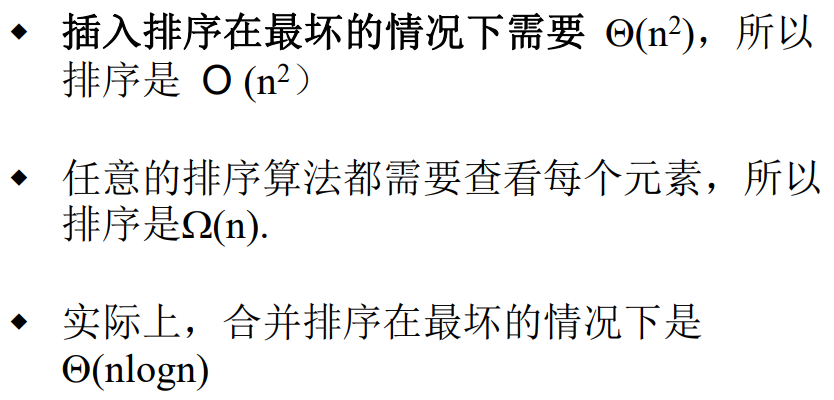
算法是指**解决问题的一种方法或一个过程**。

算法：**输入、输出、确定性、有限性**

程序是**算法用某种程序设计语言的具体实现**。可以不满足**有限性**

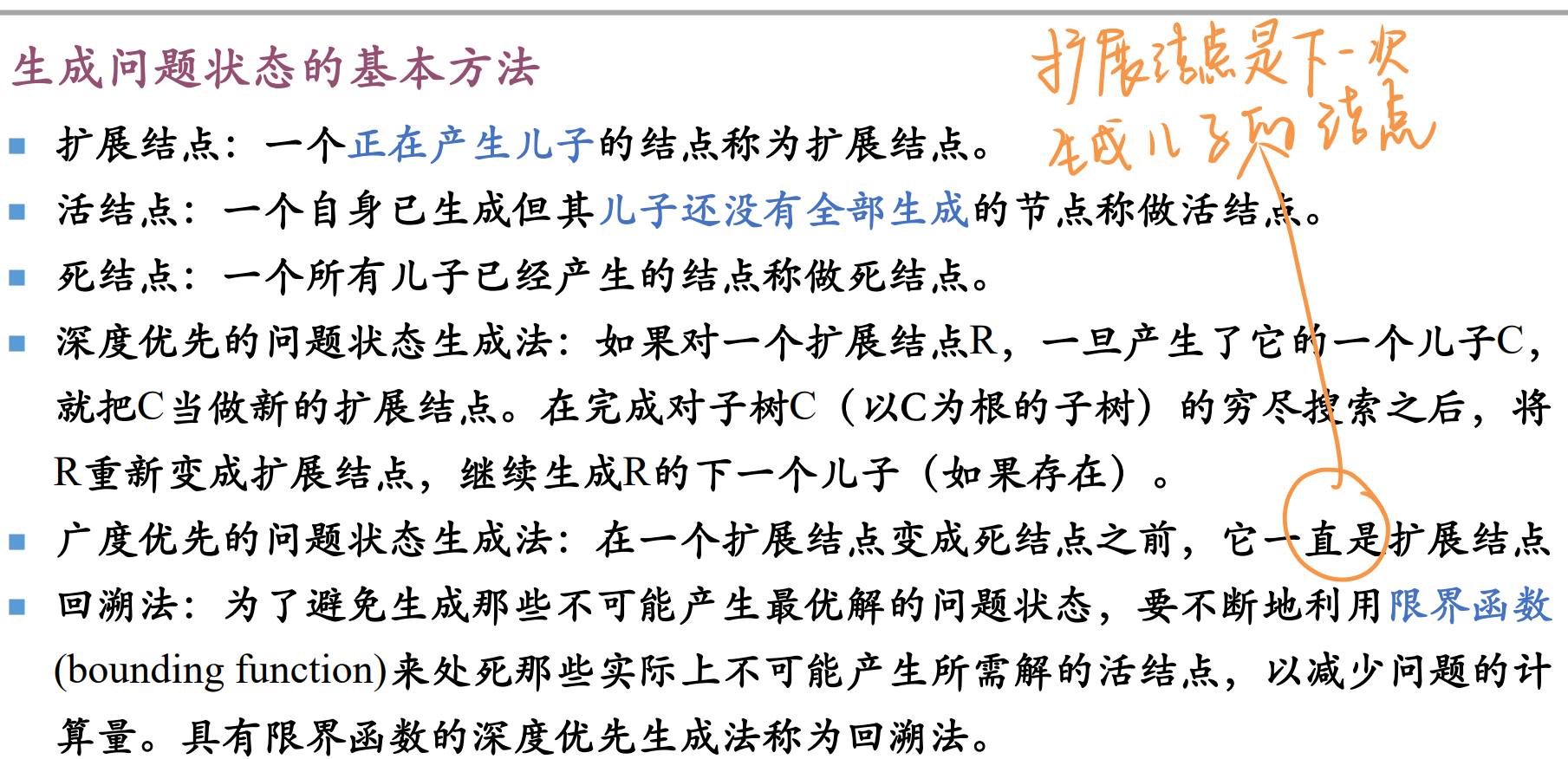
描述算法的形式有**自然语言、程序流程图、伪代码**

算法复杂性分析有**事后统计**和**事前分析**

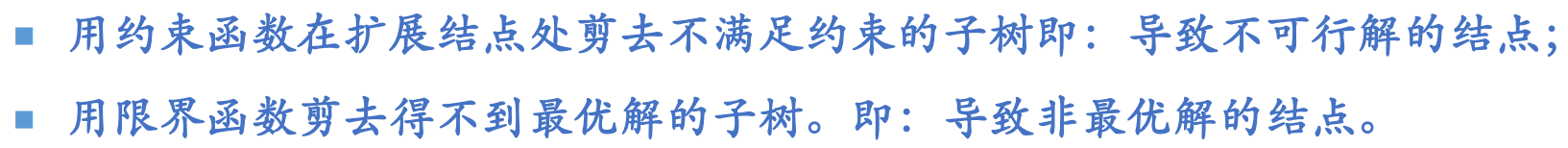
****

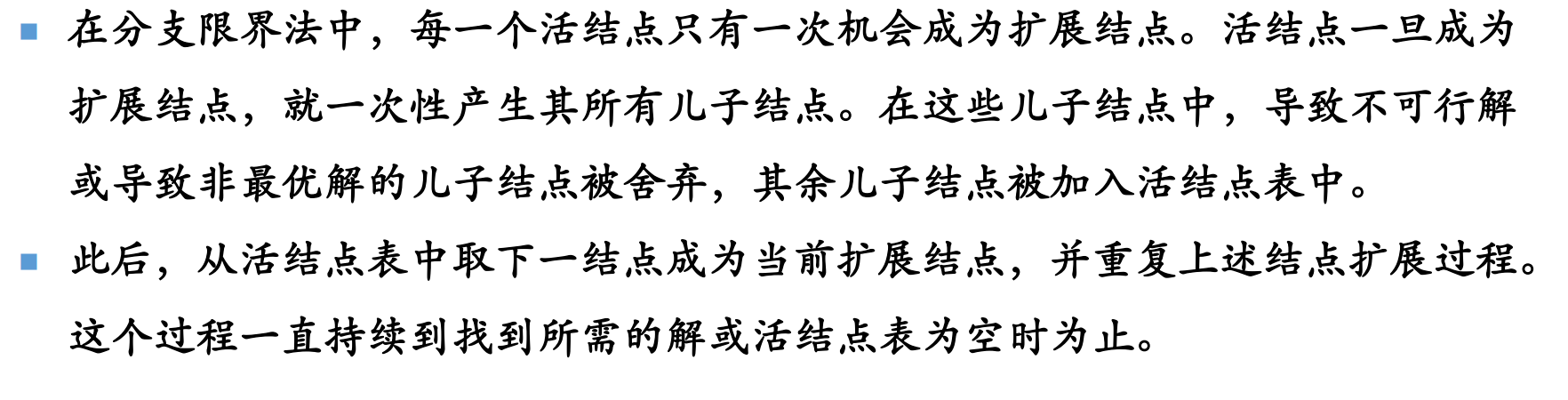
最优子结构是指**问题的最优解中包含子问题的最优解**

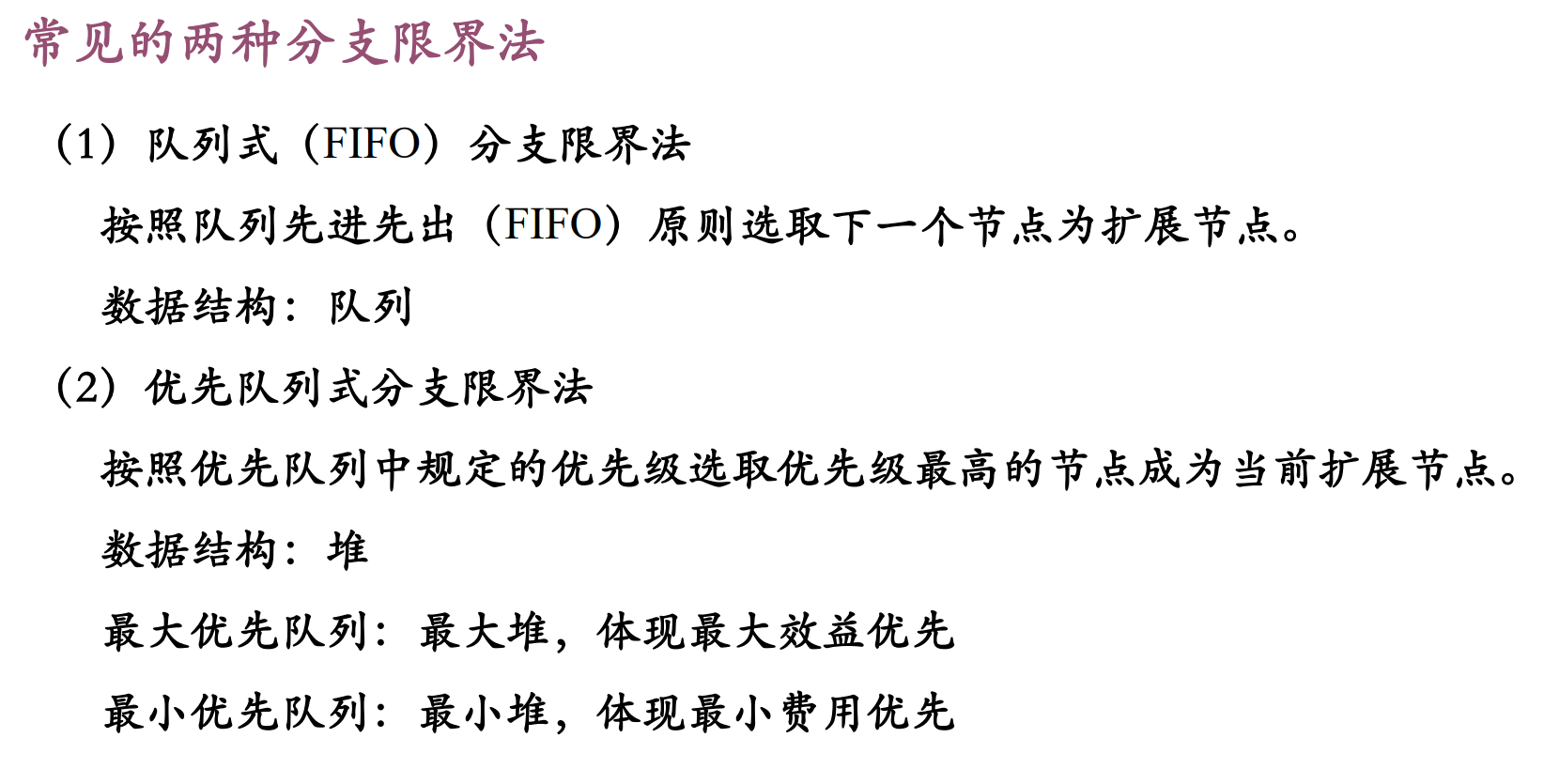
贪心选择性质是指**所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到。**

****

**约束函数和限界函数都是剪枝函数**

****

****

****

**动态规划和分治法的区别**

1.子问题重叠性

分治算法：子问题之间相互独立，没有重叠的部分。

动态规划算法：子问题之间存在重叠的部分，即不同的子问题可能会多次使用相同的中间结果。

2.求解顺序

分治算法：通常采用自顶向下的递归方式，先求解较大的子问题，再合并得到最终解。

动态规划算法：通常采用自底向上的迭代方式，按照阶段顺序从初始阶段开始逐步计算解决每个阶段的状态，直到求解出最终的问题。

3.时间复杂度

分治算法：依赖于问题规模的指数幂，通常情况下时间复杂度较高。

动态规划算法：通过保存中间状态，避免了大量的重复计算，时间复杂度通常相对较低。

**动态规划与贪心策略**

动态规划和贪心算法都是一种递推算法，均有最优子结构性质，通过局部最优解来推导全局最优解。

两者之间的区别在于：

贪心算法中作出的每步贪心决策都无法改变，因为贪心策略是由上一步的最优解推导下一步的最优解，而上一步之前的最优解则不作保留，贪心算法每一步的最优解一定包含上一步的最优解。动态规划算法中全局最优解中一定包含某个局部最优解，但不一定包含前一个局部最优解，因此需要记录之前的所有最优解。

动态规划算法通常以自底向上的方式解各子问题，而贪心算法则通常以自顶向下的方式进行，以迭代的方式作出相继的贪心选择，每作一次贪心选择就将所求问题简化为规模更小的子问题。

**回溯法与分支限界法**

1. 求解目标：回溯法的求解目标是找出解空间树中满足约束条件的所有解，而分支限界法的求解目标则是找出满足约束条件的一个解，或是在满足约束条件的解中找出在某种意义下的最优解。
2. 搜索方式的不同：回溯法以深度优先的方式搜索解空间树，而分支限界法则以广度优先或以最小耗费优先的方式搜索解空间树。

分治法：分解为若干个规模较小的相同问题

动态规划：分解为若干个规模较小的子问题，子问题不独立

贪心策略：局部最优策略能导致产生全局最优解

回溯法：找出所有解集或满足约束条件的最优解

分支限界法：找出最优解

掌握动态规划算法的基本要素

（1）最优子结构性质

（2）重叠子问题性质

掌握贪心算法的基本要素

（1）最优子结构性质

（2）贪心选择性

分治法的基本步骤

（1）解决小规模的问题

（2）分解问题

（3）递归的解各子问题

（4）将各子问题的解合并为原问题的解

动态规划算法的步骤

（1）找出最优解的性质，并刻画其结构特征

（2）递归地定义最优值

（3）以自底向上的方式计算出最优值

（4）根据计算最优值时得到的信息，构造最优解。

分支限界法算法步骤

（1）定义解空间

（2）确定易于搜索的解空间结构（二叉树）

（3）确定结点的数据结构

（4）确定约束条件与限界条件用于剪枝

（5）确定活结点组织方法（队列/优先队列（优先级函数））

（6）确定答案节点识别方法

回溯法算法步骤

（1）针对所给问题，定义问题的解空间；

（2）确定易于搜索的解空间结构；

（3）以深度优先方式搜索解空间，并在搜索过程中用剪枝函数（约束条件、限界条件）避免无效搜索。

递归

◼优点：结构清晰，可读性强，而且容易用数学归纳法来证明算法的正确性，因此它为设计算法、调试程序带来很大方便。

◼缺点：递归算法的运行效率较低，无论是耗费的计算时间还是占用的存储空间都比非递归算法要多

分治法所能解决的问题一般具有以下几个特征：

◼ 该问题的规模缩小到一定的程度就可以容易地解决；

◼ 该问题可以分解为若干个规模较小的相同问题;

◼ 利用该问题分解出的子问题的解可以合并为该问题的解；

◼ 该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题。