算法设计与分析总复习

第1章 算法引论

- 算法的概念
- 算法复杂性分析

时间复杂性和空间复杂性

最坏情况、最好情况、平均情况的复杂性

渐进复杂性的意义

渐近意义下的记号：O、Ω、θ

掌握算法渐进复杂性的分析

第2章 递归与分治策略

- 算法总体思想

1、将问题分解成k个子问题，对这k个子问题分别求解。如果子问题的规模仍然不够小，则再划分为k个子问题，如此递归的进行下去，直到问题规模足够小，很容易求出其解为止。

2、将求出的小规模的问题的解合并为一个更大规模的问题的解，自底向上逐步求出原来问题的解。

- 递归的概念

第2章 递归与分治策略

- 分治法的适用条件

分治法所能解决的问题一般具有以下几个特征：

- ✓ **该问题的规模缩小到一定的程度就可以容易地解决；**
- ✓ **该问题可以分解为若干个规模较小的相同问题，即该问题具有最优子结构性质**
- ✓ **利用该问题分解出的子问题的解可以合并为该问题的解；**
- ✓ **该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题。**

第2章 递归与分治策略

- 分治法的基本步骤

```
divide-and-conquer(P)
 {
    if ( | P | <= n0) adhoc(P);   //解决小规模的问题
    divide P into smaller subinstances P1,P2,...,Pk;  //分解问题
    for (i=1,i<=k,i++)
       yi=divide-and-conquer(Pi);  //递归的解各子问题
    return merge(y1,...,yk);  //将各子问题的解合并为原问题的解
 }
```

- 分治法的复杂性分析

$$T(n)=\begin{cases} O(1) & n=1 \\ kT(n/m)+f(n) & n>1 \end{cases}$$

$$T(n)=n^{\log_m k}+\sum_{j=0}^{\log_m n-1} k^j f(n/m^j)$$

第2章 递归与分治策略

- 分治法的算法应用

1. 二分法搜索
2. 大整数的乘法
3. Strassen矩阵乘法
4. 棋盘覆盖
5. 合并排序与快速排序
6. 线性时间选择
7. 最接近点对问题
8. 循环赛日程表

第3章　动态规划

- 动态规划基本思想

保存已解决的子问题的答案，而在需要时再找出已求得的答案，就可以避免大量重复计算，从而得到多项式时间算法。

- 动态规划基本步骤

✓ 找出最优解的性质，并刻划其结构特征。

✓ 递归地定义最优值。

✓ 以自底向上的方式计算出最优值。

✓ 根据计算最优值时得到的信息，构造最优解。

- 动态规划算法的基本要素

1、最优子结构性质　2、重叠子问题

第3章　动态规划

- 动态规划算法举例
  - ✓ 完全加括号的矩阵连乘积
  - ✓ 最长公共子序列
  - ✓ 凸多边形最优三角剖分
  - ✓ 多边形游戏
  - ✓ 图像压缩
  - ✓ 流水作业调度
  - ✓ 0-1背包问题

第4章　贪心算法

- 贪心算法的基本要素

✓ **贪心选择性质**

贪心选择性质是指所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到。

✓ **最优子结构性质**

当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。

- 贪心算法与动态规划算法的差异

第4章　贪心算法

- 贪心算法应用举例
  - ✓ 活动安排问题
  - ✓ 最优装载
  - ✓ 哈夫曼编码
  - ✓ 单源最短路径
  - ✓ 最小生成树
  - ✓ 多机调度问题

第5章　回溯法

- 回溯法概念

具有限界函数的深度优先生成法称为回溯法。
回溯法的基本做法是搜索，在问题的解空间树中，按深度优先策略，从根结点出发搜索解空间树。这种方法适用于解一些组合数相当大的问题。

- 问题的解空间

1、问题的解向量、显约束、隐约束、解空间的概念
2、结点的状态：扩展结点、活结点、死结点

第5章　回溯法

- 回溯法的基本思想

(1)针对所给问题，定义问题的解空间；
(2)确定易于搜索的解空间结构；
(3)以深度优先方式搜索解空间，并在搜索过程中用剪枝函数避免无效搜索。

- 回溯算法的结构

递归回溯与迭代回溯

两种典型的回溯问题：子集树与排列树

第5章　回溯法

- 回溯法的应用算法
  - ✓ 装载问题
  - ✓ 批处理作业调度
  - ✓ 符号三角形问题
  - ✓ n后问题
  - ✓ 0-1背包问题
  - ✓ 最大团问题
  - ✓ 图的m着色问题
  - ✓ 旅行售货员问题
  - ✓ 圆排列问题
  - ✓ 连续邮资问题

第6章　分支限界法

分支限界法基本思想

分支限界法是另一种采用搜索技术求解问题的方法。

在分支限界法中，每一个活结点只有一次机会成为扩展结点。活结点一旦成为扩展结点，就一次性产生其所有儿子结点。在这些儿子结点中，导致不可行解或导致非最优解的儿子结点被舍弃，其余儿子结点被加入活结点表中。 此后，从活结点表中取下一结点成为当前扩展结点，并重复上述结点扩展过程。这个过程一直持续到找到所需的解或活结点表为空时为止。

第6章　分支限界法

- 分支限界法与回溯法的不同

（1）求解目标：回溯法的求解目标是找出解空间树中满足约束条件的所有解，而分支限界法的求解目标则是找出满足约束条件的一个解，或是在满足约束条件的解中找出在某种意义下的最优解。

（2）搜索方式的不同：回溯法以深度优先的方式搜索解空间树，而分支限界法也可以采用广度优先或以最小耗费优先的方式搜索解空间树。

（3）结点的生成和状态变化不同:回溯法中每次只扩展生成一个子结点，而分支限界法中每个活结点只有一次机会成为扩展结点，每次扩展生成所有子结点。

第6章　分支限界法

- 分支限界法应用
  - ✓ 单源最短路径问题
  - ✓ 装载问题
  - ✓ 布线问题
  - ✓ 0－1背包问题
  - ✓ 最大团问题
  - ✓ 旅行售货员问题
  - ✓ 批处理作业调度

第8章　NP完全性理论

- 基本概念

  确定性算法、非确定性算法

  P类与NP类问题

  P、NP、NP完全和NP难问题及其相互之间的关系

- 问题变换与计算复杂性归约

  问题变换的概念

  问题的变换与问题的计算复杂性归约的关系

- NP完全问题

  多项式时间变换

  判定一个问题是NP完全问题的基本方法。

第9章 近似算法

近似算法的性能比定义为$\eta=\max\left\{\frac{c}{c^*},\frac{c^*}{c}\right\}$。在通常情况下，该性能比是问题输入规模n的一个函数ρ(n)，即$\max\left\{\frac{c}{c^*},\frac{c^*}{c}\right\}\le\rho(n)$。

近似算法的相对误差定义为$\lambda=\left|\frac{c-c^*}{c^*}\right|$若对问题的输入规模n，有一函数ε(n)使得$\left|\frac{c-c^*}{c^*}\right|\le\varepsilon(n)$，则称ε(n)为该**近似算法的相对误差界**。

一些典型问题的近似算法设计与性能分析