1. 第一章：
   1. 算法的概念
      1. 算法四个特性
         1. 有限性
         2. 输入
         3. 输出
         4. 确定性
   2. 渐进复杂度 +　证明
      1. 一个for循环 O（n）
      2. 两重for循环 O(
2. 第二章：
   1. 递归的概念 边界条件+递归方程
      1. 概念
         1. 直接或者间接调用自身的算法称为递归算法；
      2. 特点
         1. 终止条件；
         2. 递归方程；
      3. 预测考点：
         1. 例２－４的排序问题，程序；
   2. 分治法4个适用条件
      1. 问题的规模缩小到一定程度时容易解决；
      2. 该问题可以分解为若干个规模较小的相同问题；（最有子结构）
      3. 分解成的子问题可以合并为该问题的解；
      4. 该问题分解的子问题相互独立，不包含公共子问题；
   3. 分治法基本步骤以及概念（ｐ１６页程序）
      1. 分解为大小相等的子问题；
      2. 递归地解各个子问题；
      3. 将各个子问题的解合并为原问题的解决；
   4. 二分搜索（了解方法）
      1. 特点：
         1. 问题已序；
   5. 棋盘覆盖画图
      1. 考点：画图（ｐ２１图２－４）

|  |  |  |  |
| --- | --- | --- | --- |
| １ | 已经填充的点 | ５ | ５ |
| １ | １ | ２ | ５ |
| ３ | ２ | ２ | ４ |
| ３ | ３ | ４ | ４ |

* 1. 合并排序思想 （给一串数字，让排序）
     1. ｐｐｔ有步骤；
     2. 可能考思路；
  2. 快速排序的概念、编程/画图
     1. 考点：
        1. 代码；
  3. 线性时间选择的思想
     1. 算法思路，　没有程序；
  4. 最接近点对的思想
     1. 算法思路，　没有程序；

1. 第三章：
   1. ０、１背包跳跃点思想可能考
   2. 动态规划的基本要素：
      1. 问题的规模缩小到一定程度时容易解决；
      2. 该问题可以分解为若干个规模较小的相同问题；（最有子结构）
      3. 分解成的子问题可以合并为该问题的解；
      4. 该问题分解的子问题包含公共子问题；
   3. 矩阵连乘 编程
      1. 考程序（ｐ１７）；
      2. 动态规划思想：
         1. 记录两个的矩阵的连称积；
         2. 记录三个矩阵的连称积；
         3. 记录四个矩阵的连称积
         4. 记录五个矩阵的连称积
         5. 记录六个矩阵的连称积
         6. 查表得到最优值；
   4. 最长公共子序列的表
      1. 可能画表；　记住公式；
   5. 最大子段和：（动态规划的方法）只有值无最优解
      1. 如果有时间可以看下（ｐ５７中间的几行程序）
   6. 电路布线 （应用）
      1. 考点，给你个问题，写出那些点可以在一层；
      2. 思路：
         1. 图３－５的解法

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| １ | ２ | ３ | ４ | ５ | ６ | ７ | ８ | ９ | １０ |
| ８ | ７ | ４ | ２ | ５ | １ | ９ | ３ | １０ | ６ |

* + 1. 思路：
       1. 第一层从左向右；
       2. 下面的点不交叉的最大点个数就是最优解；
    2. 上题的答案
       1. ３、５、７、９和４、５、７、９

1. 第四章：
   1. 动态规划和贪心的区别（同：最优子结构。不同：最优子结构性质/贪心选择性质）
   2. 最优装载：物品个数越多越好
   3. 活动安排问题：
      1. 简单证明，不考程序
   4. 哈夫曼编码
      1. 会画图
   5. 单源最短路径 算法
      1. 会算；写出方案
   6. 最小生成树
      1. 会画图
2. 第五章：
   1. 回溯的概念
      1. 以深度优先方式系统搜索问题解得算法；
   2. 算法框架
      1. 剪枝限界
   3. 符号三角形：
      1. 子集树；
      2. ＋或－不超过：
   4. 装载问题
      1. 考思路；
      2. 首先将第一艘船尽可能装满（回溯）
      3. 将剩下的箱子装上第二艘船；
   5. 图的m着色
      1. 程序
   6. 回溯法的效率分析；
3. 第六章：
   1. 0-1背包的优先级，选优先级高的扩充
      1. 优先级定义：
         1. 单位质量的价值高的优先级高
            1. 价值／重量
   2. 考虑优先级就好
   3. 最大团问题考点
      1. 约束函数：
         1. ｃｎ　＋ｎ－ｉ　＜ｂｅｓｔｎ剪掉
            1. ｃｎ：　当前团个数；
            2. ｎ：点的个数
            3. ｉ：当天所在层；
            4. ｂｅｓｔｎ：　已得到的最有团的点个数；
      2. 优先级：
         1. ｕｎ＝ｃｎ＋ｎ－ｌｅｖｅｌ＋１
            1. ｃｎ：当前团类点的个数
            2. ｎ：问题的点的个数
4. 第七章
   1. 有点可能考
      1. 用投点法计算和定积分；