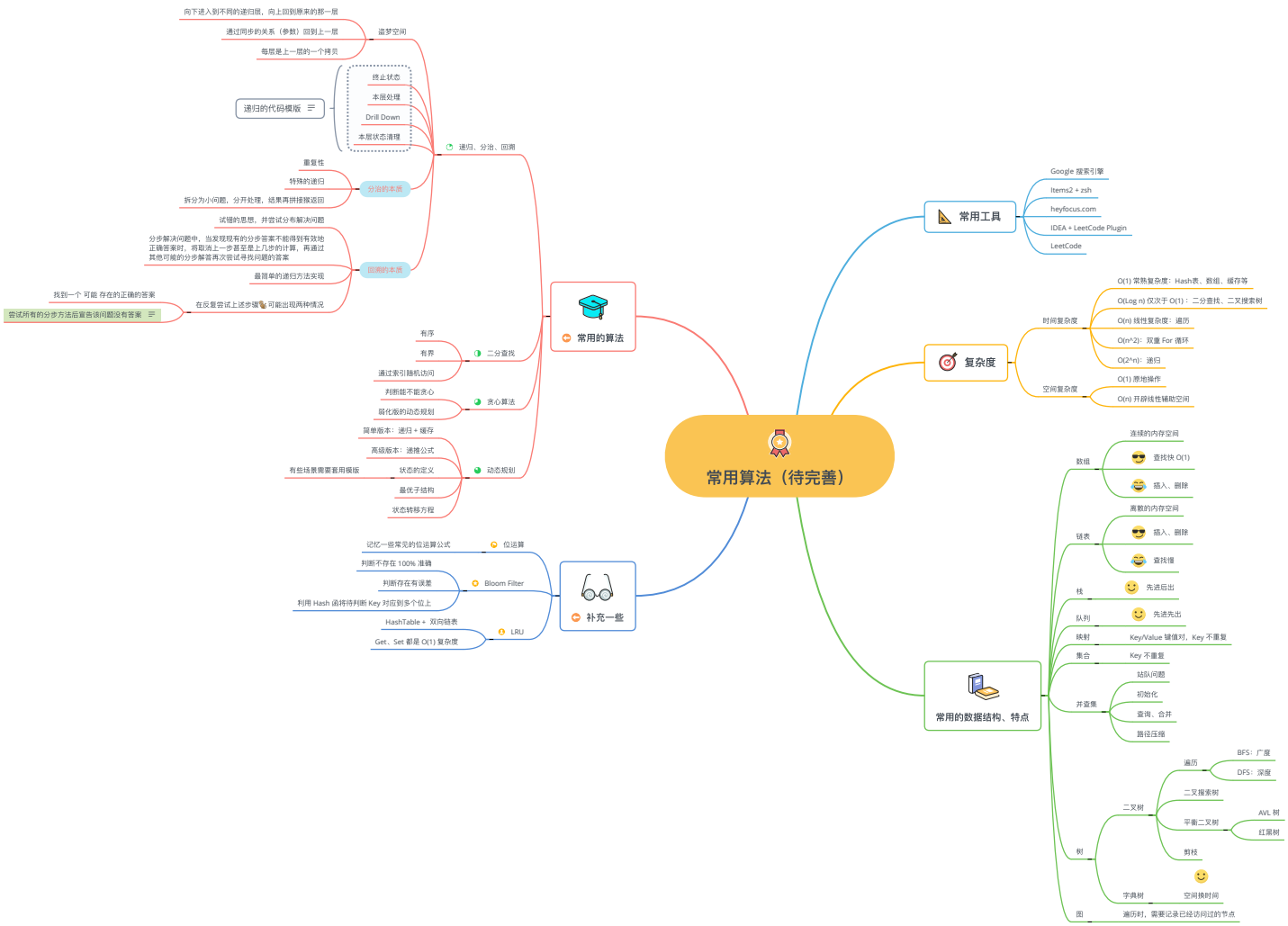
常用算法（待完善）
常用的算法
递归、分治、回溯
递归的代码模版
终止状态
本层处理
Drill Down
本层状态清理
盗梦空间
向下进入到不同的递归层，向上回到原来的那一层
通过同步的关系（参数）回到上一层
每层是上一层的一个拷贝
分治的本质
重复性
特殊的递归
拆分为小问题，分开处理，结果再拼接报返回
回溯的本质
试错的思想，并尝试分布解决问题
分步解决问题中，当发现现有的分步答案不能得到有效地正确答案时，将取消上一步甚至是上几步的计算，再通过其他可能的分步解答再次尝试寻找问题的答案
最简单的递归方法实现
在反复尝试上述步骤🐂可能出现两种情况
找到一个 可能 存在的正确的答案
尝试所有的分步方法后宣告该问题没有答案
二分查找
有序
有界
通过索引随机访问
贪心算法
判断能不能贪心
弱化版的动态规划
动态规划
简单版本：递归 + 缓存
高级版本：递推公式
状态的定义
有些场景需要套用模版
最优子结构
状态转移方程
补充一些
位运算
记忆一些常见的位运算公式
Bloom Filter
判断不存在 100% 准确
判断存在有误差
利用 Hash 函待待判断 Key 对应到多个位上
LRU
HashTable + 双向链表
Get、Set 都是 O(1) 复杂度
常用工具
Google 搜索引擎
Items2 + zsh
heyfocus.com
IDEA + LeetCode Plugin
LeetCode
复杂度
时间复杂度
O(1) 常熟复杂度：Hash表、数组、缓存等
O(Log n) 仅次于 O(1)：二分查找、二叉搜索树
O(n) 线性复杂度：遍历
O(n^2)：双重 For 循环
O(2^n)：递归
空间复杂度
O(1) 原地操作
O(n) 开辟线性辅助空间
常用的数据结构、特点
数组
连续的内存空间
查找快 O(1)
插入、删除
链表
离散的内存空间
插入、删除
查找慢
栈
先进后出
队列
先进先出
映射
Key/Value 键值对，Key 不重复
集合
Key 不重复
并查集
动态问题
初始化
查询、合并
路径压缩
树
二叉树
遍历
BFS：广度
DFS：深度
二叉搜索树
平衡二叉树
AVL 树
红黑树
剪枝
字典树
空间换时间
图
遍历时，需要记录已经访问过的节点