EX3\_task1说明文档及学习心得

1. 这题的关键是用二叉树来实现。由于有出栈和入栈两种选择（状态），可以将入栈看为左子树，出栈看为右子树。
2. 先画出n=3的二叉树情况，发现边界条件是：没有未处理的元素——即所有元素都已经入栈或者出栈。因此用变量i来记录轮到哪个元素入栈。
3. 我觉得递归算法的一大难点是——在结点之间回退要保护现场和恢复现场。用m记录刚出栈元素，方便回退。用j记录所有出栈元素，方便最后打印。
4. 进入下一次递归前要保护现场，递归函数结束前要恢复现场：

f(用于回退的参数)

{

保护现场

↓

f(用于回退的参数);//递归函数

↓

恢复现场

}

1. 还有一点要注意的是，f函数以栈stk为参数，由于栈中包含指针，参数传递和返回值赋值处都调用了拷贝构造函数，此时就必须用深拷贝，需要自己重写一个，将指针指向的值一一对应的赋值过去。