

计算机算法设计与分析复习题

一、填空题

- 1、一个算法复杂性的^{高低}体现在计算机运行该算法所需的时间和存储器资源上，因此算法的复杂性有 时间 复杂性和空间复杂性之分。
- 2、出自于“平衡子问题”的思想，通常分治法在分割原问题，形成若干子问题时，这些子问题的规模都大致 相同。
- 3、使用二分搜索算法在 n 个有序元素表中搜索一个特定元素，在最佳情况下，搜索的时间复杂性为 $O(1)$ ，在最坏情况下，搜索的时间复杂性为 $O(\log n)$ 。
- 4、已知一个分治算法耗费的计算时间 $T(n)$ ， $T(n)$ 满足如下递归方程：

$$T(n) = \begin{cases} O(1) & n < 2 \\ 2T(n/2) + O(n) & n \geq 2 \end{cases}$$

解得此递归方程可得 $T(n) = O(n \log n)$ 。

- 5、动态规划算法有一个变形方法 备忘录方法。这种方法不同于动态规划算法“自底向上”的填充方向，而是“自顶向下”的递归方向，为每个解过的子问题建立了备忘录以备需要时查看，同样也可避免相同子问题的重复求解。
6. 递归的二分查找算法在 divide 阶段所花的时间是 $O(1)$ ，conquer 阶段所花的时间是 $T(n/2)$ ，算法的时间复杂度是 $O(\log n)$ 。
7. Prim 算法利用 贪心 策略求解 最小生成树 问题，其时间复杂度是 $O(n^2)$ 。
8. 背包问题可用 贪心法，回溯法 等策略求解。
9. 用动态规划算法计算矩阵连乘问题的最优值所花的时间是 $O(n^3)$ ，子问题空间大小是 $O(n^2)$ 。
10. 图的 m 着色问题可用 回溯 法求解，其解空间树中叶子结点个数是 m^n ，解空间树中每个内结点的孩子数是 m 。
11. 单源最短路径问题可用 贪心法、分支限界 等策略求解。
- 12、一个算法的优劣可以用（时间复杂度）与（空间复杂度）来衡量。
- 13、回溯法在问题的解空间中，按（深度优先方式）从根结点出发搜索解空间树。
- 14、直接或间接地调用自身的算法称为（递归算法）。
- 15、 θ 记号在算法复杂性的表示法中表示（渐进确界或紧致界）。
- 16、在分治法中，使子问题规模大致相等的做法是出自一种（平衡(balancing)子问题）的思想。
- 17、动态规划算法适用于解（具有某种最优性质）问题。
- 18、贪心算法做出的选择只是（在某种意义上的局部）最优选择。

- 19、最优子结构性质的含义是（**问题的最优解包含其子问题的最优解**）。
- 20、回溯法按（**深度优先**）策略从根结点出发搜索解空间树。
- 21、拉斯维加斯算法找到的解一定是（**正确解**）。
- 22、按照符号 O 的定义 $O(f)+O(g)$ 等于 $O(\max\{f(n), g(n)\})$ 。
- 23、二分搜索技术是运用（**分治**）策略的典型例子。
- 24、动态规划算法中，通常不同子问题的个数随问题规模呈（**多项式**）级增长。
- 25、（**最优子结构性质**）和（**子问题重叠性质**）是采用动态规划算法的两个基本要素。
- 26、（**最优子结构性质**）和（**贪心选择性质**）是贪心算法的基本要素。
- 27、（**选择能产生最优解的贪心准则**）是设计贪心算法的核心问题。
- 28、分支限界法常以（**广度优先**）或（**以最小耗费(最大效益)优先**）的方式搜索问题的解空间树。
- 29、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，即贪心选择达到。
- 30、按照活结点表的组织方式的不同，分支限界法包括（**队列式(FIFO)分支限界法**）和（**优先队列式分支限界法**）两种形式。
- 31、如果对于同一实例，蒙特卡罗算法不会给出两个不同的正确解答，则称该蒙特卡罗算法是（**一致的**）。
- 32、哈夫曼编码可利用（**贪心法**）算法实现。
- 33 概率算法有**数值概率算法，蒙特卡罗 (Monte Carlo) 算法，拉斯维加斯 (Las Vegas) 算法和舍伍德 (Sherwood) 算法**
- 34 以自顶向下的方式求解最优解的有（**贪心算法**）
- 35、下列算法中通常以自顶向下的方式求解最优解的是（**贪心法**）。
- 36、在对问题的解空间树进行搜索的方法中，一个活结点有多次机会成为活结点的是（**回溯法**）
- 37、旅行售货员问题不能用（）解决 **可以用回溯法解决，分支限界法，NP 完全性理论与近似算法**
- 38、贪心算法不能解决（**0-1 背包问题 N 皇后问题**）。可以解决背包问题
- 39、投点法是（**概率算法**）的一种。
- 40、若线性规划问题存在最优解，它一定不在（**可行域内部**）

二、简答题

- 1、(8分) 写出下列复杂性函数的偏序关系（即按照渐进阶从低到高排序）：

$$2^n \quad 3^n \quad \log n \quad n! \quad n \log n \quad n^2 \quad n^n \quad 10^3$$

参考解答: $10^3 < \log n < n \log n < n^2 < 2^n < 3^n < n! < n^n$

2、(8分) 现在有 8 位运动员要进行网球循环赛, 要设计一个满足以下要求的比赛日程表:

- (1) 每个选手必须与其他选手各赛一次;
- (2) 每个选手一天只能赛一次;
- (3) 循环赛一共进行 $n - 1$ 天。

请利用分治法的思想, 给这 8 位运动员设计一个合理的比赛日程。

参考解答:

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

3、(8分) 某体育馆有一羽毛球场出租, 现在总共有 10 位客户申请租用此羽毛球场, 每个客户所租用的时间单元如下表所示, $s(i)$ 表示开始租用时刻, $f(i)$ 表示结束租用时刻, 10 个客户的申请如下表所示:

i	1	2	3	4	5	6	7	8	9	10
s(i)	0	3	1	5	3	5	11	8	8	6
f(i)	6	5	4	9	8	7	13	12	11	10

同一时刻, 该羽毛球场只能租借给一位客户, 请设计一个租用安排方案, 在这 10 位客户里面, 使得体育馆能尽可能满足多位客户的需求, 并算出针对上表的 10 个客户申请, 最多可以安排几位客户申请。

参考解答: 将这 10 位客户的申请按照结束时间 $f(i)$ 递增排序, 如下表:

i	1	2	3	4	5	6	7	8	9	10
s(i)	1	3	0	5	3	5	6	8	8	11
f(i)	4	5	6	7	8	9	10	11	12	13

(1) 选择申请 1 (1, 4)

(2) 依次检查后续客户申请, 只要与已选择的申请相容不冲突, 则选择该申请。直到所有申请检查完毕。申请 4 (5, 7)、申请 8 (8, 11)、申请 10 (11, 13)

(3) 最后, 可以满足: 申请 1 (1, 4)、申请 4 (5, 7)、申请 8 (8, 11)、申请 10 (11, 13) 共 4 个客户申请。这已经是可以满足的最大客户人数。

4、(8分) 对于矩阵连乘所需最少数乘次数问题，其递归关系式为：

$$m[i, j] = \begin{cases} 0 & i = j \\ \min_{i \leq k < j} \{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\} & i < j \end{cases}$$

其中 $m[i, j]$ 为计算矩阵连乘 $A_i \cdots A_j$ 所需的最少数乘次数， p_{i-1} 为矩阵 A_i 的行， p_i 为矩阵 A_i 的列。现有四个矩阵，其中各矩阵维数分别为：

A_1	A_2	A_3	A_4
50×10	10×40	40×30	30×5
$p_0 \times p_1$	$p_1 \times p_2$	$p_2 \times p_3$	$p_3 \times p_4$

请根据以上的递归关系，计算出矩阵连乘积 $A_1A_2A_3A_4$ 所需要的最少数乘次数。

参考解答：

$$m[1][4] = \min \begin{cases} m[1][1] + m[2][4] + p_0p_1p_4 = 0 + 8000 + 50 \times 10 \times 5 = 10500 \\ m[1][2] + m[3][4] + p_0p_2p_4 = 20000 + 6000 + 50 \times 40 \times 5 = 36000 \\ m[1][3] + m[4][4] + p_0p_3p_4 = 27000 + 0 + 50 \times 30 \times 5 = 34500 \end{cases}$$

= 10500

5、(8分) 有这样一类特殊 0-1 背包问题：可选物品重量越轻的物品价值越高。

$n=6$, $c=20$, $P=(4, 8, 15, 1, 6, 3)$, $W=(5, 3, 2, 10, 4, 8)$ 。

其中 n 为物品个数， c 为背包载重量， P 表示物品的价值， W 表示物品的重量。请问对于此 0-1 背包问题，应如何选择放进去的物品，才能使到放进背包的物品总价值最大，能获得的最大总价值多少？

参考解答：因为该 0-1 背包问题比较特殊，恰好重量越轻的物品价值越高，所以优先取重量轻的物品放进背包。最终可以把重量分别为 2, 3, 4, 5 的三个物品放进背包，得到的价值和为 $15 + 8 + 6 + 4 = 33$ ，为最大值。

6. 请用英文写出三种以上能求解 0-1 背包问题的设计算法策略。

参考解答：

Dynamic Programming

Backtrack

Branch-and-Bound

(每答对一条给一分)

7. 请说明动态规划方法为什么需要最优子结构性质。

参考解答：最优子结构性质是指大问题的最优解包含子问题的最优解。

动态规划方法是自底向上计算各个子问题的最优解，即先计算子问题的最优解，然后再利用子问题的最优解构造大问题的最优解，因此需要最优子结构

8. 请说明：(1) 优先队列可用什么数据结构实现？(2) 优先队列插入算法基本思

想？(3) 优先队列插入算法时间复杂度？

参考解答：(1) 堆。(1分)

(2) 在小根堆中，将元素 x 插入到堆的末尾，然后将元素 x 的关键字与其双亲的关键字比较，若元素 x 的关键字小于其双亲的关键字，则将元素 x 与其双亲交换，然后再将元素 x 与其新双亲的关键字相比，直到元素 x 的关键字大于双亲的关键字，或元素 x 到根为止。(4分)

(3) $O(\log n)$ (1分)

9. 设计动态规划算法的主要步骤是怎么的？请简述。

参考解答：(1) 找出最优解的性质，并刻画其结构特征。(6分)

(2) 递归地定义最优值。

(3) 以自底向上的方式计算出最优值。

(4) 根据计算最优值时得到的信息，构造最优解。

10. 分治法所能解决的问题一般具有哪几个特征？请简述。

参考解答：(1) 该问题的规模缩小到一定的程度就可以容易地解决；(6分)

(2) 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；

(3) 利用该问题分解出的子问题的解可以合并为该问题的解；

(4) 原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。

11. 分支限界法的搜索策略是什么？

参考解答：在扩展结点处，先生成其所有的儿子结点（分支），然后再从当前的活结点表中选择下一个扩展结点。为了有效地选择下一扩展结点，加速搜索的进程，在每一个活结点处，计算一个函数值（限界），并根据函数值，从当前活结点表中选择一个最有利的结点作为扩展结点，使搜索朝着解空间上有最优解的分支推进，以便尽快地找出一个最优解。(6分)

12. 算法的要特性是什么？

参考解答：确定性、可实现性、输入、输出、有穷性

13. 算法分析的目的是什么？

参考解答：分析算法占用计算机资源的情况，对算法做出比较和评价，设计出额更好的算法。

14. 算法的时间复杂性与问题的什么因素相关？

参考解答：算法的时间复杂性与问题的规模相关，是问题大小 n 的函数。

15. 算法的渐进时间复杂性的含义？

参考解答：当问题的规模 n 趋向无穷大时，影响算法效率的重要因素是 $T(n)$ 的数量级，而其他因素仅是使时间复杂度相差常数倍，因此可以用 $T(n)$ 的数量级

(阶)评价算法。时间复杂度 $T(n)$ 的数量级(阶)称为渐进时间复杂性。

16 最坏情况下的时间复杂性和平均时间复杂性有什么不同?

参考解答: 最坏情况下的时间复杂性和平均时间复杂性考察的是 n 固定时, 不同输入实例下的算法所耗时间。最坏情况下的时间复杂性取的输入实例中最大的时间复杂度:

$$W(n) = \max \{ T(n, I) \}, \quad I \in D_n$$

平均时间复杂性是所有输入实例的处理时间与各自概率的乘积和:

$$A(n) = \sum P(I) T(n, I) \quad I \in D_n$$

17 简述二分检索(折半查找)算法的基本过程。

参考解答: 设输入是一个按非降次序排列的元素表 $A[i:j]$ 和 x , 选取 $A[(i+j)/2]$ 与 x 比较, 如果 $A[(i+j)/2]=x$, 则返回 $(i+j)/2$, 如果 $A[(i+j)/2]<x$, 则 $A[i:(i+j)/2-1]$ 找 x , 否则在 $A[(i+j)/2+1:j]$ 找 x 。上述过程被反复递归调用。

18 背包问题的目标函数和贪心算法最优化量度相同吗?

参考解答: 不相同。目标函数: 获得最大利润。最优量度: 最大利润/重量比。

19 采用回溯法求解的问题, 其解如何表示? 有什么规定?

参考解答: 问题的解可以表示为 n 元组: (x_1, x_2, \dots, x_n) , $x_i \in S_i$, S_i 为有穷集合, $x_i \in S_i$, (x_1, x_2, \dots, x_n) 具备完备性, 即 (x_1, x_2, \dots, x_n) 是合理的, 则 (x_1, x_2, \dots, x_i) ($i < n$) 一定合理。

20 回溯法的搜索特点是什么?

参考解答: 在解空间树上跳跃式地深度优先搜索, 即用判定函数考察 $x[k]$ 的取值, 如果 $x[k]$ 是合理的就搜索 $x[k]$ 为根节点的子树, 如果 $x[k]$ 取完了所有的值, 便回溯到 $x[k-1]$ 。

21 n 皇后问题回溯算法的判别函数 place 的基本流程是什么?

参考解答: 将第 k 行的皇后分别与前 $k-1$ 行的皇后比较, 看是否与它们相容, 如果不相容就返回 false, 测试完毕则返回 true。

22 为什么用分治法设计的算法一般有递归调用?

参考解答: 子问题的规模还很大时, 必须继续使用分治法, 反复分治, 必然要用到递归。

23 为什么要分析最坏情况下的算法时间复杂性?

参考解答: 最坏情况下的时间复杂性决定算法的优劣, 并且最坏情况下的时间复杂性较平均时间复杂性更可操作。

24 简述渐进时间复杂性上界的定义。

参考解答: $T(n)$ 是某算法的时间复杂性函数, $f(n)$ 是一简单函数, 存在正整数 N_0 和 C , $n > N_0$, 有 $T(n) < C f(n)$, 这种关系记作 $T(n) = O(f(n))$ 。

25 二分检索算法最多的比较次数?

参考解答: 二分检索算法的最多的比较次数为 $\log n$ 。

26 快速排序算法最坏情况下需要多少次比较运算?

参考解答: 最坏情况下快速排序退化成冒泡排序, 需要比较 n^2 次。

27 贪心算法的基本思想？

参考解答：是一种依据最优化量度依次选择输入的分级处理方法。基本思路是：首先根据题意，选取一种**量度标准**；然后按这种量度标准对这 n 个输入排序，依次选择输入量加入部分解中。如果当前这个输入量的加入，不满足约束条件，则不把此输入加入到这部分解中

28 回溯法的解 (x_1, x_2, \dots, x_n) 的隐约束一般指什么？

参考解答：回溯法的解 (x_1, x_2, \dots, x_n) 的隐约束一般指个元素之间应满足的某种关系。

29 阐述归并排序的分治思路。

参考解答：讲数组一分为二，分别对每个集合单独排序，然后将已排序的两个序列归并成一个含 n 个元素的分好类的序列。如果分割后子问题还很大，则继续分治，直到一个元素。

30 快速排序的基本思想是什么。

参考解答：快速排序的基本思想是在待排序的 N 个记录中任意取一个记录，把该记录放在最终位置后，数据序列被此记录分成两部分。所有关键字比该记录关键字小的放在前一部分，所有比它大的放置在后一部分，并把该记录排在这两部分的中间，这个过程称作一次快速排序。之后重复上述过程，直到每一部分内只有一个记录为止。

31 什么是直接递归和间接递归？消除递归一般要用到什么数据结构？

参考解答：在定义一个过程或者函数的时候又出现了调用本过程或者函数的成分，既调用它自己本身，这称为直接递归。如果过程或者函数 P 调用过程或者函数 Q ， Q 又调用 P ，这个称为间接递归。消除递归一般要用到栈这种数据结构。

32 什么是哈密顿环问题？

参考解答：哈密顿环是指一条沿着图 G 的 N 条边环行的路径，它的访问每个节点一次并且返回它的开始位置。

33 用回溯法求解哈密顿环，如何定义判定函数？

参考解答：当前选择的节点 $X[k]$ 是从未到过的节点，即 $X[k] \neq X[i] (i=1, 2, \dots, k-1)$ ，且 $C(X[k-1], X[k]) \neq \infty$ ，如果 $k=-1$ ，则 $C(X[k], X[1]) \neq \infty$ 。

34 请写出 prim 算法的基本思想。

参考解答：思路是：最初生成树 T 为空，依次向内加入与树有最小邻接边的 $n-1$ 条边。处理过程：首先加入最小代价的一条边到 T ，根据各节点到 T 的邻接边排序，选择最小边加入，新边加入后，修改由于新边所改变的邻接边排序，再选择下一条边加入，直至加入 $n-1$ 条边。

三、算法设计题

1. 【最长上升子序列问题】(8分) —— 提示：此题可采用动态规划算法实现

对于给定的一个序列 (a_1, a_2, \dots, a_N) , $1 \leq N \leq 1000$ 。我们可以得到一些递增

上升的子序列 $(a_{i_1}, a_{i_2}, \dots, a_{i_K})$, 这里 $1 \leq i_1 < i_2 < \dots < i_K \leq N$ 。比如, 对于序列 (1, 7, 3, 5, 9, 4, 8), 有它的一些上升子序列, 如 (1, 7), (3, 4, 8) 等等。这些子序列中最长的长度是 4, 比如子序列 (1, 3, 5, 8)。你的任务: 就是对于给定的序列, 求出最长上升子序列的长度。要求写出你设计的算法思想及递推函数的公式表达。.

参考解答: 设 $f(i)$ 表示: 从左向右扫描过来直到以 $a[i]$ 元素结尾的序列, 获得的最长上升子序列的长度, 且子序列包含 $a[i]$ 元素 ($1 \leq i \leq n$)。

$$f(i) = \begin{cases} 1 & i=1 \\ \max\{f(j)+1: \text{当 } a[i] > a[j]; 1 \leq j < i\} & i>1 \\ 1 & i>1; \forall j (1 \leq j < i), \text{ 都有 } a[i] \leq a[j] \end{cases}$$

即, $f(i)$ 是从 $f(1)$, $f(2)$ …… 到 $f(i-1)$ 中找最大的一个值, 再加 1。或者就是 1。主要是看 $a[i]$ 这个元素能否加入到之前已经获得的最长上升子序列, 如果能加入, 是之前已获得的最长上升子序列长度加一; 如果不能加入, 就取这最后一个元素作为一个单独子序列, 长度为 1。

最后, 所要求的整个序列的最长公共子序列长度为 $\max\{f(i): 1 \leq i \leq n\}$

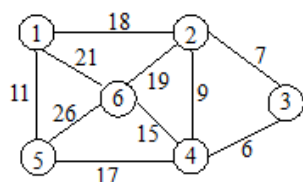
例如, 对于序列: 4 2 6 3 1 5 2

i	1	2	3	4	5	6	7
array	4	2	6	3	1	5	2
f(i)	1	1	2	2	1	<u>3</u>	2

评分准则:

- 1) 答到使用动态规划算法, 并且推导出动态规划算法的递推函数公式表达, 边界设定清晰, 本题即可得满分; (阅卷时仔细看递推公式表达, 公式表达含义正确即可, 因其表达形式可能不唯一)
- 2) 说明使用动态规划算法, 但对递推函数表达错误或含糊, 扣 2 分以上;
- 3) 其它情况酌情考虑。

2. (10分) 对下图所示的连通网络 G , 用克鲁斯卡尔(Kruskal)算法求 G 的最小生成树 T , 请写出在算法执行过程中, 依次加入 T 的边集 TE 中的边。说明该算法的贪心策略和算法的基本思想, 并简要分析算法的时间复杂度。



参考解答

$TE = \{(3, 4), (2, 3), (1, 5), (4, 6), (4, 5)\}$ (5分)

贪心策略是每次都在连接两个不同连通分量的边中选权值最小的边。

基本思想：首先将图中所有顶点都放到生成树中，然后每次都在连接两个不同连通分量的边中选权值最小的边，将其放入生成树中，直到生成树中有 $n-1$ 条边。(4分)

时间复杂度为： $O(e \log e)$ (1分)

3. (15分) 考虑 $n=3$ 的批处理作业调度实例：

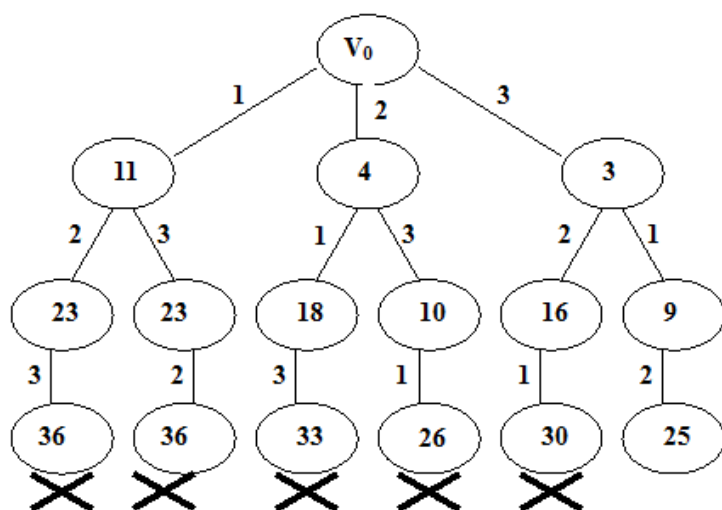
t_{ji}	机器 1	机器 2
作业 1	1	10
作业 2	3	1
作业 3	2	1

其中 t_{ji} 是作业 J_i 需要在机器 j 上处理的时间。对于给定的 3 个作业，制定一个最佳作业调度方案，使其完成时间和达到最小。

要求：

- (1) 画出该问题的解空间树； (5分)
- (2) 写出该问题的剪枝策略(即限界条件), 要求只保留第一个最优解; (2分)
- (3) 按优先队列式分支限界法搜索解空间树, 并用剪枝策略对解空间树中该剪枝的位置打×; (5分)
- (4) 给出最优解及最优值。 (3分)

参考解答 (1) 5分



(2) 若当前代价 $f \geq$ 当前最优解代价 $bestf$, 则剪枝。2 分

(3) 见 (1) 中所画的图。5 分

(4) 最优解为 {3, 1, 2}, 最优值为 25。3 分

4. 【Gray 码构造问题】(8 分) —— 提示: 此题可采用分治递归算法实现

问题描述: “格雷码”是一个长度为 2^n 的序列, 满足:

- (a) 每个元素都是长度为 n 比特的串
- (b) 序列中无相同元素
- (c) 连续的两个元素恰好只有 1 个比特不同

例如: $n=2$ 时, 格雷码为 {00, 01, 11, 10}。

Gray 码是一种编码, 这种编码可以避免在读取时, 因各数据位时序上的差异造成的误读。格雷码在工程上有广泛应用。但格雷码不便于运算, 请你设计一种构造方法, 输入长度序列 n , 输出格雷码 (你只要做出一种构造方案即可, 格雷码并不唯一)。

参考解答: 此题可用分治法解决。

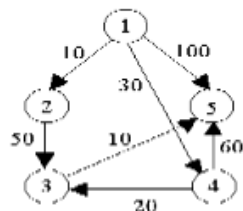
当 $n=1$ 时, 输出格雷码 {0, 1}

当 $n>1$ 时, 格雷码的长度为 2^n , 即共有 2^n 个码序列。此时, 将问题一分为二, 即上半部分和下半部分。上半部分最高位设为 0, 下半部分最高位设为 1。剩下 $n-1$ 位的格雷码的构造采用递归的思路。

评分准则:

- 1) 答到使用分治算法, 并且推导出分治算法的过程, 边界设定清晰 (即当仅输出 1 位的格雷码如何处理), 本题即可得满分;
- 2) 说明使用分治算法, 但漏边界条件, 扣 2 分以上;

其它情况酌情考虑。



5. (13 分) 给定带权有向图 (如下图所示) $G=(V,E)$, 其中每条边的权是非负实数。另外, 还给定 V 中的一个顶点, 称为源。现在要计算从源到所有其它各顶点的最短路长度。这里路的长度是指路上各边权之和。现采用 Dijkstra 算法计算从源顶点 1 到其它顶点间最短路径。请将此过程填入下表中。

迭代	S	u	dist[2]	dist[3]	dist[4]	dist[5]
初始	{1}					
1						
2						
3						
4						

参考解答: (13 分)

迭代	S	u	dist[2]	dist[3]	dist[4]	dist[5]
初始	{1}	-	10		30	100
1	{1, 2}	2	10	60	30	100
2	{1, 2, 4}	4	10	50	30	90
3	{1, 2, 4, 3}	3	10	50	30	60
4	{1, 2, 3, 4, 5}	5	10	50	30	60

6. (13 分) 有 0-1 背包问题如下:

$n=6, c=20, P=(4, 8, 15, 1, 6, 3), W=(5, 3, 2, 10, 4, 8)$ 。

其中 n 为物品个数, c 为背包载重量, P 表示物品的价值, W 表示物品的重量。请问对于此 0-1 背包问题, 应如何选择放进去的物品, 才能使到放进背包的物品总价值最大。

$P=(15, 8, 6, 4, 3, 1) \quad W=(2, 3, 4, 5, 8, 10)$, 单位重量物品价值
(7.5, 2.67, 1.5, 0.8, 0.375, 0.1)

参考解答: (13 分)

可知随着物品的重量增加, 物品的价值减少; 因此可以用贪心算法来求解。以选取单位重量物品价值高为贪心策略。

1. 先把重量为 2 的物品放进背包, 此时剩余载重量为 17, P 为 15。
2. 把重量为 3 的物品放进背包, 此时剩余载重量为 14, P 为 23;
3. 把重量为 4 的物品放进背包, 此时剩余载重量为 10, P 为 29;

4. 把重量为 5 的物品放进背包, 此时剩余载重量为 5, P 为 33.

由于 $8 > 5$, 所以不能再放进背包。

结果是把重量为 2, 3, 4, 5 的物品装进背包, 总价值最大为 33.

7. 将所给定序列 $a[1:n]$ 分为长度相等的两段 $a[1:n/2]$ 和 $a[n/2+1:n]$, 分别求出这两段的最大子段和, 则 $a[1:n]$ 的最大子段和有哪三种情形? (10 分)

参考解答: (1) $a[1:n]$ 的最大子段和与 $a[1:n/2]$ 的最大子段和相同。

(2) $a[1:n]$ 的最大子段和与 $a[n/2+1:n]$ 的最大子段和相同。

(3) $a[1:n]$ 的最大子段和为 $\sum a_k (i \leq k \leq j)$ 且 $1 \leq i \leq n/2, n/2+1 \leq j \leq n$ 。

8 写出 maxmin 算法对下列实例中找最大数和最小数的过程。

数组 $A = (48, 12, 61, 3, 5, 19, 32, 7)$

参考解答: 写出 maxmin 算法对下列实例中找最大数和最小数的过程。

数组 $A = ()$

1、 48, 12, 61, 3, 5, 19, 32, 7

2、 48, 12 61, 3 5, 19 32, 7

3、 48~61, 12~3 19~32, 5~7

4、 61~32 3~5

5、 61 3

9 速排序算法对下列实例排序, 算法执行过程中, 写出数组 A 第一次被分割的过程。

$A = (65, 70, 75, 80, 85, 55, 50, 2)$

参考解答: 第一个分割元素为 65

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	i	p
65	70	75	80	85	55	50	2	2	8
65	2	75	80	85	55	50	70	3	7
65	2	50	80	85	55	75	70	4	6
65	2	50	55	85	80	75	70	4	6
55	70	75	80	85	65	50	2		

10 归并排序算法对下列实例排序, 写出算法执行过程。

$A = (48, 12, 61, 3, 5, 19, 32, 7)$

参考解答:

48, 12, 61, 3	5, 19, 32, 7
48, 12 61, 3	5, 19 32, 7
12, 48 3, 61	5, 19 7, 32
3, 12, 48, 61	5, 7, 19, 32
3, 5, 7, 12, 19, 32, 48, 61	