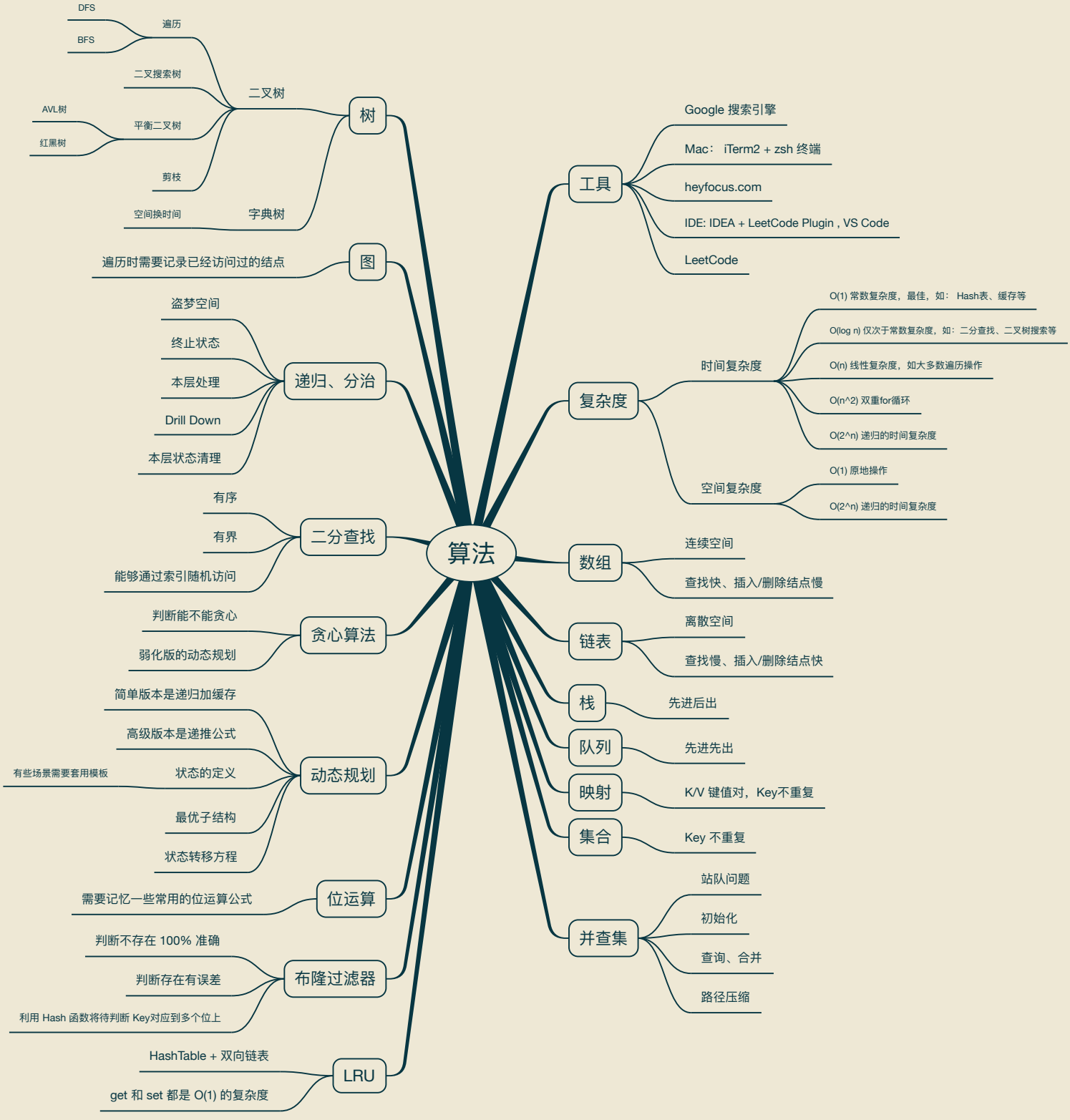
算法
树
二叉树
遍历
DFS
BFS
二叉搜索树
平衡二叉树
AVL树
红黑树
剪枝
字典树
空间换时间
图
遍历时需要记录已经访问过的结点
递归、分治
盗梦空间
终止状态
本层处理
Drill Down
本层状态清理
二分查找
有序
有界
能够通过索引随机访问
贪心算法
判断能不能贪心
弱化版的动态规划
动态规划
简单版本是递归加缓存
高级版本是递推公式
状态的定义
有些场景需要套用模板
最优子结构
状态转移方程
位运算
需要记忆一些常用的位运算公式
布隆过滤器
判断不存在 100% 准确
判断存在有误差
利用 Hash 函数将待判断 Key对应到多个位上
LRU
HashTable + 双向链表
get 和 set 都是 O(1) 的复杂度
工具
Google 搜索引擎
Mac： iTerm2 + zsh 终端
heyfocus.com
IDE: IDEA + LeetCode Plugin , VS Code
LeetCode
复杂度
时间复杂度
O(1) 常数复杂度，最佳，如： Hash表、缓存等
O(log n) 仅次于常数复杂度，如：二分查找、二叉树搜索等
O(n) 线性复杂度，如大多数遍历操作
O(n^2) 双重for循环
O(2^n) 递归的时间复杂度
空间复杂度
O(1) 原地操作
O(2^n) 递归的时间复杂度
数组
连续空间
查找快、插入/删除结点慢
链表
离散空间
查找慢、插入/删除结点快
栈
先进后出
队列
先进先出
映射
K/V 键值对，Key不重复
集合
Key 不重复
并查集
站队问题
初始化
查询、合并
路径压缩