## 数据结构和算法

1. 马马踏棋盘：使用图的深度优化遍历算法(DFS) + 贪心算法优化
2. 八皇后问题：分治算法
3. 字符串匹配：KMP算法《部分匹配表》
4. 修路问题：最小生成树（普里姆算法）
5. 最短路径问题：佛罗伊德算法
6. 数据机构分为线性结构和非线性结构
7. 线性结构常见的有：数组，队列，链表和栈
8. 稀疏数组本质就是保存所有有效数据
9. 栈的特点：先入后出
10. 栈(stack)的介绍
    1. 栈是一个先入后出(filo-Frist In last out)的有序列表
    2. 栈是限制线性表中元素的插入和删除只能在线性表的同一端进行的一种特殊线性表。允许插入和删除的一端，为变化的一端，称为栈顶（top），另一端为固定的一端，称为栈底（bottom）
    3. 根据栈的定义可知，最先放入栈中元素在栈底，最后放入的元素在栈顶，而删除元素刚好相反，最后放入的元素最先删除，最先放入的元素最后删除
11. 递归(recursion)
    1. 简单地说：递归就是方法自己调用自己，每次调用时传入不同的变量，递归有助于编程者解决复杂的问题，同时可以让代码变得简洁
    2. 每次执行到一个方法时都会在栈中开辟一个空间，存储当前方法的变量信息，递归既是
    3. 回溯既是方法执行之前方法未完成的代码
    4. 就迷宫案例来讲，一旦遇到return 的值不满足条件，即1,2,3，就会返回return false进行回溯，即：回溯既是方法执行之前方法未完成的代码
12. 八大排序
    1. 插入排序：直接插入排序，希尔排序
    2. 选择排序：简单选择排序，堆排序
    3. 交换排序：冒泡排序，快速排序
    4. 归并排序
    5. 基数排序
13. 算法的时间复杂度，即代码的执行次数，一个算法中的语句执行的次数越多，花费的时间就会越多，一个算法中的语句的执行次数称为语句频度或时间频度
    1. 常数项可以忽略
    2. 低次项可以忽略
    3. 系数可以忽略
14. 选择排序：选择排序也属于内部排序法，是从欲排序的数据中，按指定规则选出某一些元素，在依规定交换位置后达到排序的目的
15. 插入排序法:插入式排序属于内部排序法，是对于欲排序的元素以插入的方式找寻该元素的适当位置，以达到排序的目的
16. 基数排序不可以排负数，如果要排负数则需要其他的改进操作，及数组中不能出现负数，基数排序排位数高的数效率会略微下降，比如订单号，QQ号等
17. 查找算法：顺序查找，二分查找/折半查找，插值查找，斐波那契查找
18. 树的带权路径长度（wpl）最小，这样的二叉树为最优二叉树，称为赫夫曼树
19. Jdk8加入红黑树应该是为了提升查询删除修改的效率，因为左右两个子树的高度差的绝对值不会超过1
20. 平衡二叉树算法：红黑树，AVL，替罪羊树，Treap，伸展树等
21. 平衡二叉树进行添加时旋转的条件
    1. 左旋转：如果它的，(右子树的高度 - 左子树的高度) > 1
    2. 右旋转：如果它的，(左子树的高度 - 右子树的高度) > 1
    3. 双旋转：
       1. 先对右子节点（子树）右旋转，当前节点左旋转
          1. 如果它的右子树的左子树高度大于它的右子树的右子树高度
          2. 先对右子节点进行右旋转
          3. 然后在对当前节点进行左旋转
       2. 先对左子节点（子树）左旋转，当前节点右旋转
          1. 如果它的左子树的右子树高度大于它的左子树的高度
          2. 先对左子节点进行左旋转
          3. 然后在对当前节点进行右旋转
22. 贪心算法：贪心算法所得到的结果不一定是最优的结果（有时候是最优解），但都是相对近似（接近最优解）的结果