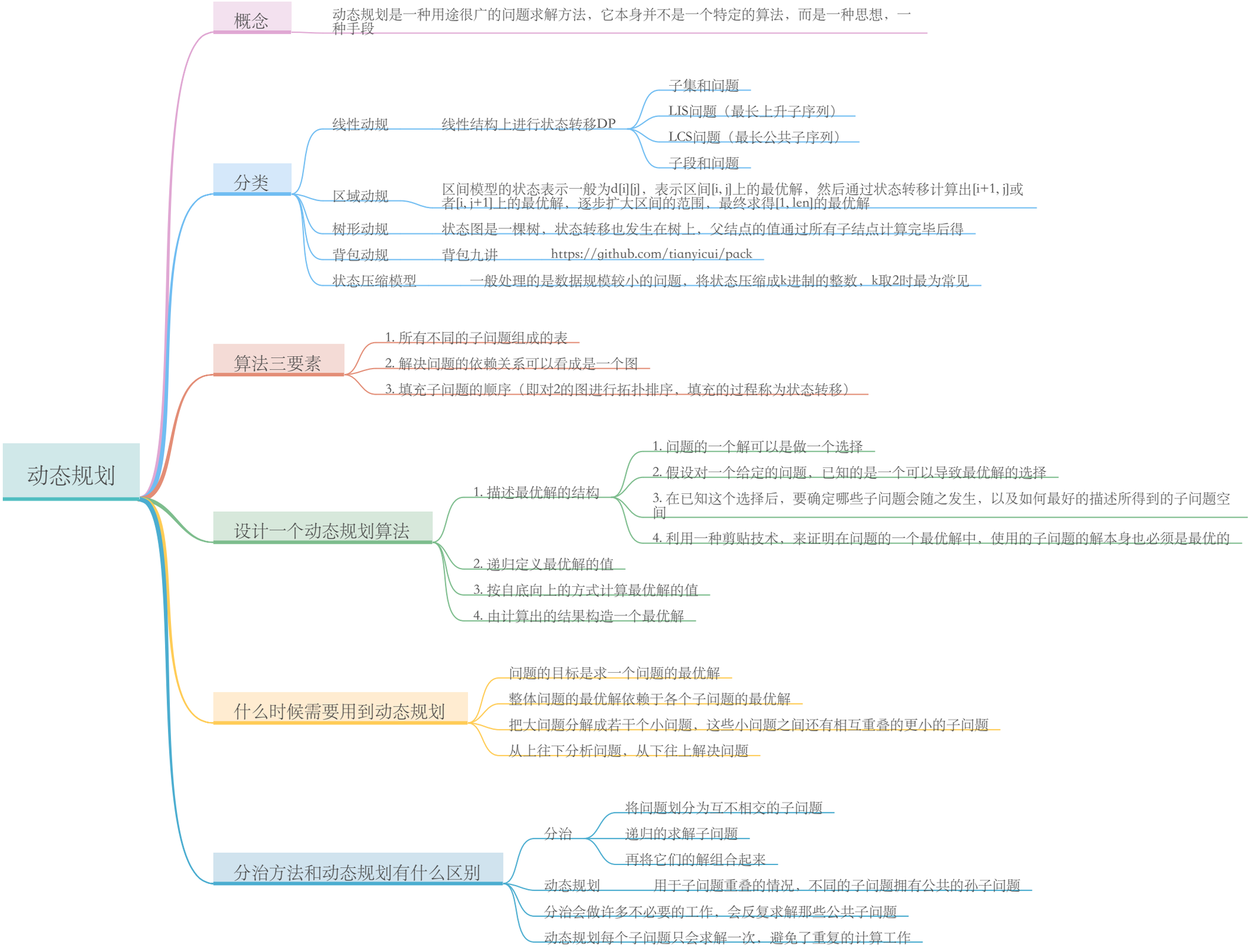
动态规划
概念
动态规划是一种用途很广的问题求解方法，它本身并不是一个特定的算法，而是一种思想，一种手段
分类
线性动规
线性结构上进行状态转移DP
子集和问题
LIS问题（最长上升子序列）
LCS问题（最长公共子序列）
子段和问题
区域动规
区间模型的状态表示一般为d[i][j]，表示区间[i, j]上的最优解，然后通过状态转移计算出[i+1, j]或者[i, j+1]上的最优解，逐步扩大区间的范围，最终求得[1, len]的最优解
树形动规
状态图是一棵树，状态转移也发生在树上，父结点的值通过所有子结点计算完毕后得
背包动规
背包九讲
https://github.com/tianyicui/pack
状态压缩模型
一般处理的是数据规模较小的问题，将状态压缩成k进制的整数，k取2时最为常见
算法三要素
1. 所有不同的子问题组成的表
2. 解决问题的依赖关系可以看成是一个图
3. 填充子问题的顺序（即对2的图进行拓扑排序，填充的过程称为状态转移）
设计一个动态规划算法
1. 描述最优解的结构
1. 问题的一个解可以是做一个选择
2. 假设对一个给定的问题，已知的是一个可以导致最优解的选择
3. 在已知这个选择后，要确定哪些子问题会随之发生，以及如何最好的描述所得到的子问题空间
4. 利用一种剪贴技术，来证明在问题的一个最优解中，使用的子问题的解本身也必须是最优的
2. 递归定义最优解的值
3. 按自底向上的方式计算最优解的值
4. 由计算出的结果构造一个最优解
什么时候需要用到动态规划
问题的目标是求一个问题的最优解
整体问题的最优解依赖于各个子问题的最优解
把大问题分解成若干个小问题，这些小问题之间还有相互重叠的更小的子问题
从上往下分析问题，从下往上解决问题
分治方法和动态规划有什么区别
分治
将问题划分为互不相交的子问题
递归的求解子问题
再将它们的解组合起来
动态规划
用于子问题重叠的情况，不同的子问题拥有公共的孙子问题
分治会做许多不必要的工作，会反复求解那些公共子问题
动态规划每个子问题只会求解一次，避免了重复的计算工作