一、算法基本概念

循环不变式

在第一次进入循环之前成立、以后每次循环之后还成
立的关系称为“循环不变关系”或“循环不变式”、“循环不变性质”。

二、函数增长

1. O, o, ω, Ω记号定义的不同 存在与任意的不同
2. θ 符号与渐进紧确界

三、分治法

1.算法

- 非空连续子数组 T(n)=2T(n/2)+$\theta(n)$
- Strassen矩阵 T(n)=7T(n/2)+$a\theta(n^2)$
- 最近点对 T(n)=2T(n/2)+$\theta(n)$

2.求解递归式

1.三种方法

- 主方法 注意非渐进的小于和大于 eg. T(n)=2T(n/2)+nlogn
- 递归树
- 代换法 注意变量代换 T(n) = 2T($\sqrt{n}$) + log n 要令m = log n

四、中位数和顺序统计量

1. RANDOMIZED-PARTITION，借助QUICKSORT的PARTITION过程
   - 最坏时间O(n^2) 期望时间O(n)
2. 二次取中算法 T(n) = T(n/5)+T(3n/4)+cn
   - 注意在有相同元素时，有两种改进办法。一是划分为三个集合，二是选取更大的r值

五、动态规划

思想：最优子结构和无后向性，子问题不交叉（最短路）

所谓“剪切粘贴法”：本质是反证法，把更优的局部解代换，可以得到
更好的全局解。

什么叫子问题图 用于描述子问题与子问题之间的依赖关系

算法

- 钢条切割： 存储每个长度的最优切割方案
- 矩阵链乘法： 存储每个局部矩阵链的最优乘法
- 最长公共子序列： 存储两个序列的各自的前缀序列的最长公共子序列

- 最优二叉搜索树： 存储每个局部词汇的最优二叉搜索树

六、贪心方法

与动态规划的区别
贪心算法解决的问题仍然具有最优子结构，但是每次贪心操作后，它把问题转化成了只有一个子问题需要求解。

所以贪心算法的关键在于贪心选择性质和最优子结构

算法：

- 活动选择问题
- 霍夫曼编码问题

七、搜索

- BFS算法（无须赘述）
- DFS算法（无须赘述）
- D-search算法：就是在BFS中把队列换成栈来实现（名字看起来还挺唬人的）
- 所谓回溯法：使用限界函数的深度优先搜索（DFS），限界函数就是一些剪枝操作的函数
- 所谓分支限界法：使用限界函数的宽度优先搜索（BFS）。所谓FIFO：就是队列！ 所谓LIFO：就是栈！
- 所谓LC（Least Cost）检索：就是A*算法！

关于A*，如果要找到最优解，对于估价函数的限制。（需要证明可以在网上找）

几个要掌握的算法例子

- N-皇后问题
- 子集合数 要记得其中的几个剪枝是怎么做的
- 15-迷问题 要记得判定这个状态是否可以推到最终的状态
- 惩罚作业调度

八、最小生成树

最小生成树在数据结构，离散数学课程中已经多次讲述。重点知道kruskal和prim即可。在这门课程中可以注意一下

- 横跨切割
- 轻量边
- 尊重

这几个概念。

九、最短路径

算法

- Bellman-ford 可以处理一般的最短路问题（包含负权重的边，包含负权重的环都可以处理）可以报告有负权重的环

- Dijkstra 无法处理有负权重的边的图
- 差分约束
  关于差分约束，可以这么想。我们定义的边是（v_i, v_j）对应$x_j - x_i \leq b_k$那么就有$x_j \leq x_i + b_k$，这是满足下面的三角不等式的性质的。

最好了解清楚的概念（防止考概念的时候傻眼）

- 前驱子图 算法终止时，G_π是一棵最短路径树
- INITIALIZE-SINGLE-SOURCE过程，即V.π（前驱） 和v.d（最短路径权重上界）的初始化过程
- 松弛操作 u,v,w 简单的说，就是检查走（u，v)这条边可不可以使到v的距离更小。
- 三角不等式性质 $\delta(s,v) \leq \delta(s,u) + w(u,v)$
- 上界性质 v.d是最短路径上界
- 非路径性质 不存在路径的话，v.d为无穷
- 路径松弛性质 -》用于证明Bellman-ford

十、最大流最小割

Ford-Fulkerson和Edmonds-Karp

Edmonds-Karp基于Ford-Fulkerson 使用深度优先搜索来解决问题

- 切割的净流量的定义
- 切割的容量的定义