EX5说明文档及学习心得

1. 写函数前发现代码中似乎有些错误，报错提醒是找不到current和T.root的出处；查了网上发现又是和模版有关系。很多东西不用模版写不会出错，但是一用模版写，由于模版的种种特性导致有很多需要注意的地方。我查到的内容是这么说的：

“因为有偏特化，所以一个模板子类其实是不能在实例化之前就知道他的模板父类到底是谁，因此名字也无法resolve； 对于一个非依赖型名称（不以任何方式依赖于模板参数的名称），在模板声明进行解析的时候就会进行查找。但C++标准中规定，一个非受限的名称查找的时候将不会考虑依赖型的基类。所以查找不到就会错误。解决办法是把它变成一个依赖型名称：this 、B<T>是依赖型名称。“

1. 具体代码要实现的功能是：让结点只入栈、出栈两次，找current(当前结点)的后继----即其在后序遍历中的下一个结点。
2. 我的思路是：用poptime记录结点出栈次数，有0、1、2三种结果。用p记录操作时的当前结点、w是poptime+p实际放入栈中的栈结点。当前结点刚入栈，poptime为0，此时让左孩子入栈，当前结点出栈，poptime更新为1，此时让右孩子入栈，当前结点出栈，poptime更新为2。这是后序遍历结点只入栈、出栈两次的基本过程。
3. 而要找当前结点的后继则可以看作是，current传入该函数，让栈顶结点出栈来进行后序遍历找到符合要求的结点。因此有三个判断条件poptime==0、1、2。poptime为2时，表示当前结点恰好出栈2次，即为所求，更新current为q，结束函数；poptime为0时，此时准备让左孩子入栈，左孩子空则让当前结点出栈又重新入栈，poptime+1；poptime为2时，此时准备让右孩子入栈，右孩子空则让当前出栈，poptime+1。
4. 比较麻烦的是循环内开始的操作，由于根首次进来的时候栈内是空的，所以要分两种情况判断：进来的结点是根结点时，将p赋值为w的Node，让结点入栈；进来的结点不是根结点时，让栈顶结点出栈，更新w、p，poptime+1，若此时poptime为2则获得所需结点，不入栈；若此时poptime为1则继续将该结点放入栈中，作后续条件的判断和操作。
5. 最后就是这些代码的安放顺序：结束的判断条件放在最上面，避免符合条件后又做一次开始的操作；然后就是分两种情况的开始操作；接着就是poptime==0、1的判断条件和相应操作。