**数据结构课程设计题目**

1. 排序算法的性能分析

**问题描述**

设计一个测试程序，比较几种内部排序算法的关键字比较次数和移动次数以取得直观感受。

**基本要求**

1. 对冒泡排序、直接排序、选择排序、箱子排序、堆排序、快速排序及归并排序算法进行比较。
2. 待排序表的表长不小于100，表中数据随机产生，至少用5组不同数据作比较，比较指标：关键字参加比较次数和关键字的移动次数（关键字交换记为3次移动）。
3. 输出比较结果。

**选做内容**

1. 对不同表长进行比较。
2. 验证各算法的稳定性。
3. 输出界面的优化。

2. 排序算法思想的可视化演示—1

基本要求

排序数据随机产生，针对随机案例，对冒泡排序、箱子排序、堆排序、归并算法，提供排序执行过程的动态图形演示。

3. 排序算法思想的可视化演示—2

基本要求

排序数据随机产生，针对随机案例，对插入排序、选择排序、基数排序、快速排序算法，提供排序执行过程的动态图形演示。

4. 线性表的实现与分析

基本要求

1. 设计并实现线性表。
2. 线性表分别采取数组（公式化描述）、单链表、双向链表、间接寻址存储方式
3. 针对随机产生的线性表实例，实现线性表的插入、删除、搜索操作动态演示(图形演示)。

5. 等价类实现及其应用

问题描述：某工厂有一台机器能够执行n个任务，任务i的释放时间为ri（是一个整数），最后期限为di（也是整数）。在该机上完成每个任务都需要一个单元的时间。一种可行的调度方案是为每个任务分配相应的时间段，使得任务i的时间段正好位于释放时间和最后期限之间。一个时间段不允许分配给多个任务。

基本要求：

使用等价类实现以上机器调度问题。

等价类分别采取两种数据结构实现。

6. 一元稀疏多项式计算器

**问题描述**

设计一个一元稀疏多项式简单计算器。

**基本要求**

一元稀疏多项式简单计算器的基本功能是：

1. 输入并建立多项式；
2. 输出多项式，输出形式为整数序列：n，c1，e1，c2，e2，…,cn，en，其中n是多项式的项数，ci，ei，分别是第i项的系数和指数，序列按指数降序排序；
3. 多项式a和b相加，建立多项式a+b；
4. 多项式a和b相减，建立多项式a-b；
5. 计算多项式在x处的值；
6. 计算器的仿真界面

7. 线索二叉树

**问题描述：**

二叉树在一般情况下无法直接找到某结点在某种遍历序列中的前驱和后继结点。若增加指针域来存放前驱和后继结点信息，将大大降低存储空间的利用率。考察 n 个结点的二叉树，其中有 n+1 个空指针域，它们可以被用来存放“线索”。加了线索的二叉树称为线索二叉树。

对一棵非线索二叉树以某种次序遍历使其变为一棵线索二叉树的过程称为二叉树的线索化。线索化是将二叉链表中的空指针改为指向结点前驱或后继的线索，而一个结点的前驱或后继结点的信息只有在遍历时才能得到，因此线索化的过程即为在遍历过程中修改空指针的过程。

**基本要求**：

实现线索二叉树的建立、插入、删除和遍历

8.家族关系查询系统

建立家族关系数据库，实现对家族成员的相关查询

**基本要求**：

(1)建立家族关系并能存储到文件中

(2)实现家族成员的添加

(3)可以查询家族成员的双亲、祖先、兄弟、孩子和后代等信息

(4)该家族树宽度至少5，高度至少5。

9. 简单计算器

**基本要求：**

输入：不含变量的数学表达式的中缀形式，可以接受的操作符包括+、-、\*、/、%和（、）。

输出：（1） 如果表达式正确，则输出表达式的结果，如果表达式非法，则输出错误信息。

注： 输入/输出形式可采取终端设备输入/输出，也可采用文件输入/输出,一个文件中可包含多个表达式

（2）计算器的仿真界面

10. 迷宫问题-1

**问题描述**

以一个m\*n的长方阵表示迷宫，0和1分别表示迷宫中的通路和障碍。设计一个程序，对任意设定的迷宫，求出一条从入口到出口的通路，或得出没有通路的结论。

**基本要求**

1. 实现一个以链表作存储结构的栈类型，然后编写一个求解迷宫的非递归程序。求得的通路一三元组（i，j，d）的形式输出，其中：（i，j）指示迷宫中的一个坐标，d表示走到下一坐标的方向。
2. 编写递归形式的算法，求得迷宫中所有可能的通路；
3. 以方阵形式输出迷宫及其通路

11． 迷宫问题-2

程序开始运行时显示一个迷宫地图，迷宫中央有一只老鼠，迷宫的右下方有一个粮仓。游戏的任务是使用键盘上的方向键操纵老鼠在规定的时间内走到粮仓处。

要求：

（1）老鼠形象可辨认，可用键盘操纵老鼠上下左右移动；

（2）迷宫的墙足够结实，老鼠不能穿墙而过；

（3）正确检测结果，若老鼠在规定时间内走到粮仓处，提示成功，否则提示失败；

（4）添加编辑迷宫功能，可修改当前迷宫，修改内容：墙变路、路变墙；

（5）找出走出迷宫的所有路径，以及最短路径；

利用序列化功能实现迷宫地图文件的存盘和读出等功能。

12. 应用等价类生成随机迷宫并寻找迷宫路径

问题描述：

使用等价类来构造一个N×N的从左上角到右下角只有一条路径的随机迷宫，然后在这一迷宫上寻找迷宫路径。该设计共包含如下四个部分：

① 等价类数据结构的设计和实现

② 构建随机迷宫

③ 寻找迷宫路径

④ 将迷宫和路径用图形方式画出

用图形方式将上述算法获得的随机迷宫及其上的最短路径画出。用线段来表示迷宫中的墙，用在每个方格中心的点来表示路径。

13、跳表（Skip List）的实现与分析

基本要求

1. 构造并实现跳表（Skip List）的ADT ADT中应包括初始化、查找、插入、删除等基本操作。

② 分析各基本操作的时间复杂性。

③ 针对一个实例实现Skip List的动态演示(图形演示)。

14. LZW压缩算法及应用

基本要求

① 在一个文本文件上实现LZW压缩和解压缩，其中每个字符就是该文本的8位ASCII码。

② 在实现LZW过程中需要仔细考虑如何在编译表中找到匹配或找不到匹配，需要注意匹配算法的时间、空间开销。

③ (选做)应用LZW算法实现256色灰度BMP图像文件的压缩和解压缩。

15. 二叉树的实现及分析

基本要求

（1）设计实现链表存储的二叉树ADT

（2）实现基本操作实现过程（前序遍历、中序遍历、后序遍历、层序遍历等）的动态演示(图形演示)。

（3）应用二叉树，实现信号放大器的设置。

16. 应用堆实现一个优先队列并实现作业的优先调度

问题描述

优先队列***priority queue***是一种可以用于很多场合的数据结构，应用堆结构设计并实现一个优先队列。应用该优先队列实现作业的优先调度：

一个作业ti =（si，ei），si为作业的开始时间（进入时间），ei为作业的结束时间（离开时间）。作业调度的基本任务是从当前在系统中的作业中选取一个来执行，如果没有作业则执行nop操作。本题目要求的作业调度是基于优先级的调度，每次选取优先级最高的作业来调度，优先级用优先数（每个作业一个优先数pi）表征，优先数越小，优先级越高。作业ti进入系统时，即si时刻，系统给该作业指定其初始优先数pi = ei - si，从而使越短的作业优先级越高。该优先数在作业等待调度执行的过程中会不断减小，调整公式为：pi = pi - wi，其中的wi为作业ti的等待时间：wi = 当前时间-si。一旦作业被调度，该作业就一直执行，不能被抢占，只有当前执行作业指向完成时，才产生下一轮调度。所以可以在每次调度前动态调整各作业的优先数。

编程实现这样一个作业调度系统。

基本要求

1. 以堆结构实现优先队列。②作业集合中的各作业随机生成，根据作业的s属性和e属性动态调整作业队列，不断加入作业，作业结束删除作业。

③要对作业调度的结果给出清晰的输出信息，包括：何时作业进入，何时调度哪个作业，何时离开，每个作业等待多长时间，优先数的动态变化情况等。

④

17. 哈夫曼编/译码器

**问题描述**

利用哈夫曼编码进行信息通信可以大大提高信道利用率，缩短信息传输时间，降低传输成本。但是，这要求在发送端通过一个编码系统对待传数据预先编码；在接收端将传来的数据进行译码（复原）。对于双工信道（即可以双向传输信息的信道），每端都需要一个完整的编/译码系统。试为这样的信息收发站写一个哈夫曼码的编译码系统。

**基本要求**

一个完整的系统应具有以下功能：

1. I：初始化（Initialization）。从终端读入字符集大小n及n个字符和m个权值，建立哈夫曼树，并将它存于文件hfmtree中。
2. C：编码（Coding）。利用已建好的哈夫曼树（如不在内存，则从文件hfmtree中读入），对文件tobetrans中的正文进行编码，然后将结果存入文件codefile中。
3. D：解码（Decoding）。利用已建好的哈夫曼树将文件codefile中的代码进行译码，结果存入文件textfile中。
4. P：打印代码文件（Print）。将文件codefile以紧凑格式显示在终端上，每行50个代码。同时，将此字符形式的编码文件写入文件codeprint中。
5. T：打印哈夫曼树（Tree printing）。将已在内存中的哈夫曼树以直观的方式（树或凹入表形式）显示在终端上，同时将此字符形式的哈夫曼树写入文件treeprint中。

18. AVL树的实现及分析

**基本要求**

（1） 编写AVL树判别程序，并判别一个二叉搜索树是否为AVL树。二叉搜索树用其先序遍历结果表示，如：5，2，1，3，7，8。

（2）实现AVL树，其上的基本操作包括：Search，Insert，Delete，和Ascend；

（3）实现基本操作的动态演示(图形演示)。

（4）扩展：

a.实现带索引的AVL搜索树，实现其上的基本操作：Search，Insert，Delete，IndexSearch，IndexDelete和Ascend。前5种函数的时间复杂性应为O(logn)，最后一种函数的时间复杂性应为O(n)。

b. 搜索树中有一些元素的关键值相同。

19.图的问题

问题描述：

首先基于班级的所有同学做为图G的顶点集合V，然后，邀请班级的同学，每人书写三个同学的名字，做为图G中的边，循环收集大家写的名字，构建图G的边集合E。

基本要求：

基于图G，计算同学之间的最短路径，输出所有同学之间的最短路径。

20. 箱子装载问题

问题描述：

在箱子装载问题中，有若干个容量为c的箱子和n个待装载入箱子中的物品。物品i需占s[i]个单元（0<s[i]≤c）。所谓成功装载（feasible packing），是指能把所有物品都装入箱子而不溢出，而最优装载（optimal packing）则是指使用了最少箱子的成功装载。

基本要求：

至少给出求解箱子装载问题的三种方案，并进行性能比较。

21. B-树的实现及分析

基本要求：

（1）实现在B-树上的查找，并分析其时间复杂性。

（2）实现B-树的ADT，包括其上的基本操作：结点的加入和删除。

（3）要求B-树结构中的M=3或5，实现其中的一种即可。

（4）实现基本操作的动态演示(图形演示)。

22. 图的实现与分析—1

**问题描述：**

分别对有向图、无向图、带权有向图、带权无向图实现对图的基本操作（创建、求顶点的度数、增加/删除边、判断边是否存在、DFS、BFS、判断是否连通、连通构件的标识，求生成树等）。

**基本要求：**

图使用邻接矩阵存储。

提供随机案例，对任意随机案例，实现DFS和BFS实现过程的动态演示(图形演示)。

对DFS提供递归与非递归两种方法的实现，并通过输出进行性能比较。

23. 图的实现与分析—2

**问题描述：**

分别对有向图、无向图、带权有向图、带权无向图实现对图的基本操作（创建、求顶点的度数、增加/删除边、判断边是否存在、DFS、BFS、判断是否连通、连通构件的标识，求生成树等）。

**基本要求：**

图使用邻接链表存储。

提供随机案例，对任意随机案例，实现DFS和BFS实现过程的动态演示(图形演示)。

对DFS提供递归与非递归两种方法的实现，并通过输出进行性能比较。

24. 校园导游

**问题描述：**

用无向网表示校园景点平面图，图中顶点表示主要景点，存放景点的编号、名称、简介等信息，图中的边表示景点间的道路，存放路径长度等信息。要求能够回答有关景点介绍、游览路径等问题。

**基本要求：**

① 查询任意景点的相关信息；

② 查询图中任意两个景点间的最短路径。

③ 查询图中任意两个景点间的所有路径。

④ 增加、删除、更新有关景点和道路的信息。

（选作）\* 求多个景点的最佳（最短）游览路径。

25. 全国交通咨询模拟

**问题描述**

处于不同目的的旅客对交通工具有不同的要求。例如，因公出差的旅客希望在旅途中的时间尽可能地短，出门旅游的旅客则期望旅费尽可能省，而老年旅客则要求中转次数最少。编织一个全国城市间的交通资讯程序，为旅客提供两种或三种最优决策的交通咨询。

**设计要求**

1. 提供对城市信息进行编辑（如添加或删除）的功能。
2. 城市之间有两种交通工具：火车和飞机。提供对列车时刻表和飞机航班进行编辑（增设或删除）的功能。
3. 提供两种最优决策：最快到达和最省钱到达。全程只考虑一种交通工具。
4. 旅途中耗费的总时间应该包括中转站的等候时间。
5. 咨询以用户和计算机的对话方式进行。由用户输入起始站、终点站、最优决策原则和交通工具。输出信息：最快需要多长时间才能到达或者最少需要多少旅费才能到达，并详细说明依次于何时乘坐哪一趟列车或那一次班机到何地。

**实现提示**

1. 对全国城市交通图和列车时刻表及飞机航班表进行编辑，应该提供文件形式输入和键盘输入两种方式。飞机航班表的信息应包括：起始站的出发时间、终点站的到达时间和票价；列车时刻表则需根据交通图给出各个路段的详细信息，例如：对从北京到上海的火车，需给出北京至天津、天津至徐州及徐州至上海各段的出发时间、到达时间及票价等信息。
2. 以邻接表座交通图的存储结构，表示边的结构内除含有邻接点的信息外，还应包括交通工具、路程中耗费的时间和花费以及出发和到达的时间等多种属性。
3. 增加旅途中转次数最少的最优决策。

26. 公交线路上优化路径的查询

**问题描述**

最短路径问题是图论中的一个经典问题，其中的Dijkstra算法一直被认为是图论中的好算法，但有的时候需要适当的调整Dijkstra算法才能完成多种不同的优化路径的查询。

对于某城市的公交线路，乘坐公交的顾客希望在这样的线路上实现各种优化路径的查询。设该城市的公交线路的输入格式为：

线路编号：起始站名(该站坐标)；经过的站点1名(该站坐标)；经过的站点2名(该站坐标)；……；经过的站点n名(该站坐标)；终点站名(该站坐标)。该线路的乘坐价钱。该线路平均经过多少时间来一辆。车速。

例如：63：A(32,45)；B(76,45)；C(76,90)；……；N(100,100)。1元。5分钟。1/每分钟。

假定线路的乘坐价钱与乘坐站数无关，假定不考虑公交线路在路上的交通堵塞。

对这样的公交线路，需要在其上进行的优化路径查询包括：任何两个站点之间最便宜的路径；任何两个站点之间最省时间的路径等等。

**基本要求：**

① 根据上述公交线路的输入格式，定义并建立合适的图模型。

② 针对上述公交线路，能查询获得任何两个站点之间最便宜的路径，即输入站名S，T后，可以输出从S到T的最便宜的路径，输出格式为：线路x：站名S，…，站名M1；换乘线路x：站名M1，…，站名M2；…；换乘线路x：站名MK，…，站名T。共花费x元。

③ 针对上述公交线路，能查询获得任何两个站点之间最省时间的路径（不考虑在中间站等下一辆线路的等待时间），即输入站名S，T后，可以输出从S到T的考虑在中间站等下一辆线路的等待时间的最省时间的路径，输出格式为：线路x：站名S，…，站名M1；换乘线路x：站名M1，…，站名M2；…；换乘线路x：站名MK，…，站名T。共花费x时间。

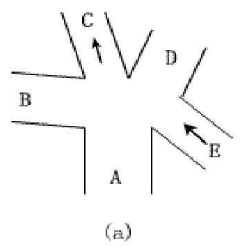
④ 针对上述公交线路，能查询获得任何两个站点之间最省时间的路径（要考虑在中间站等下一辆线路的等待时间），即输入站名S，T后，可以输出从S到T的考虑在中间站等下一辆线路的等待时间的最省时间的路径，输出格式为：线路x：站名S，…，站名M1；换乘线路x：站名M1，…，站名M2；…；换乘线路x：站名MK，…，站名T。共花费x时间。

(4) 实现提示

需深入考虑，应根据不同的应用目标，即不同的优化查询来建立合适的图模型。

27. 多叉路口交通灯的管理问题

通常，在十字交叉路口只需设红、绿两色的交通灯便可保持正常的交通秩序，而在多叉路口需设几种颜色的交通灯才能既使车辆相互之间不碰撞，又能达到车辆的最大流通。假设有一个如图(a)所示的五叉路口，其中C和E为单行道。在路口有13条可行的通路，其中有的可以同时通行，如A→B和E→C，而有的不能同时通行，如E→B和A→D。那么，在路口应如何设置交通灯进行车辆的管理呢？



28. 文档集合上的查询

(1) 问题描述

设计数据结构完成在一个文档集合的存储，并构造算法实现其内容的查询。该设计包括三个部分：

(一)应用数据结构完成文档集合的内容(基于单词的)存储，并为下一步的查询建立索引。

(二)就单个单词的查询请求，设计算法进行查询。

(三)对多个单词通过AND和OR构造的复杂查询进行处理(此处可只做两个单词的情况)。

具体情形如下面的例子：

**Example**

*Doc1:* I like the class on data structures and algorithms.

*Doc2:* I hate the class on data structures and algorithms.

*Doc3:* Interesting statistical data may result from this survey.

Here are the answers to some queries:

*Query 1:* data

Doc1, Doc2, Doc3

*Query2:* data AND structures

Doc1, Doc2

*Query 3:* like OR survey

Doc1, Doc3

文档集合上的查询实例

基本要求

① 文档集合中的文档数不能少于20个。

② 数据结构的设计以及查找算法的构造应考虑如何最大程度的提高查询效率。

③ 查询效率的提高应是综合多种查询的，而不是只针对一种查询的优化。

④ 给出查询效率的模拟实验数据。

实现提示

AND和OR查询可转变为单个单词查询结果的组合。

29. 教学计划编制问题

**问题描述**

大学的每个专业都要制订教学计划。假设任何专业都有固定的学习年限，每学年含两学期，每学期的时间长度和学分上限值均相等。每个专业开设的课程都是确定的，可以有任意多门，也可以没有。每门课恰好占一个学期。试在这样的前提下设计一个教学计划编制程序。

**基本要求**

1. 输入参数包括：学期总数，一学期的学分上限，每门课的课程号（固定占3位的字母数字串）、学分和直接先修课的课程号。
2. 允许用户指定下列两种编排策略之一：一是使学生在各学期中的学习负担尽量均匀；二是是课程尽可能地集中在前几个学期中。
3. 若根据给定的条件问题无解，则报告适当的信息；否则，将教学计划输出到用户指定的文件中。计划的表格格式自行设计。

30. 最小生成树—Prim算法

设计实现无向网结构，针对随机无向网实例和随机起点，用PRIM算法的基本思想求解出所有的最小生成树，并给出求解过程的动态图形演示。

可考虑实现不同存储结构上的实现。

31. 最小生成树—Kruskal算法

设计实现无向网结构，针对随机无向网实例和随机起点，用Kruskal算法的基本思想求解出所有的最小生成树，并给出求解过程的动态图形演示。

可考虑实现不同存储结构上的实现。

32. 最短路径—Dijkstra算法

设计实现有向网结构，针对随机有向网实例和随机源点，用Dijkstra算法求解出单源点到其他各顶点的最短路径，给出求解过程的动态演示。

可考虑实现不同存储结构上的实现。

33. 拓扑排序

对给定的AOV网，产生所有的拓扑序列，给出求解过程的动态演示。

34. 成绩分析问题

**问题描述**

录入、保存一个班级学生多门课程的成绩，并对成绩进行分析。

**基本要求**

1. 通过键盘输入个学生的多门课程的成绩，建立相应的文件input.dat。
2. 对文件input.dat中的数据进行处理，要求具有以下功能：

按各门课程成绩排序，并生成相应的文件输出;

计算每人的平均成绩，按平均成绩排序，并生成文件;

求出各门课程的平均成绩、最高分、最低分、不及格人数、60~69分人数、70~79分人数、80~89分人数、90分以上人数;

根据姓名或学号查询某人的各门课成绩，重名情况也能处理.

1. 界面美观

35. 检查网络

**问题描述**：

给定一个计算机网络以及机器间的双向连线列表，每一条连线与允许两端的计算机进行直接的文件传输，其他计算机间若存在一条连通路径，也可以进行间接的文件传输。至少使用两种结构实现一下功能：

任意指定两台计算机，判断它们之间是否可以进行文件传输？

判断整个网络中是否任意两台机器间都可以文件传输？若不可以，请给出当前网络中连通分量的个数及各个连通分量中的机器。

增加两台计算机之间的连线。

36. 调度问题

（1）机器调度

现有*n*件任务和无限多台的机器，任务可以在机器上得到处理。每件任务的开始时间为*si*，完成时间为*fi*， *si* < *fi* 。[*si*, *fi*]为处理任务*i*的时间范围。两个任务*i*，*j*重叠是指两个任务的时间范围区间有重叠，而并非是指*i*，*j*的起点或终点重合。每台机器在任何时刻最多只处理一个任务。最优分配是指使用的机器最少的可行分配方案。

要求：

输入任务个数及每个任务的名称，开始时间，完成时间

给出最优分配方案。

（2）任务调度

现在有n项作业，J1，J2，…Jn，要求按顺序执行，已知各作业对应的运行所需时间分别为t1，t2，…tn，要求这些作业在一个处理器上运行，并且要求完成这n个作业的平均完成时间最小。注：每个作业的完成时间等于作业的等待时间与它的执行时间的和，这里假设一旦开始运行一个作业，那么在该作业完成之前，其他作业都只能等待。

**要求：**

输入作业个数及每个作业的名称，执行时间

给出最优调度方案。

37. 学校超市选址问题（带权有向图的中心点）。

基本要求：对于某一学校超市，其他各单位到其的距离不同，同时各单位人员去超市的频度也不同。请为超市选址，要求实现总体最优。

38. 设计散列表实现电话号码查找系统。

基本要求：

（1）设每个记录有下列数据项：电话号码、用户名、地址；

（2）从键盘输入各记录，分别以电话号码和用户名为关键字建立散列表；

（3）查找并显示给定电话号码的记录；

（4）查找并显示给定用户名的记录。

（5）尝试不同类型处理冲突的方法，考察平均查找长度的变化。

39. 字典树

构建字典树的数据结构，并实现基于树的查找、插入和删除。

40. 扫雷游戏

实现一个N\*M的扫雷游戏。

**41 马的遍历问题：**

设计程序完成如下要求：在中国象棋盘上，对任意位置上放置一个马，均能选择一个合适的路线，使得该棋子能够按照象棋的规则不重复的走过棋盘上的每一位置。

要求：（1）依次输出走过的各位置的坐标

（2）最好能画出棋盘的图形形式，并在其上动态的标注行走过程

（3）程序能方便的移植到其他规格的棋盘上。

**42．**国际象棋

在 8×8的国际象棋棋盘上，如果放置若干个马后，使得整个棋盘的任意空位置上所放置的棋子均能被这些马吃掉，则称这组放置为棋盘的一个满覆盖。若去掉满覆盖中的任意一个棋子都会使这组放置不再是满覆盖，则称这一满覆盖为极小满覆盖。设计程序完成如下要求：

（1）求解一个极小满覆盖

（2）最好能画出棋盘的图形形式，并在其上动态的演示试探过程

（3）程序能方便的移植到其他规格的棋盘上。

**43．在中国象棋上实现上一课题的任务**

要求：除了上一课题的要求外，还要考虑到“别马腿”的规定。

**44．简易五子棋游戏**

设计程序实现一个人机对弈的简单的五子棋游戏。游戏规则如下：在19×19的围棋交叉点上，对弈双方轮流放子，最先在棋盘上摆成（按照水平、垂直或者对角线方向）连续五个子的一方为胜方。

**45．约瑟夫环**

**[问题描述]**

约瑟夫（Joeph）问题的一种描述是：编号为1,2,…,n的n个人按顺时针方向围坐一圈，每人持有一个密码（正整数）。一开始任选一个正整数作为报数上限值m，从第一个人开始按顺时针方向自1开始顺序报数，报到m时停止报数。报m的人出列，将他的密码作为新的m值，从他在顺时针方向上的下一个人开始重新从1报数，如此下去，直至所有人全部出列为止。试设计一个程序求出出列顺序。

**[基本要求]**

利用单向循环链表存储结构模拟此过程，按照出列的顺序印出各人的编号。

**[测试数据]**

m的初值为20；密码：3，1，7，2，4，8，4（正确的结果应为6，1，4，7，2，3，5）。

**[实现提示]**

程序运行后首先要求用户指定初始报数上限值，然后读取各人的密码。设n≤30。

**46．merkle树**

构建merkle树的数据结构，并实现基于树的插入和删除。

47． MPT树

Merkle Patricia Tree(简称MPT树,实际上是一种trie前缀树)，请构建mpt树的数据结构，并实现基于树的插入和删除。

48．chord环

用于p2p网络中节点的组织及数据的访问，实现节点的加入和退出功能，以及数据的查找功能。

**49．** 您在一家大型科技公司工作，负责定义系统管理员的日程安排，这些管理员将在接下来的 30 个晚上值夜班（编号为0, 1, . . . , 29)。有一组 n 个系统管理员（编号为 0、1、...、n-1）可供使用

一些夜班并获得更高的时薪。但是每个系统管理员对他们想要工作的夜晚有自己的偏好（以及他们不喜欢的夜晚）想工作）。您将获得一个列表首选项列表作为输入。每个内部列表代表不同的一天。如果编号为 j 的系统管理员有兴趣在编号为 i 的夜晚的班次，preferences[i][j] 否则等于 0。贵公司制定了以下政策：

• 有一个整数参数 sysadmins\_per\_night 指定确切的每晚应该值班的系统管理员（相同数量的系统管理员将每晚都需要）。它认为 0 ≤ sysadmins\_per\_night ≤ n。

• 有一个整数参数 max\_unwanted\_shifts，n 个系统管理员中的每一个都应该最多分配给他/她不感兴趣的 max\_unwanted\_shifts 个夜班。它认为 0 ≤ max\_unwanted\_shifts ≤ 30。

• 有一个整数参数min\_shifts，n个系统管理员中的每一个都应该分配到至少min\_shifts 个夜班。它认为 0 ≤ min\_shifts ≤ 30。

• 只要您遵守上述三个限制条件，您就可以自由分配一个系统管理员可以根据需要进行尽可能多的夜班。

为了解决这个问题，你应该写一个函数：

allocate(preferences, sysadmins\_per\_night,max\_unwanted\_shifts, min\_shifts) 返回：

• 无（即 Python NoneType），如果不存在满足所有约束的分配。

• 否则，它返回列表分配列表。分配[i][j] 应该等于1 如果编号为 j 的系统管理员被分配在编号为 i 的夜晚工作，并且 如果未将编号为 j 的系统管理员分配给在编号为 i 的晚上工作。

**基本要求：**

解决方案的最坏情况时间复杂度应为 O （n2）。

**50.贪心聚类算法**

1）使用经典的无权网络：Karate网络（34个节点、78个边）、Football网络（115个节点、613个边）、Polbooks网络（105个节点、441个边）和Dolphins网络（62个节点、159个边），验证贪心聚类算法

2）使用经典的有权网络：Wine网络（178个节点、31684个边）和Butterfly网络（818个节点、669124个边），使用ACC和NMI两个指标评价算法的优劣

**51.谱聚类算法**

1）使用经典的无权网络：Karate网络（34个节点、78个边）、Football网络（115个节点、613个边）、Polbooks网络（105个节点、441个边）和Dolphins网络（62个节点、159个边），验证谱聚类算法

2）使用经典的有权网络：Wine网络（178个节点、31684个边）和Butterfly网络（818个节点、669124个边），使用ACC和NMI两个指标评价算法的优劣

52. 0/1背包问题动态规划求解算法思想的可视化演示

基本要求

物品数据随机产生，针对随机案例，对0/1背包问题动态规划求解提供执行过程的动态图形演示。

53. 所有点对之间的最短路径问题动态规划求解算法思想的可视化演示

基本要求

加权有向图随机产生，针对随机案例，对所有点对之间的最短路径问题动态规划求解提供执行过程的动态图形演示。

53. 爬楼梯问题动态规划求解算法思想的可视化演示

基本要求

楼层数随机产生，每次爬的楼梯数为1层或者2层，针对随机案例，对所有可能的组合动态规划求解提供执行过程的动态图形演示。

54. 求组合问题动态规划求解算法思想的可视化演示

基本要求

m，n随机产生，针对随机案例，对所有可能的组合动态规划求解提供执行过程的动态图形演示。

