P 问题是那些能够在多项式时间内解决的问题，通常能找到高效的确定性算法。
NP 问题是那些给定解后可以在多项式时间内验证解是否正确的问题，尚未知道是否能在多项式时间内找到解。
常见的 NP 问题包括：旅行商问题、背包问题、图着色问题等。

分治法

基本思想:将一个规模为 n 的问题分解为 k 个规模较小的子问题，这些子问题互相独立且与原问题相同。递归地解这些子问题，然后将各子问题的解合并得到原问题地解。

动态规划

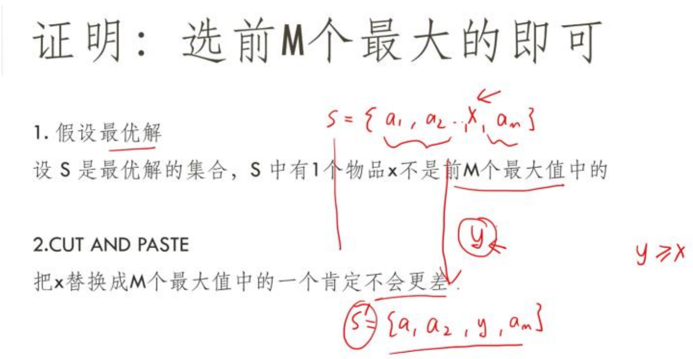
动规基本要素:
（1）最优子结构性质
（2）重叠子问题性质

步骤:
① 找出最优解的性质，并刻画其结构特征
② 递归的定义最优质
③ 以自底向上的方式计算最优值
④ 根据计算最优值时得到的信息，构造最优解

贪心

基本要素：1.贪心选择性质 2.最优子结构性质
贪心选择性质: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。
最优子结构性质：一个问题的最优解包含其子问题的最优解。

如何证明贪心选择性质?
CUT AND PATSE
1.假设一个与贪心策略违背的解 2.通过贪心策略构造一个更优的解



动态规划与贪心的比较:

分治法与动态规划法的异同

相同点

1. 都通过分解子问题求解：都将原问题分解成若干个子问题，然后逐步求解。

2. 适合最优子结构问题：两种方法都能用于具有最优子结构性质的问题，即整体最优解可以通过子问题的最优解来构建。

3. 需要定义子问题边界：两者在设计时都需要明确子问题的边界和递归关系。

不同点

1. 子问题的独立性：

- 分治法适合于子问题相互独立的问题，不会重复计算子问题。
- 动态规划法适合于子问题重叠的情况，通过存储子问题的解来避免重复计算。

2. 计算方式：

- 分治法通常使用递归自顶向下解决问题。
- 动态规划法多使用自底向上填表的方式，也可以通过递归加记忆化来实现。

3. 时间复杂度：

- 分治法的时间复杂度较高，尤其在重复子问题较多的情况下效率不佳。
- 动态规划法通过表格记录子问题解，时间复杂度通常较低，适合求解重叠子问题的情况。

4. 是否保证最优解：

- 分治法不一定总是得到全局最优解，特别是分解和合并的过程可能不会保证最优性。
- 动态规划法通常用于求解全局最优解问题，通过重叠子问题和最优子结构的性质构建最优解。

分治、动规、贪心比较:

4. 总结比较

特性	分治法	动态规划法	贪心算法
适用问题类型	子问题独立，能分解成小的子问题	子问题重叠，需要避免重复计算	子问题可以通过局部最优解构建全局最优解
计算方式	递归分解问题，合并子问题的解	递推计算每个子问题的解，利用状态转移方程	逐步选择局部最优解，不回溯
是否保证最优解	不一定，依赖于问题的结构	保证最优解	不一定，依赖于问题的贪心性质
例子	归并排序、快速排序、矩阵乘法	背包问题、最长公共子序列、最短路径问题	活动选择问题、最小生成树、霍夫曼编码

影响回溯的因素:

① 产生 x[k]的时间

② 满足显约束的 x[k]值的个数

③ 计算约束函数的时间

④ 计算上界函数 Bound()的时间

⑤ 满足约束函数和上界函数约束的所有 x[k[的个数

分支限界

基本思想：广度优先或以最小耗费(最大收益)优先的方式搜索解空间树。

- 队列式(**FIFO**)分支限界法：按队列先进先出（**FIFO**）原则选取下一节点为扩展节点。
- 优先队列式分支限界法：按照优先队列中规定的优先级选取优先级最高的节点成为当前扩展节点。

回溯与分支限界对比：
回溯法：适合需要穷举所有解的场景，采用深度优先搜索，适合组合问题。它的特点是遍历所有可能的路径，直到找到所有解。
分支限界法：适合优化问题，采用广度优先搜索或优先级广度优先搜索，通过剪枝策略避免不必要的计算，优先搜索潜力较大的分支，最终找到最优解或一个解。

随机化算法

1. 数值概率算法
主要思想：利用随机化方法求得近似解（非精确解）。
特征：
a) 通过多次随机试验，得到一个近似的解。
b) 适用于求解复杂问题，特别是那些没有快速精确解法的问题。
c) 结果的准确性依赖于试验次数，试验次数越多，结果越接近真实值。
典型例子：模拟退火算法（Simulated Annealing）、遗传算法（Genetic Algorithm）。

2. 舍伍德思想（Sherwood's Idea）
主要思想：总能求解问题的解，且求得的解总是正确的。
特征：
a) 舍伍德思想通常用于优化问题，通过随机化方法找到一个接近最优的解。
b) 通过随机选择和调整，逐步逼近最优解。
c) 适用于大规模优化问题，特别是那些传统方法难以处理的问题。
典型例子：随机梯度下降（Stochastic Gradient Descent）。

3. 拉斯维加斯算法（Las Vegas Algorithm）
主要思想：解一定是正确的，但有可能得不到解。
特征：
a) 保证解的正确性，但不保证在有限时间内找到解。
b) 如果找到解，解一定是正确的；如果没有找到解，算法可能会继续运行或终止。
c) 适用于那些解唯一且难以找到的复杂问题。
典型例子：快速选择算法（Quickselect）用于找到数组的第 k 大元素。

4. 蒙特卡罗算法（Monte Carlo Algorithm）

主要思想：在 P 正确一致的前提下，通过随机化方法求得正确解。
特征：
a) 通过随机试验和统计方法，得到问题的解。
b) 解的正确性依赖于试验次数和概率分布。
c) 适用于数值积分、优化、统计模拟等问题。
d) 解的准确性可以通过增加试验次数来提高。
典型例子：蒙特卡罗积分（Monte Carlo Integration）、蒙特卡罗模拟（Monte Carlo Simulation）。

总结
数值概率算法：利用随机化方法求得近似解（非精确解）。

舍伍德思想：总能求解问题的解，且求得的解总是正确的。
线性时间选择算法

拉斯维加斯算法：保证解的正确性，但有时找不到解。
N 后问题、整数因子分解

蒙特卡罗算法：在 P 正确一致的前提下，通过随机试验和统计方法，得到正确解。