

附录 A: 本学期的编程任务

A.1: 点共线问题

任给平面上的若干个点, 判断位于同一直线上的点的最大个数。

给出该问题的形式化描述, 确定输入数据格式, 并编写一个程序解决这个问题。

A.2: 多项式规约问题

写一个程序, 实现从 3-CNF-SAT 到 CLIQUE 问题的多项式规约。

A.3: 蠕虫问题

6	8	18	15	24	20	2	20
6	2	15	2	17	15	3	7
0	11	18	16	20	15	1	11
6	2	6	13	4	17	20	16
5	12	7	2	3	5	18	23
7	13	3	2	2	11	4	23
16	23	10	2	4	12	5	20
17	12	10	1	13	12	6	20

假设蠕虫生活的世界是如左图所示的正整数组成的方阵。蠕虫可能降生在任一方格中。从降生点开始蠕虫开始移动, 移动只能选择上下左右四个方向中的一个, 并且不可以越出边界。由于不可知的遗传缺陷, 当它遇到一个前面已经经过的数字时就将死亡。(例如, 在左图中就是一个蠕虫的“生命轨迹”。

编一个程序, 针对一个输入, 找出其最长的“生命轨迹”。为了简单起见, 假设方阵一定是 8×8 的, 而数字一定在集合 $\{1, 2, \dots, 25\}$ 中。

A.4: 颜色串接问题

我们面前是 k 个接力棒, 每个棒用两种颜色着色, 以中间为分界。总共使用 n 种不同的颜色。如果我们在每个棒的两端装一个可以闭合的弯勾, 就可以将这些棒串接成一个环。我们希望直接相邻的棒正好以相同的颜色相接。

编一个程序判定所给的输入是否有解, 如果有, 则给出一种解法。

A.5: 校园活动频度分析问题

假设校园中有 n 个学生活动团体, h_i 表示第 i 个团体举办活动的平均周期 (即第 i 个团体平均每 h_i 天举办一次活动)。编一个程序模拟一年中的活动安排情况, 给出没有任何活动的天数的期望值和分布, 给出没有任何活动的双休日天数的期望值。并试图分析不同参数值对于结果的影响。

A.6: 共同祖先结点问题

编写一个程序完成以下的任务: 输入一个根树, 指定该树中两个结点, 找出该两个结点的共同祖先中离树根最远的一个。(注意: 任意一个结点也可以看作是其自己的祖先。)

A.7: 大小关系序列问题

对于 $(1, 2, \dots, n)$ 的任意一个排列: $A = (a_1, a_2, \dots, a_n)$, 定义一个新的序列 $P(A) = (p_1, p_2, \dots, p_{n-1})$,

其中: p_i 的值按如下方式确定: 若 $a_i > a_{i+1}$, 则 $p_i = 0$, 若 $a_i < a_{i+1}$, 则 $p_i = 1$ 。编写一个程序, 对于上述任意的; 排列 A , 确定有 $(1, 2, \dots, n)$ 的多少个不同排列, 与 A 具有相同的 P 序列。

A.8: 带权图的最小点覆盖问题

实现一个解带权图的最小点覆盖问题的近似算法。并且运行几个实例, 验证理论推导的近似度。

A.9: 背包问题

实现一个解背包问题的近似算法。并且运行几个实例, 验证理论推导的近似度。

A.10: 旅行推销商问题

实现一个解旅行推销商问题的近似算法。并运行几个实例, 验证理论推导的近似度。

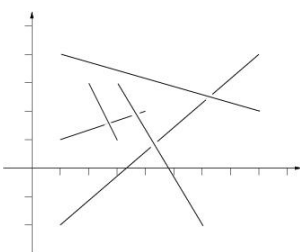
A.11: Bin-Packing 问题

实现一个解 Bin-Packing 问题的近似算法。并且运行几个实例, 验证理论推导的近似度。

A.12: 汽车轮渡模拟问题

编写一个程序模拟一个汽车轮渡的汽车等待情况。假设只有一条轮渡, 每次最多搭载 n 辆车, 往返于河的南北岸之间, 单程渡河时间是 t 分钟。模拟时间段长度为 720 分钟。假设只要有车等候, 轮渡到岸卸载后立刻装载起航。装卸时间忽略不计。如果两岸均无等待车辆, 轮渡就停在当前所在岸边, 一旦有车到达 (不管是哪一边) 立即起航; 如果等待的车数量超过 n , 等待时间长的先商船。随机生成如下形式的输入: 有序对的序列, 每个有序对中第一个值是一辆车到达岸边的时间 (相对于模拟起始时间的位移, 假设最早到达时间是 0, 最晚到达时间是 $720 - 2t$); 第 2 个值指明到达的是南岸还是北岸 (假设模拟开始时轮渡在北岸)。输出每辆量车到达对岸的时间以及在岸边等待的时间。选择不同的参数值比较结果。(了解并选用特定分布的随机数生成器产生需要的输入。)

A.13: 细木棒的问题



将 k 根非常纤细 (就是说可以忽略它们的直径) 的长度不等的细木棒一根一根地依次抛在桌面上, 判断哪些会在 “顶上” 的位置, 即没有任何其它木棒压住它们。

用一个四元组的序列表示木棒被抛下的过程, 每个四元组含四个数值, 分别表示该木棒落下后两个端点在桌面上的坐标 (为了简单假设均为整数, 整个桌面为 1024×1024), 当然这也就确定了该木棒的长度。每个四元组在序列中的顺序就是该木棒抛下的顺序。写一个程序输出所有顶上的木棒的序号, 并输出一个类似于附图的上下位置关系图。(为了简单, 假设木棒最多 100 根, 你可以人工给出四元组序列, 也可以利用某种随机数生成器生成。)

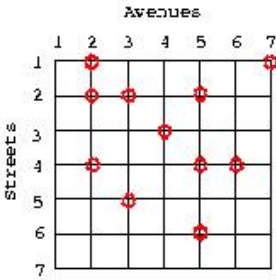
A.14: 等价测试问题

编一个程序实现解决 1BP 等价问题的算法。

A.15: 最小边割集问题

编一个程序实现解决图中最小边割集问题的随机优化算法。

A.16: 聚会地点问题



一个规划的很完美的城市如图中的示例，城市中的街道构成完美的方格阵，假设每个相邻街口的距离恰好是 1。一群朋友定期聚会的地点总是安排在某个街口，而且要求与每个人住处的距离之和为最小。每次聚会参与的人可能变化。编写一个程序，对于每次确定的聚会参加人（输入为各人的住处坐标）选定合适的聚会地点。如果多于一个街口满足上述条件，则选择其中编号和为最小的那个。

A.17: 建木围栏问题

假设在一片林地上生长着 n 株树，为了保护它们，需要用围栏将它们围起来，但是为了获得建造围栏的材料，必须选择其中部分树木砍伐。对于每棵树我们有下列信息：位置（假设这片林地完全是平坦的）、它的价值、用它做材料可以建的围栏的长度。写一个程序，对特定的树木集合，选择其一个子集，将子集中的树木砍伐后可以建立保护剩余所有树木的围栏，且树木价值的损失最小。（如果两个或更多的子集同时满足上述要求，则选择其中棵数少的。）

A.18: 整数排列问题

输入 $\{1,2,\dots,n\}$ 的任意一种排列，你可以想像将这个序列首尾相接排成一个圆圈。你不能随便更改其内部的排列次序，但是允许你选择任意一个长度为 4 的子串，颠倒其中数字的顺序。编写一个程序，对于任意输入的一个上述序列，经过若干次上述颠倒操作，使序列成为自然递增序列；或者判断其无解。（例如：对于输入 “8, 1, 2, 3, 7, 10, 4, 6, 5, 9”，则操作过程为：→ “8, 1, 2, 3, 7, 10, 9, 5, 6, 4” → “4, 8, 1, 2, 3, 7, 6, 5, 9, 10” → “4, 3, 2, 1, 8, 7, 6, 5, 9, 10” → “4, 3, 2, 1, 5, 6, 7, 8, 9, 10” → “1, 2, 3, 4, 5, 6, 7, 8, 9, 10”）