且所有以"ap"为前缀的单词都在以这个节点为根的子树里~”小Ho惊喜道。

“是的呢~那你对怎么解决我的问题有想法了么？”小Hi追问道。

“唔...那就是每次拿到你的字符串之后，我在树上找到其对应的那个结点，然后统计这个节点中有多少个标记节点？”小Ho不是很确定的答道：“但是这样...似乎在最坏情况，也就是你每次给个字符串都很短的时候，我还是要扫描这棵树的很大一部分呢？也就是说虽然平均时间复杂度降低了，但是最坏情况时间复杂度还是很高的样子！”

小Hi笑嘻嘻道：”没想到你自己看出来了呢~我还以为又要教训你了！~那你有什么好的解决方法么？”

“没呢！小Hi你就别卖关子了，赶紧告诉我吧！”被折磨的够呛的小Ho开始求饶。

“好吧！就帮你这一回~”

“小Ho你有没有想过这样一个问题？**不妨称以T为根的子树中标记节点的个数为L[T]，既然我要统计某个L[T1]，，而这个结点是不确定的，我有没有办法一次性把所有结点的L[T]求出来呢？**”小Hi整理了下思绪，问道。

“似乎是有的，老师以前说过，递归什么的。。”小Ho答道。

“递归太复杂了！我们可以之后再说，你这么想，在你构建Trie树的时候，当你经过一个结点的时候，说明了什么？”小Hi撇了撇头，继续问道。

“我想想，经过一个结点……标记结点……说明了以这个结点为根的子树中将要多出来一个标记结点？”

“没错！那你有没有什么办法来记录这个改变呢？”

“我想想，**我在最开始置所有L[T]=0，然后每次添加一个新的单词的时候，都将它经过的所有结点的L[T]全部+1，这样我构建完这棵Trie树的时候，我也就能够同时统计到所有L[T]了**，对么？”小Ho开心道。