算法分析与设计

课程设计题

倪卫明
weimingni@fudan.edu.cn

2025, May

算法设计课程设计题

用 C/C++ 或 Java 或 Python 程序设计语言, 选择合适的数据结构实现算法.

一. 编程实现下列排序算法 (至少实现其中的五种算法)

- 选择排序 (SelectionSort);
- 插入排序 (InsertionSort);
- 合并排序 (MergeSort);
- 快速排序 (QuickSort);
- 基数排序 (RadixSort)
- 希尔排序 (ShellSort)
- 随机化快速排序 (RandomizedSort)

(a) 给出各种算法时间、空间复杂性的渐近表达式.

(b) 随机产生测试数据文件, 其中分别包含 $n = 10^2, 10^3, 10^4$ 个正整数作为测试数据, 比较各种排序算法的时间复杂度, 画出 $T(n)$ 关于问题规模的曲线图.

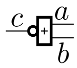
二. 从下列 20 个算法设计问题, 任意选做 2 题.

1. **问题**: 计算与非树的输出.

   与非树是一个完全或几乎完全二进制树, 它具有下述性质:

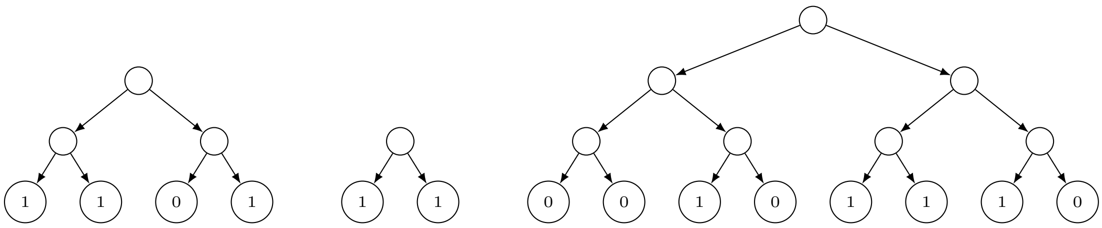
   (a) 每颗树叶可标注为 0 或 1;

   (b) 树的每个内部节点表示一个与非门. 与非门是一个包含两个输入的逻辑门, 它的输入输出关系如下:

	a	b	c
	0	0	1
	0	1	1
	1	0	1
	1	1	0
	x	-	x
	-	x	x

   其中, "-" 对应无输入, 输入 "x" 代表 0 或 1, 输出 "x" 表示输出等于输入值 (对应仅有一个树叶情形).

   例如: 下列三个图对应的与非树最终输出分别为: 1,0,1.



   给定与非树的 $n(k=\lceil\log_2 n\rceil,\ 2^{k-1}\leq n\leq 2^k)$ 个树叶的取值, 求计算其输出值的多项式时间复杂性的算法.

2. **问题**: 约束调度. 假设有一台超级计算机一次可完成一项任务, 现在有 n 项任务 $\mathcal{J}=\{1,2,\ldots,n\}$ 需要完成, 完成这些任务必须满足下列约束:

   (a) 设完成任务 $j\in\mathcal{J}$ 所需的时间是 t_j, 每一项任务一旦开始就不能中断, 直到完成为止.

   (b) 完成这些任务须满足约束, 通常用有向图 $G=(\mathcal{J},\mathcal{E})$ 来表示约束, 其中顶点集为 $\mathcal{J}$, 边集 $\mathcal{E}$ 中的每条边 $\langle j,k\rangle$ 意味着任务 j 必须在任务 k 之前完成.

   (c) 当给定 n 项任务的可行排序 $\{j_1,j_2,\ldots,j_k,\ldots,j_n\}$ 后 (满足任务约束), 完成第 k 项任务 $j_k\in\mathcal{J}$ 的费用为 $c_k(t)$, t 表示任务 j_k 的完成时刻, 每项任务的完成费用是关于完成时刻 t 的单调递增函数.

   要求给出 n 项任务的 "最好" 排序方式, 使得在满足约束前提下完成所有任务的总费用最小, 即:

$$\min \sum_{j_k\in\mathcal{J}} c_k\left(f(j_k)\right)$$

   其中 $f(j_k)$ 表示任务 j_k 结束时刻 (第一项任务开始时刻为 0). 显然, 若边 $\langle j_i,j_k\rangle\in\mathcal{E}$, 则 $f(j_i)<f(j_k)$.

   为了方便起见, 设每项任务完成的时间间隔是整数, 完成所有任务的总时间为 T.

- 任务的完成时间用 n 元数组 $L[1...n]$ 表示, 其中 $L[i]$ 为完成任务 $i(1 \le i \le n)$ 所需时间, $T = \sum_{i=1}^{n} L[i]$.
- 约束关系用 $n \times n$ 邻接矩阵 $\mathbf{A} = [a_{ij}]$ 表示, 其中

$$a_{ij} = \begin{cases} 1, & \text{存在若任务}i\text{指向任务}j\text{的边;} \\ 0, & \text{否则;} \end{cases}$$

- 用 $n \times T$ 矩阵 $\mathbf{C} = [c_{ij}]$ 表示任务的完成费用, 其中 c_{ij} 表示第 i 项任务若在 j 时刻完成所需的费用.

采用动态规划法设计算法, 算法输出: 满足约束的最优任务排序及完成任务的最小费用. 分析算法的时间、空间复杂性. 扩展讨论, 若有两台相同的超级计算机在同样约束下完成 n 项任务的算法设计.

3. **问题**: 多色生成树.

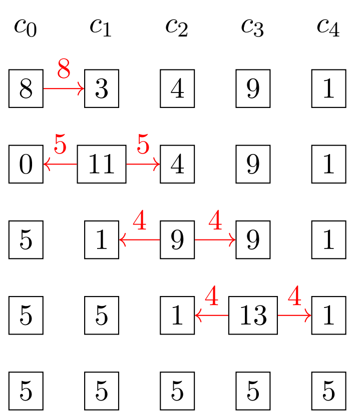
给定连通、无向简单图 $G = (V, E)$, 图中每条边用红或蓝着色, 其中 $|V| = n, |E| = m$.

(a) 设计多项式时间复杂度的算法, 寻找包含最少蓝色边的生成树, 实现该算法, 分析算法的时间、空间复杂性.

(b) 设计多项式时间复杂度的算法, 寻找包含最多蓝色边的生成树, 实现该算法, 分析算法的时间、空间复杂性.

(c) 给定正整数 k, 设计确定图 G 是否存在恰好包含 k 条蓝色边的生成树的算法, 分析算法的复杂性 (表示成 n 和 m 的函数的阶), 并以例子说明算法是如何运行的.

4. **问题**: 有 n 个瓮分别用 $0, 1, \ldots, n-1$ 编号, 在这 n 个瓮中各放置了一些球, 各个瓮中放置的球的数量为 $c_j(j = 0, 1, \ldots, n-1)$, 其中 $c_j \ge 0$, 设 $N = \sum_{j=0}^{n-1} c_j$ 能被 n 整除, 现移动瓮中的球使得最后每个瓮中球的数量相同, 并且约定若某次移动选择第 i 号瓮, 且移动数量为 k 个球, 则要求第 i 号瓮中至少包含 $2k$ 个球, 移动时将 k 个球将放入第 $i-1$ 号瓮中, 另 k 个球放入第 $i+1$ 个瓮中; 只有当 $i = 0$ 或 $n-1$ 时, 只需移动 k 个球到 1 号瓮或 $n-2$ 号瓮中. 例如: 有五个瓮, 初始时瓮中的球数量分别为: $8, 3, 4, 9, 1$, 移动过程如下:



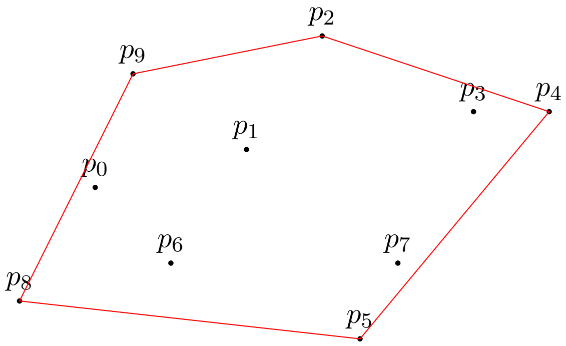
设计一个算法, 求解符合要求的移动, 而且要求移动次数越少越好, 进行空间、时间复杂度分析.

5. 求解查找假币问题.

问题描述 有 $n(n > 3)$ 个硬币, 其中有一个假币, 且假币较轻, 采用天平称重方式找到这个假币, 并给出操作步骤. (假设 n 个硬币编号 $0 \sim n-1$, 用数组存放这些硬币的重量, 不妨设真币中 2, 假币重 1.8, 假币采用随机方式产生.)

6. **问题**: (最大公因子) 利用算法实现求两个大整数的最大公因子.

7. **问题**: (点集 Q 的凸包 (Convex Hull)) 给定一个点集 Q, 设计算法求一个最小凸多边形, 满足 Q 中的点或者在多边形边上或者在其内. 例如, 图中由红色线段表示的多边形就是点集 $Q=\{p_0,p_1,\ldots,p_9\}$ 的凸包。



8. **问题**: (装箱问题) 有一些产品分别包装在 n 个长方体小纸箱中, 设第 i 个产品的包装箱大小为 $a_i\times b_i\times h(i=1,2,\ldots,n)$. 现有长宽高分别为 M,N,h 的大箱子若干, 其中 a_i,b_i,M,N 都是正整数, 且 $a_i\le M,b_i\le N(i=1,2,\ldots,n)$, 请设计一个装箱算法, 将这些小箱子装入大箱子, 使得所用大箱子的数量越少越好. (假定所有大小箱子的高度一样, 即只考虑平面装箱问题).

9. **问题** 8: (凸多边形最优三角剖分) 多边形是平面上一条分段线性的闭曲线, 当一个简单多边形及其内部构成一个闭凸集时, 称该简单多边形为凸多边形. 给定一个 n 边的凸多边形 $P=v_0v_1v_2\cdots v_{n-1}$, 其中 v_i 表示凸多边形的顶点, 且它与 v_{i-1} 和 v_{i+1} 相邻, 当 $i=0$ 时, v_0 与 v_{n-1},v_1 相邻; 当 $i=n-1$ 时, v_{n-1} 与 v_0,v_{n-2} 相邻. 若 v_iv_j 是多边形上不相邻的顶点时, 称 v_iv_j 为多边形的弦. 定义凸多边形的三角形剖分是将一个凸多边形分割成相互不重叠的三角形的弦的集合 T. 例如图所示的是凸五边形 $P=v_0v_1v_2v_3v_4$ 的两种三角形剖分. 所谓最优三角剖分问题是指: 给定一个凸多边形 $P=v_0v_1v_2\cdots v_{n-1}$, 以及定义由多边形的边和弦构成的三角形上的权函数 ω, 求一个三角剖分使得剖分中各三角形权函数之和最小. 若将权函数取为三角形的周长, 相应的最优三角剖分称为最小弦长三角剖分. 以最小弦长三角剖分为例, 给出最小三角剖分的算法和实现.

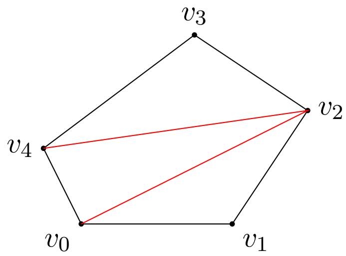


图 (a): 三角形剖分 $T_1=\{v_0v_2,v_2v_4\}$

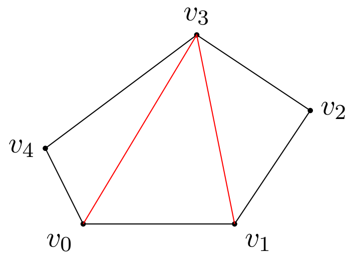


图 (b): 三角形剖分 $T_2=\{v_0v_3,v_3v_1\}$

10. **问题**: 给定 N 个大小不等的圆 $C_1,C_2,\cdots,C_N$, 现将这 N 个圆排进一个矩形框中, 且要求个圆与矩形框的底边相切. 圆排列问题要求从 N 个圆的所有排列中选出使得包含所有圆的矩形框长度最短的排列. 采用回溯法设计算法, 输入数组 $R[1,2,...,N]$ 存储 N 各圆的半径, 输出相应于最优解的 N 个圆的排列.

11. **问题**: (哈夫曼编/译码器) 给定任一文本文件实现哈夫曼编码.
    实现要求:

    (1) 初始化: 对文本文件 (xxx.txt) 中的字符进行统计, 建立哈夫曼树 (存储格式自行设计), 存储在 (xxx.hfmtree) 中.

    (2) 编码: 利用已建立的哈夫曼树对文本文件内容进行编码, 将结果文件存入 (xxx.code) 中.

    (3) 译码: 利用哈夫曼树将编码文件 (xxx.code) 进行译码, 结果存入 (xxx-decode.txt) 中.

12. **问题**: 有重复元素的排列问题.

问题描述: 设 $R = \{r_1, r_2, \cdots, r_n\}$ 是要进行排列的 n 个元素. 其中元素 $r_1, r_2, \cdots, r_n$ 可能相同. 试设计一个算法, 列出 R 的所有排列.

算法设计: 给定 n 及待排列的 n 个元素. 计算出这 n 个元素的所有不同排列.

数据输入: 由文件 "input.txt" 提供输入数据, 文件的第 1 行是元素个数 n, $1 \le n \le 500$. 接下来的 1 行是待排列的 n 个元素.

结果输出: 将计算出的 n 个元素的所有不同排列输出到文件 "output.txt". 文件最后 1 行中的数是排列总数.

输入文件示例	输出文件示例
input.txt	output.txt
4	aacc
aacc	acac
	acca
	caac
	caca
	ccaa
	6

13. **问题**: 石子合并问题

问题描述: 在一个圆形操场的四周摆放着 n 堆石子. 现要将石子有次序地合并成一堆. 规定每次只能选相邻的 2 堆石子合并成新的一堆, 并将新的一堆石子数记为该次合并的得分. 试设计一个算法, 计算出将 n 堆石子合并成一堆的最小得分和最大得分.

算法设计: 对于给定 n 堆石子, 计算合并成一堆的最小得分和最大得分.

数据输入: 由文件 "input.txt" 提供输入数据. 文件的第 1 行是正整数 $n(1 \le n \le 100)$, 表示有有 n 堆石子. 第 2 行有 n 个数, 分别表示每堆石子的个数.

结果输出: 将计算结果输出到文件 "output.txt". 文件第 1 行的数是最小得分, 第 2 行中的数是最大得分.

输入文件示例	输出文件示例
input.txt	output.txt
4	43
4 4 5 9	54

14. **问题**: 最小长度电路板排列问题.

问题描述: 最小长度电路板排列问题是大规模电子系统设计中提出的实际问题. 该问题的提法是, 将 n 块电路板以最佳排列方案插入带有 n 个插槽的机箱中. n 块电路板的不同的排列方式对应于不同的电路板插入方案.

设 $B = \{1, 2, \cdots, n\}$ 是 n 块电路板的集合. 集合 $L = \{N_1, N_2, \cdots, N_m\}$ 是 n 块电路板的 m 个连接块. 其中每个连接块 N_i 是 B 的一个子集, 且 N_i 中的电路板用同一根导线连接在一起. 在最小长度电路板排列问题中. 连接块的长度是指该连接块中第 1 块电路板到最后 1 块电路板之间的距离.

试设计一个回溯法, 找出所给 n 个电路板的最佳排列, 使得 m 个连接块中最大长度达到最小.

算法设计: 对于给定的电路板连接块, 设计一个算法, 找出所给 n 个电路板的最佳排列, 使得 m 个连接块中最大长度达到最小.

数据输入: 由文件 "input.txt" 给出输入数据. 第 1 行有 2 个正整数 n 和 $m(1 \leq m, n \leq 20)$. 接下来的 n 行中, 每行有 m 个数. 第 k 行的第 j 个数为 0 表示电路板 k 不在连接块 j 中, 为 1 表示电路板 k 在连接块 j 中.

结果输出: 将计算的电路板排列最小长度及其最佳排列输出到文件 "output.txt". 文件的第一行是最小长度; 接下来的 1 行是最佳排列.

输入文件示例	输出文件示例
input.txt	output.txt
8 5	4
1 1 1 1 1	5 4 3 1 6 2 8 7
0 1 0 1 0	
0 1 1 1 0	
1 0 1 1 01 0 1 0 0	
1 1 0 1 0	
0 0 0 0 1	

15. **问题**: 最小重量机器设计问题.

问题描述: 设某一机器由 n 个部件组成, 每种部件都可以从 m 个不同的供应商处购得. 设 w_{ij} 是从供应商 j 处购得的部件 i 的重量, c_{ij} 是相应的价格. 试设计一个算法, 给出总价格不超过 c 的最小重量机器设计.

算法设计: 对于给定的机器部件重量和机器部件价格, 计算总价格不超过 d 的最小重量机器设计.

数据输入: 由文件 "input.txt" 给出输入数据. 第一行有 3 个正整数 n,m 和 d. 接下来的 $2n$ 行, 每行 n 个数. 前 n 行是 c, 后 n 行是 w.

结果输出: 将计算的最小重量及每个部件的供应商输出到文件 "output.txt".

输入文件示例	输出文件示例
input.txt	output.txt
3 3 4	4
1 2 3	1 3 1
3 2 1	
2 2 2	
1 2 3	

```
3 2 1
2 2 2
```

16. 问题: 求解矩阵最小路径和问题

问题描述: 给定 m 行 n 列的矩阵, 从左上角开始每次只能向右或者向下移动, 最后到达右下角的位置, 路径上的所有数字累计起来作为这条路径的路径和. 写一个算法求所有路径中的最小路径和.

例如: 以下矩阵中的路径 $1 \to 3 \to 1 \to 0 \to 6 \to 1 \to 0$ 是所有路径中路径和最小的, 返回结果是 12.

$$\begin{matrix} 1 & 3 & 5 & 9 \\ 8 & 1 & 3 & 4 \\ 5 & 0 & 6 & 1 \\ 8 & 8 & 4 & 0 \end{matrix}$$

17. 问题: 求解添加最少括号数问题

问题描述: 括号序列有 (),{},[ ] 组成, 例如 "(([{}]))()" 是合法的, 而 "(}{)","(}(}" 和 "({)}" 都是不合法的. 如果一个序列不合法, 设计一个算法求添加最小括号数, 使这个序列变成合法的.

例如: "(}(}" 最少添加 4 个括号变成合法, 即变为 "(){}(){}".

18. 问题: 求解一个序列中出现次数最多的元素问题.

问题描述: 给定 n 个正整数, 设计算法从中找出出现次数最多的数, 若这样的数有多个, 请输出其中最小的一个.

数据输入: 输入的第 1 行中只有一个正整数 $n(1 \le n \le 10000)$, 表示数字的个数; 输入的第 2 行中有 n 个整数 $s_1, s_2, \cdots, s_n (1 \le s_i \le 10000,\ 1 \le i \le n)$. 相邻的数用空格分隔.

结果输出: 输出 n 个出现次数中出现次数最多的数. 如果这样的数有多个, 输出其中最小的一个.

输入文件示例	输出文件示例
input.txt	output.txt
6	10
10 1 10 20 30 20	

19. 问题: 求解删数问题.

问题描述: 给定共有 n 位的正整数 d, 去掉其中任意 $k \le n$ 个数字后剩下的数字按原次序排列组成一个新的正整数. 对于给定的 n 位正整数 d 和正整数 k, 设计算法找出剩下数字组成的新数最小的删数方案.

例如, $d = 5004321$, $k = 2$, 删除 "500543" 后剩下的最小删数 21.

20. 问题: 求解股票经纪人问题

问题描述: 股票经纪人要在一群人 (n 个人编号为 $0 \sim n-1$) 中散布一个谣言, 传言只在认识的人中间传递. 题目中给出了人与人的认识关系以及传言在某两个认识的人中传递所需的时间. 设计一个算法求出哪个人为起点可以耗时最短的情况下让所有人收到消息.

例如, 输入数据: 第 1 行 2 个正整数, $n = 4$(人数),
后续输入 m 组数据, 如 $m = 4$(边数, 认识的人及传递信息的时间), 4 条边情形如下:
0 1 2 (第 0 人与第 1 人认识, 他们之间传递谣言需要时间为 2)
0 2 5
0 3 1
2 3 3
输出数据: 选择第 3 人为起点, 让所有人收到消息最短所需时间 3.