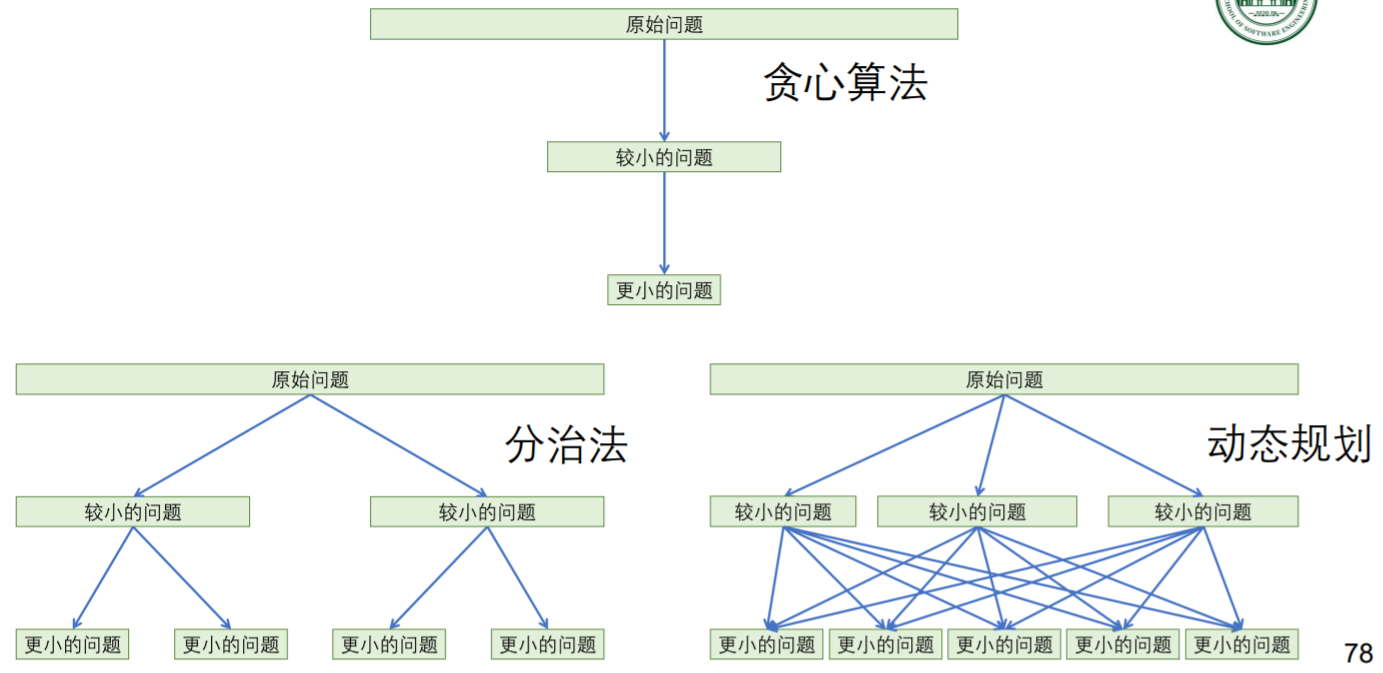
**常见问题**

➢ 无界限背包问题

➢ 活动选择问题

➢ 调度问题 （考察每个任务的 单位时间内等待代价/执行时间 ）

➢ 最优编码问题



**贪心算法**：每次找到一个最优的解法，直接化简成唯一的较小问题，再也不回头。可能不是最优结果。

**动态规划**(Dynamic Programming)：分解为较小的问题，但较小问题的解法可能互相关联

* 记忆化(Memorization)搜索：保存子问题结果，避免重复计算。
* 填表法(Tabulation)：自底向上计算，用先解决的小问题的结果去解决大问题。

例子及其代码实现：

* 贝尔曼-福特算法的动态规划版本维护了一个最短路径表。
* 所有顶点对最短路径-弗洛伊德算法(Floyd-Warshall)：



* 最长公共子序列 (Longest Common Subsequence, LCS)

