递推与递归

**递推与递归所涉及的内容是动态规划思想与分治策略的基础。**

**递归：**程序调用自身。

**递推：**从初值出发反复进行某一运算得到所需结果。从已知到未知，从小到大(比如每年长高9cm，20年180，30年270)。

**迭代：**是重复反馈过程的活动，其目的通常是为了逼近所需目标或结果。每一次对过程的重复成为一次“迭代”，而每一次迭代得到的结果会作为下一次迭代的初始值。

**回溯法：**从一条路往前走，能进则进，不能进则退回来，换一条路再试。八皇后问题就是回溯算法的典型。回溯法与深度优先遍历有相同点也有不同点。

1. 递推思想
2. 知道递推式，也知道初始条件，从初始条件开始往上顺推直到求得目标解的思想就是**递推思想**。
3. 当就某个问题能写出递推式、能确定初始(边界)条件，那么可以考虑使用递推。对于某些数据规模很大的递推任务可以使用**矩阵加速**提升效率。
4. 递归思想
5. 构造函数，这个函数在运行过程中调用自己，从而解决问题的思路就称为**递归思想**。
6. **快速排序**使用到了递归思想。
7. “记忆化搜索”：每个数字最多只计算一次，一旦计算完成就会被存下来，便于日后使用。

递推和递归思想往往可以相互转换。有的情况下进行递推，需要求出初始条件，还需要确定递推顺序(尤其是多维递推时更加复杂)，所以这时使用递归思想会容易一些。

如果能将一个大的任务分解成若干规模较小的任务，而且这些任务的形式与结构和原问题一致，就可以考虑使用递归。当问题规模足够小或者达到了边界条件就要停止递归。分解完问题后还要对这些规模小的任务合并然后返回解，最后逐级上报，解决最大规模的问题。有些问题使用递推策略和递归策略都能解决，但有些问题只能将大问题分割成小问题，但是却很难建立递推式，在这种情况下应当使用递归策略。