

2021-2022(1) 数据结构课程设计

总体要求：

- (1) 每人独立一题，题目可相同，但代码不得重复。
- (2) 每个学生必须充分分析设计要求，给出解决方案，建立必要的数据结构，然后设计总体流程、详细设计必要的算法，并最终显示结果。
- (3) 每人提交一份课程设计报告和能够运行通过的完整程序（二者缺一不可，并且其中有一项不及格，则总成绩视为不及格）。
- (4) 编程语言只能是 C 或 C++，并且代码中采用的数据结构、数据类型的定义、存储结构和操作算法都必须是本学期所学的，不可以使用面向对象编程技术，否则都视为成绩无效。

纪律要求

杜绝报告和代码抄袭。如若发现有抄袭现象，抄袭者和被抄袭者的课程设计成绩全部取消。

参考题目

从以下题目中选择一个或自拟题目（若自拟题目，工作量和难度要与参考题目相当），完成问题算法的分析、设计和实现。

1、通讯录的制作

用双向链表作数据结构，编写一个通讯录管理系统。

每条信息至包含：姓名（NAME）街道（STREET）城市（CITY）邮编（EIP）国家（STATE）几项，且应完成一下几方面的功能：

- (1) 输入信息——enter();
- (2) 显示信息——display();
- (3) 查找以姓名作为关键字 ——search();
- (4) 删除信息——delete();
- (5) 存盘——save ();
- (6) 装入——load();

2、仓库管理系统（线性表应用）

采用双向链表的存储结构建立一个仓库管理程序，可以按顺序和货物名称查询仓库存储情况，也可以增加或删除货物信息、实现货物的入库出库，要求能用文件保存仓库货品信息。

3、敢死队问题

有 M 个敢死队员要炸掉敌人的一碉堡，谁都不想去，排长决定用轮回数数的办法来决定哪个战士去执行任务。如果前一个战士没完成任务，则要再派一个战士上去。现给每个战士编一个号，大家围坐成一圈，随便从某一个战士开始计数，当数到 5 时，对应的战士就去执行任务，且此战士不再参加下一轮计数。如果此战士没完成任务，再从下一个战士开始数数，被数到第 5 时，此战士接着去执行任务。以此类推，直到任务完成为止。

排长是不愿意去的，假设排长为 1 号，请设计程序，求出从第几号战士开始计数才能让排长最后一个留下来而不去执行任务（至少采用两种不同的数据结构的方法实现）。

4、停车场管理

设停车场是一个可停放 n 辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时间的先后顺序依次排列。若场内已停满 n 辆车，则后来的汽车只能在门外的便道上等候，一旦有车开走，则排在便道上的第一辆车即可开入。当停车场内某辆车要离开时，在它之后进入的车辆必须退出车场为它让路，待该车开出大门外，其他车辆再按原次序进入车场，每辆停放在车场的车在它离开车场时必须按它停留的时间长短交纳费用。试为停车场编制按上述要求进行管理的模拟程序。

以栈模拟停车场，以队列模拟场外的车道，按照从终端读入的输入数据序列进行模拟管理。每一组输入数据包括三个数据项：汽车到达或离去的信息、汽车牌照号码以及到达或离去的时刻。

对每一组输入数据进行操作后的输出信息为：若车辆到达，则输出汽车在停车场的内或便道上的位置，若车辆离去，则输出汽车在停车场内停留的时间和应交纳的引用。

5、车厢调度问题

假设停在铁路调度站(如教科书中图 3.1(b)所示)入口处的车厢系列的编号依次为 1, 2, 3, ..., n 。要求：

- (1) 设计一个程序，求出由一个编号依次为 1, 2, ..., n 的车厢序列可能产生的所有出栈系列。
- (2) 利用双向栈存储结构实现调度站和输出序列这两个栈的空间共享。
- (3) 对于每个输出序列演示出所有操作序列的变化过程。

6、求解迷宫问题

以一个 $m \times n$ 的矩阵表示迷宫，0 和 1 分别表示迷宫中的通路和障碍，设计程序，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。要求：

- (1) 以栈作为存储结构，编写一个求解迷宫的非递归程序，并将求得的通路以三元组 (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。
- (4) 输出堆栈的变化过程及探测的过程

7、HTML 文档标记匹配算法

输入一段 HTML 代码，判断该代码是否符合 HTML 的语法。HTML 文档由不同的标记划分为不同的部分与层次。与括号类似，这些标记需要成对出现，对于名为 `<myTag>` 的起始标记，相应的结束标记为 `</myTag>`。常用的 HTML 标记：

```
<html> </html>: HTML 文档
<title> </title>: 文档标题
<body> </body>: 文档体
<h1> </h1>: 节的头部
<center> </center>: 居中对齐
<left> </left>: 左对齐
<p> </p>: 段落
...
```

8、双层停车场管理

有一个两层的停车场，每层有 6 个车位，当第一层车停满后才允许使用第二层。（停车场可用一个二维数组实现，每个数组元素存放一个车牌号）每辆车的信息包括车牌号，层

号，车位号，停车时间共 4 项。其中停车时间按分钟计算。 要求：

(1) 假设停车场初始状态为第一层已经停有 4 辆车，其车位号依次为 1—4，停车时间依次为 20，15，10，5。即先将这四辆车的信息存入文件“car.dat”中（数组的对应元素也要进行赋值）

(2) 停车操作：当一辆车进入停车场时，先输入其车牌号，再为它分配一个层号和一个车位号（分配前先查询车位的使用情况，如果第一层有空则必须停在第一层），停车时间设为 5，最后将新停入的汽车的信息添加文件“car.dat”中，并将在此之前的所有车的停车时间加 5。

(2) 收费管理（取车）：当有车离开时，输入其车牌号，先按其停车时间计算费用，每 5 分钟 0.2 元。（停车费用可设置一个变量进行保存），同时从文件“car.dat”中删除该车的信息，并将该车对应的车位设置为可使用状态（即二维数组对应元素清零）。按用户的选择来判断是否要输出停车收费的总计。

(4) 输出停车场中全部车辆的信息。

(5) 退出系统。

9、模式匹配算法的应用

文学研究人员需要统计某篇英文小说中某些形容词的出现次数和位置。试写一个实现这一目标的文字统计程序

(1) 英文小说存于一个文本文件中。待统计的词汇集合要一次输入完毕，即统计工作必须在程序的一次运行之后就全部完成。程序的输出结果是每个词的出现次数和出现位置所在的行的行号，格式自行设计。待统计的“单词”在文本串中不跨行出现，它或者从行首开始，或者前置以一个空格符。

(2) 模式匹配基于 KMP 算法。

10、一元多项式计算器

设有一元多项式 $A_m(x)$ 和 $B_n(x)$ 。

$$A_m(x) = A_0 + A_1x^1 + A_2x^2 + A_3x^3 + \dots + A_mx^m$$

$$B_n(x) = B_0 + B_1x^1 + B_2x^2 + B_3x^3 + \dots + B_nx^n$$

试求 $M(x) = A_m(x) + B_n(x)$ 、 $M(x) = A_m(x) - B_n(x)$ 和 $M(x) = A_m(x) \times B_n(x)$ 。

(1) 首先判定多项式是否稀疏；

(2) 分别采用顺序和链式结构实现；

(3) 结果 $M(x)$ 中无重复阶项和无零系数项；

(4) 要求输出结果的升幂和降幂两种排列情况。

11、家谱管理

设计并实现一个简单的家谱管理系统。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、删除功能。

(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) 按某种顺序输出家谱信息（树的遍历操作）、以树型结构输出家谱资料等功能。

12、哈夫曼编码/译码器

准备一个字符文件，要求：

(1) 统计该文件中各种字符的频率；

(2) 对各字符进行 Huffman 编码，显示每个字符的编码；

(3) 以及将该文件翻译成 Huffman 编码文件；

(4) 再将 Huffman 编码文件翻译成源文件；

(5) 显示每个字符以一个字节进行二进制编码后的编码文件；

13、图的遍历和生成树求解

- (1) 先任意创建一个图；
- (2) 图的 DFS, BFS 的递归和非递归算法的实现
- (3) 最小生成树（两个算法）的实现，求连通分量的实现
- (4) 要求用邻接矩阵、邻接表、十字链表多种结构存储实现

14、校园导航问题

设计一个校园导航程序，为来访的客人提供各种信息查询服务。要求：

- (1) 设计校园平面图，所含景点不少于 6 个。以图中顶点表示校内各景点，存放景点名称、代号、简介等信息；以边表示路径，存放路径长度等相关信息。
- (2) 为来访客人提供图中任意景点相关信息的查询。
- (3) 为来访客人提供图中任意景点的问路查询，即查询任意两个景点之间的一条最短的简单路径。

选做内容：

- (1) 提供途中任意景点问路查询，即求任意两个景点间的所有路径。
- (2) 提供校园图中多个景点的最佳访问路线查询，即求途经这多个景点的最佳（短）路径。

15、地图着色问题

设计地图着色软件，对江苏省地图中的地级市进行着色，要求相邻地级市所使用的颜色不同，并保证使用的颜色最少。要求：

- (1) 地图采用图型数据结构，每个地级市为一个节点，边表示对应的两个地级市相邻。
- (2) 设计着色算法，保证邻接点不是同一种颜色。
- (3) 演示程序以用户和计算机的对话方式进行。

16、最短旅程的求解

有 n 个城市（编号从 1 到 n ），它们之间通过双向的道路相连。那里只有 $n-1$ 条道路，但是，它们的连接方式使得从任意城市都可以走到其他的任何城市。一天，某个游客到了编号为 k 的城市。他计划从城市 k 开始，游遍所有的城市 $m_1, m_2, m_3, \dots, m_i, \dots$ （不一定要按这个顺序旅游）。每个城市 m_i 都是不同的，并且，也与 k 不同。他想要以最短的路程旅行完所有的城市（从城市 k 开始）。

请计算旅游完上述的城市最短需要多少路程。

17、关键路径问题

设计一个程序，求出完成整项工程至少需要多少时间，以及整项工程中的关键活动。

- (1) 对一个描述工程的 AOE 网，应判断其是否能够顺利进行。
- (2) 若该工程能顺利进行，输出完成整项工程至少需要多少时间，以及每一个关键活动所依附的两个顶点、最早发生时间、最迟发生时间。

18、软件工程进度规划

设计一个软件，需要进行用户需求分析、系统需求确认、系统概要设计、设计用例场景、系统的详细设计、数据库详细设计、编码、单元测试、集成测试、系统测试、维护等活动。用户需求分析需要在系统需求确认之前完成，系统的详细设计必须在系统的概要设计、设计系统用例和设计用例场景之前完成。如表所示，是一系列活动之间的关系。

表 系统活动之间的关系

活动代码	活动名称	先需活动
A1	用户需求分析	无
A2	系统需求确认	A1
A3	系统概要设计	A2
A4	设计用例场景	无
A5	系统的详细设计	A3, A4
A6	数据库详细设计	A3
A7	编码	A5, A6
A8	单元测试	A7
A9	集成测试	A8
A10	系统测试	A7
A11	维护	A11

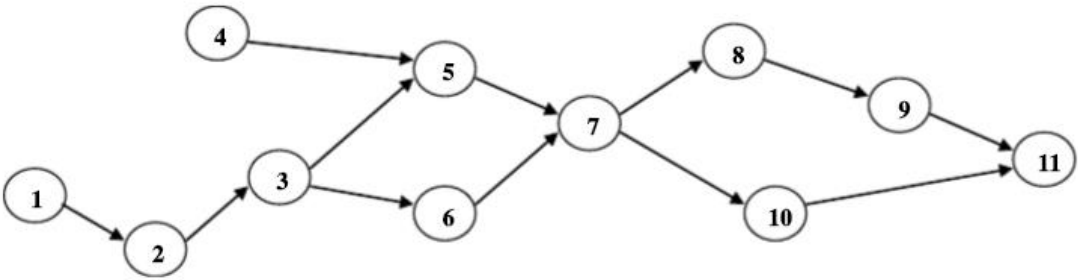


图 软件设计流程的 AOV 网

请设计算法判断该软件设计流程是否有回路,若无请给出该软件设计 AOV 网的拓朴序列。

19、教学计划编制问题

大学的每个专业都要制定教学计划。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，而且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程是确定的，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。

(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；二是使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。

先修关系如下：

课程编号	课程名称	先决条件
C1	程序设计基础	无
C2	离散数学	C1

C3	数据结构	C1, C2
C4	汇编语言	C1
C5	语言的设计和分析	C3, C4
C6	计算机原理	C11
C7	编译原理	C5, C3
C8	操作系统	C3, C6
C9	高等数学	无
C10	线性代数	C9
C11	普通物理	C9
C12	数值分析	C9, C10, C1

实现提示:

可设学期总数不超过 12, 课程总数不超过 100。如果输入的先修课程号不在该专业开设的课程序列中, 则作为错误处理。应建立内部课程号与课程号之间的对应关系。

20、排序综合

利用随机函数产生 N 个随机整数 (20000 以上), 对这些数进行多种方法进行排序。

要求:

- (1) 至少采用三种方法实现上述问题求解 (提示, 可采用的方法有插入排序、希尔排序、起泡排序、快速排序、选择排序、堆排序、归并排序)。并把排序后的结果保存在不同的文件中。
- (2) 统计每一种排序方法的性能 (以上机运行程序所花费的时间为准进行对比), 找出其中两种较快的方法。
- (3) 如果采用 4 种或 4 种以上的方法者, 可适当加分。

21、二叉排序树的实现

用顺序和二叉链表作存储结构

- (1) 以回车 ('\n') 为输入结束标志, 输入数列 L, 生成一棵二叉排序树 T;
- (2) 对二叉排序树 T 作中序遍历, 输出结果;
- (3) 输入元素 x, 查找二叉排序树 T, 若存在含 x 的结点, 则删除该结点, 并作中序遍历 (执行操作 2); 否则输出信息 “无 x”。

22、散列表的设计与实现

设计散列表实现电话号码查找系统。

- (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) 采用一定的方法解决冲突;
- (4) 查找并显示给定电话号码的记录;
- (5) 查找并显示给定用户名的记录。

进一步完成内容:

- (1) 系统功能的完善;
- (2) 设计不同的散列函数, 比较冲突率;

(3) 在散列函数确定的前提下, 尝试各种不同类型处理冲突的方法, 考察平均查找长度的变化。

23、哈希表的设计与实现——线性探测再散列

设计哈希表实现电话号码查找系统。

- (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不同的哈希表;
- (3) 采用线性探测再散列的方法解决冲突;
- (4) 查找并显示给定电话号码的记录;
- (5) 查找并显示给定用户名的记录。

24、哈希表的设计与实现——二次探测再散列

设计哈希表实现电话号码查找系统。

- (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不同的哈希表;
- (3) 采用二次探测再散列的方法解决冲突;
- (4) 查找并显示给定电话号码的记录;
- (5) 查找并显示给定用户名的记录。

25、哈希表的设计与实现——链地址法

设计哈希表实现电话号码查找系统。

- (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不同的哈希表;
- (3) 采用链地址法解决冲突;
- (4) 查找并显示给定电话号码的记录;
- (5) 查找并显示给定用户名的记录。

26、小型书店的图书信息管理模拟

小型书店的图书信息管理的功能包括图书信息的增加、删除、更新和查询等。设计一个小型书店的图书信息管理模拟程序, 以使上述业务可以借助计算机来完成。

每一类图书所涉及的信息有: 图书的 ISBN 号、图书名称、作者、出版社、价格、库存数量。能实现的操作和功能如下:

(1) 查询图书信息: 根据输入的图书信息 (ISBN 号或名称) 输出下列信息: 图书的 ISBN 号、图书名称、作者、出版社、价格、库存数量。

(2) 图书信息增加或更新: 根据每次采购图书的信息要求实现对相应图书信息的增加或更新。

(3) 图书信息删除: 对于部分图书的信息需要进行删除操作。

实现提示:

图书信息可以文件方式存放, 便于数据操作。

27、简单的职工管理系统

设计简单的职工信息管理程序。对单位的职工进行管理, 包括插入、删除、查找、排序等功能。职工对象包括姓名、性别、出生年月、工作年月、学历、职务、住址、电话等信息。

- (1) 新增一名职工: 将新增职工对象按姓名以字典方式职工管理文件中。
- (2) 删除一名职工: 从职工管理文件中删除一名职工对象。
- (3) 查询: 从职工管理文件中查询符合某些条件的职工。
- (4) 修改: 检索某个职工对象, 对其某些属性进行修改。

(5) 排序：按某种需要对职工对象文件进行排序。

实现提示：

职工对象数不必很多，便于一次读入内存，所有操作不经过内外存交换。

(1) 由键盘输入职工对象，以文件方式保存。程序执行时先将文件读入内存。

(2) 对职工对象中的“姓名”按字典顺序进行排序。

(3) 对排序后的职工对象进行增、删、查询、修改、排序等操作。

选做内容：

将职工对象按散列法存储，并设计解决冲突的方法。在此基础上实现增、删、查询、修改、排序等操作。