

《数据结构与算法》 实验讲义

黑龙江大学计算机科学技术学院、软件学院

2020 版

目录

第一部分：总体说明

第二部分：《数据结构与算法》实验内容及要求

- 实验一 顺序存储的线性表（2 学时）
- 实验二 单链表上的操作（2 学时）
- 实验三 循环链表和双链表（2 学时）
- 实验四 栈和队列（2 学时）
- 实验五 树的应用（2 学时）
- 实验六 综合设计与应用一(个人项目)（6 学时）

以下题目任选其中一个：

1. 表达式求值问题
2. 哈夫曼编码译码器

实验七 图的应用（2 学时）

实验八 综合设计与应用二（团队项目）（4 学时）

以下题目任选其中一个：

1. 校园导游系统（求最短路径算法: Dijkstra, Floyd)
2. 最小生成树模拟程序(求最小生成树算法: Prim、Kruskal)
3. 迷宫问题模拟程序（栈、队列、堆的应用，各种树形搜索算法）
4. 通讯录管理系统的设计与实现（各种查找算法）
5. 内部排序性能分析演示程序（各种排序算法）
6. 教学计划编制系统的设计与实现（拓扑排序）
7. 自选题目

实验九 排序和查找（2 学时）

第三部分：选用教材及参考书

第四部分：设计报告模板

附录 1：设计报告封面

附录 2：编程规范

第一部分：总体说明

《数据结构与算法》实验讲义是《数据结构与算法》实验课程的教学指导用书，该实验课程共 24 学时，由 9 个实验单元组成，每个实验单元的内容和要求在本讲义第二部分中给出。该实验成绩占《数据结构与算法》课程总成绩的 30%，实验课成绩构成如表 1 所示。

每缺勤一次在总成绩中扣 3 分，每迟到或早退一次在总成绩中扣 1 分。

表 1 实验成绩构成表

项目号	实验项目名称	学时	构成及考核标准	分值
实验一 (计：第 3 周 软：第 7 周)	顺序存储的线性表	2	共 3 个小实验题目，要求至少完成 2 个题目。 即按时保质完成每个小实验项目可得 1 分。	2
实验二 (计：第 4 周 软：第 8 周)	单链表上的操作	2	共 3 个小实验题目，要求至完成 2 个题目。 即按时保质完成每个小实验项目可得 1 分。	2
实验三 (计：第 5 周 软：第 9 周)	循环链表与双链表	2	共 3 个小实验题目，要求至少完成 2 个题目。 即按时保质完成每个小实验项目可得 1 分。	3
实验四 (计：第 6 周 软：第 10 周)	栈和队列	2	共 3 个小实验题目，要求至少完成 2 个题目。 即按时保质完成每个小实验项目可得 1 分。	2
实验五 (计：第 7 周 软：第 11 周)	树的应用	2	共 3 个小实验题目，要求至少完成 2 个题目。 即按时保质完成每个小实验项目可得 1 分。	3
实验六 (计：第 8-10 周 软：第 12-14 周)	综合设计与应用一（个人）	6	从给定选项中任选 1 个题目。 其中，系统实现占 5 分，设计报告占 3 分。	8
实验七 (计：第 11 周 软：第 15 周)	图的应用	2	共有 2 个小实验题目，要求至少选 1 个题目。 即按时保质完成每个小实验项目可得 2 分。	3
实验八 (计：第 12-13 周 软：第 16-17 周)	综合设计与应用二（团队）	4	从给定选项中任选 1 个题目或自选题目。 考核标准已在每个题目后面给出。	5
实验九 (计：第 14 周 软：第 18 周)	排序和查找	2	共 2 个小实验题目，按时保质完成每个小实验项目可得 1 分。	2
实验总成绩				30

第二部分：《数据结构与算法》实验内容及要求

实验一 顺序存储的线性表（2 学时）

一、实验目的

- 1、掌握线性表的逻辑结构特征。
- 2、熟练掌握线性表的顺序存储结构的描述方法。
- 3、熟练掌握顺序表上各种基本操作的实现。

二、实验内容

1、设线性表的数据元素都为整数，存放在顺序表 S 中且递增有序。设计算法，将 x 插入到顺序表 S 的适当位置上，以保持线性表的有序性。要求算法的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

2、线性表使用顺序表作存储结构，设计算法，仅用一个辅助结点，实现将顺序表中的结点循环右移 k 位的运算。

3、设计算法，仅用一个辅助结点，实现将顺序存储的线性表中的数据元素逆置。

三、实验指导

- 1、本实验所有题目中的向量可以用顺序表来描述，数据元素为整数。
- 2、提示

（1）题目 1 中的顺序表在建立时即是有序表， x 的值由用户输入，应该考虑 3 种情况，①小于原表中第一个元素的值；②介于最小元素值和最大元素值之间；③大于最大元素的值。

（2）对于题目 2， k 的值由用户输入，如顺序表中元素为 1,2,3,4,5,6，当 k 的值为 4 时，结果为 3,4,5,6,1,2。

四、考核标准

- 1、至少完成 2 个题目，设计合理，结果正确，按时完成；评定为 A。
- 2、完成部分题目或未按时完成，设计比较合理，结果正确；根据实际情况评定为 B 或 C。
- 3、未独立完成实验要求；评定为 D。

（本单元共 2 分。未按时完成或算法的时间与空间复杂度过高也会酌情扣分。）

实验二 单链表（2 学时）

一、实验目的

- 1、掌握线性表的链式存储结构。
- 2、熟练掌握单链表的描述方法和建立方法。
- 3、熟练掌握单链表上基本操作及其相关应用的实现。

二、实验内容

1、已知带头结点的动态单链表 L 中的结点是按整数值递增排序的，试写一算法将值为 x 的结点插入到表 L 中，使 L 仍然有序。要求算法的时间复杂度为 $O(n)$ ，空间复杂度为 $O(1)$ 。

2、设计一算法，逆置带头结点的动态链表 L 。要求利用原表的结点空间，并要求用尽可能少的时间完成。

3、假设有两个按元素值递增有序的线性表 A 和 B，均以单链表作存储结构，试编写算法将 A 表和 B 表归并成一个按元素值递减有序的线性表 C，并要求利用原表的空间存放 C，并要求用尽可能少的时间完成。

三、实验指导

1、本实验中的动态链表都用单链表实现，结点中 data 域类型假定为 int。

2、提示

(1) 题目 1 中的单链表在建立时即为有序表，x 的值由用户输入，应该考虑 3 种情况，

①小于原表中第一个元素的值；②介于最小元素值和最大元素值之间；③大于最大元素的值。

(2) 题目 2 可以考虑用头插法思想将单链表中每个结点“卸”下来再“挂”上去。

(3) 题目 3 也可以考虑使用头插法思想，同时可参照两个递增有序单链表合成一个有序单链表的算法。

四、考核标准

1、至少完成 2 个题目，设计合理，结果正确；评定为 A。

2、完成部分题目或未按时完成，设计比较合理，结果正确；评定为 B 或 C。

3、未独立完成实验要求；评定为 D。

(本单元共 2 分。未按时完成或算法的时间与空间复杂度过高也会酌情扣分。)

实验三 循环链表和双链表 (2 学时)

一、实验目的

1、熟练掌握循环链表的存储特征和建立方法。

2、熟练掌握双链表的存储特征和建立方法。

3、掌握在循环链表以及双链表上基本应用的实现。

二、实验内容

1、假设在长度大于 1 的单循环链表中，既无头结点也无头指针。s 为指向某个结点的指针，试编写算法删除结点*s 的直接前驱结点。

2、已知由单链表表示的线性表中，含有三类字符的数据元素（如：字母、数字和其它字符），设计算法构造三个以循环链表示的线性表，使每一个表中只含同一类的字符，且利用原表中的结点空间作为这三个表的空间。（头结点可以另辟空间）

3、有一双链表，每个结点中除有 prior、data 和 next 域外，还有一访问频度域 freq，在链表被启用前，其值均初始化为零。每当在链表上进行一次 LOCATE(L,x)运算，元素值为 x 的结点中 freq 域的值增 1，并使此链表中结点保持按 freq 递减的顺序排列，以便使频繁访问的结点总是靠近表头。设计满足上述要求的 LOCATE 算法。

三、实验指导

1、认真阅读题目，根据题目内容要求建立好相应的链表。

2、提示

(1) ①题目 1 应该先建立好单循环链表，并返回主调函数该单循环链表的头指针或尾指针；②用户输入某个整数，在建好的单循环链表中查找该整数，如找到将该结点的地址传给 s；③根据 s 删除其前驱结点，特别要注意删除的就是第一个结点的前驱；④参考测试用例：如链表中为 (1, 2, 3, 4, 5)，用户输入 3，结果为 (1, 2, 4, 5)；如链表中为 (1, 2, 3, 4, 5)，用户输入 1，则结果应为 (1, 2, 3, 4)。

(2) ①题目 2 应事先建立好一个非循环单链表 L (该链表结点的 data 域类型为 char), 该链表中含三种字符, 例如该链表可为如下:

('1' -> 'c' -> '!' -> 'h' -> '\$' -> '2' -> '3' -> 'i' -> 'n' -> '#' -> 'a')

②建立三个空的循环单链表 h1、h2、h3, 分别存储数字字符、字母字符、其它字符; ③按顺序扫描 L 中的每个结点, 根据结点中 data 的特征将其链入相应的循环链表中; ③按照上述测试用例, h1 中应为 ('1' -> '2' -> '3'), h2 中应为 ('c' -> 'h' -> 'i' -> 'n' -> 'a'), h3 中应为 ('!' -> '\$' -> '#')。

(3) 题目 3 要求先建立好双链表 DL1, DL1 中内容假设为 (1,2,3,4,5,6), 每个结点的 freq 值都为 0; 用户每输入一个值, DL1 应该做出相应的调整, 并把调整结果供用户直观查看。

四、考核标准

- 1、至少完成 2 个题目, 设计合理, 结果正确; 评定为 A
- 2、完成部分题目或未按时完成, 设计比较合理, 结果正确; 评定为 B 或 C
- 3、未独立完成实验要求; 评定为 D

(本单元共 3 分。只完成前两个题目最多得 2 分, 高质量完成 3 个题目给 3 分, 未按时完成或算法的时间与空间复杂度过高也会酌情扣分。)

实验四 栈和队列 (2 学时)

一、实验目的

- 1、掌握栈和队列的逻辑特征。
- 2、熟练掌握在两种存储结构上实现栈和队列的基本运算。
- 3、学会利用栈和队列解决一些实际问题。

二、实验内容

1、设单链表中存放着 n 个字符, 设计算法, 判断该字符串中是否有中心对称关系。例如: xyzyx、xyzyx 都算是中心对称的字符串。

2、设计算法判断一个算术表达式的圆括号是否配对。(提示: 对表达式进行扫描, 遇 '(' 进栈, 遇 ')' 退掉栈顶的 '(', 表达式被扫描完毕, 栈为空)

3、假设以带头结点的循环链表表示队列, 并只设一个指针指向队尾, 编写相应的置队空、入队和出队算法。

三、实验指导

1、题目 1 可用顺序栈或链栈实现; 题目 2 可以用顺序栈实现; 题目 3 用单循环链表实现。

2、提示

(1) 题目 1 重点考虑①哪些字符进栈; ②如何利用栈判断中心对称关系。

(2) 题目 2 重点考虑①遇到左括号如何处理, 遇到右括号如何处理, 遇到其它字符如何处理, 何时结束? ②当栈空时如果扫描到右括号时应如何处理?

(3) 题目 3 除了完成基本运算还要考虑设计直观的界面使用户清晰操作并直观看到操作结果。

四、考核标准

- 1、至少完成 2 个题目, 设计合理, 结果正确, 时间与空间复杂度低; 评定为 A
- 2、完成部分题目或未按时完成, 设计比较合理, 结果正确; 评定为 B 或 C
- 3、未独立完成实验要求; 评定为 D

(本单元共 2 分。未按时完成或算法的时间与空间复杂度过高也会酌情扣分。)

实验五 树的应用（2 学时）

一、实验目的

- 1、掌握二叉树的各种存储结构的特点及适用范围。
- 2、掌握建立二叉树的方法。（包括顺序存储、二叉链表存储）
- 3、熟练掌握二叉树的前序、中序、后序遍历的递归及非递归算法；灵活运用递归的遍历算法实现二叉树的应用。

二、实验内容

- 1、以二叉链表作存储结构，设计求二叉树高度的算法。
- 2、一棵 n 个结点的完全二叉树用向量作存储结构，用非递归算法实现对该二叉树进行前序遍历。
- 3、以二叉链表作存储结构，编写非递归的前序、中序、后序遍历算法。

三、实验指导

- 1、题目 1 和题目 3 要求事先建立好二叉树（以二叉链表作存储结构）；题目 2 要求事先建立好二叉树（采用顺序存储结构）。
- 2、提示
 - （1）题目 1 可参考二叉树的递归遍历算法进行设计。
 - （2）题目 2 考虑是否可以借助其他辅助结构进行设计。
 - （3）题目 3 可以考虑利用栈来实现，难点在于后序遍历的非递归算法的实现（何时入栈，如何标识何时出栈）。

四、考核标准

- 1、至少完成 2 个题目，设计合理，结果正确；评定为 A
 - 2、完成部分题目或未按时完成，设计比较合理，结果正确；评定为 B 或 C
 - 3、未独立完成实验要求；评定为 D
- （本单元共 3 分。只完成前两个题目最多得 2 分，高质量完成 3 个题目给 3 分，未按时完成或算法的时间与空间复杂度过高会酌情扣分。）

实验六 综合设计与应用一（个人项目）（6 学时）

本单元为个人综合项目，上机实现部分占 5 分，报告占 3 分，总计 8 分，共占 6 学时。

一、实验目的

1. 熟练掌握线性结构（线性表、栈、队列等）的存储方式，及在其上进行复杂操作的方式。
2. 熟练掌握非线性结构----树形结构的存储方式，及在其上进行复杂操作的方式。
3. 提高学生设计能力以及编写综合性大实验的编程能力。

二、实验内容（以下题目任选其中一个）

1. 表达式求值问题
2. 哈夫曼编码译码器

三、实验指导

1. 表达式求值问题

(1) 问题描述

表达式求值是程序设计语言编译中的一个最基本问题。人们在书写表达式时通常采用将运算符放在两个操作数中间的“中缀”表示形式，称为中缀表达式。但是这种表达式形式对计算机处理来说是不太适合的。在计算机领域，经常将算术表达式表示成“后缀”表示形式，称为后缀表达式。如：中缀表达式 $3+2*(7-5)$ 对应的后缀表达式为 $3275-*+$ 。

表达式是由操作数、运算符、界限符组成的。操作数既可以是常数，也可以是说明为变量或常量的标示符；运算符可以分为算术运算符、关系运算符和逻辑运算符 3 类；基本界限符有左右括号和表达式结束符等。

(2) 功能与界面要求

要求以字符序列的形式从终端输入或从文件输入语法正确的、不含变量的整数或小数表达式，并根据输入表达式实现：① 算数四则运算中缀表达式到后缀表达式的转换；② 后缀表达式的求值；③ 中缀表达式的求值。

要求演示在求值过程中运算符栈、操作数栈、输入字符和主要操作过程及运算结果。在本题目中可将表达式中的操作数规定为 1 位数字字符。也可根据个人的能力对这部分功能进行扩充，使得操作数可以是多位数甚至是小数。

算法应该能够过滤掉输入符号之间的空格。也可对本算法功能进行扩充，使其具有对输入表达式进行语法检查的功能。

为了简化问题，运算符可只包含 +、-、*、/ 四种基本运算，括号只有圆括号。可根据需要及个人能力对算法的功能进行扩充，允许有其它运算符。

本项目可使用控制台界面或可视化图形界面，若使用可视化图形界面有加分。要求配备菜单，至少含如下选项：

表达式求值

1. 中缀表达式到后缀表达式的转换
2. 后缀表达式的计算
3. 中缀表达式的计算
4. 退出

(3) 数据结构要求

要求输入表达式采用字符串存储，操作数和运算符采用栈存储，待输出的表达式用队列存储。

(4) 算法与数据结构设计指导

① 中缀表达式到后缀表达式的转换

● 问题分析

若一个中缀表达式中没有括号（如 $4+2*3$ ，它的后缀表达式为 $423*+$ ）。在扫描到中缀表达式中的 2 后，能立即输出 +，因为 * 具有较高优先级，必须先运算，因此需先保存 +。也就是说，新扫描运算符优先级必须与前一个运算符的优先级做比较，如果新的运算符优先级高，

就要像前一个运算符那样保存它，直到扫描到第二个操作数，将它输出后才能将该运算符输出。因此，在转化中必须保存两个运算符，后保存的运算符先输出。用计算机来实现这个转化过程，就需要用到能后进先出的数据结构----栈。

若在中缀表达式中含小括号，那么由于括号隔离了优先级规则，它在整个表达式的内部产生了完全独立的子表达式。因此，前面的算法就需要有所改变。当扫描到一个左括号时，需要将其压入栈中，使其在栈中产生一个“伪栈底”。这样算法就可以像前面一样进行。但当扫描到一个右括号时，需要将栈中从栈顶到“伪栈底”之间的所有运算符弹出，然后再将这个“伪栈底”删除。

● 数据结构与算法设计思想

顺序扫描中缀表达式，当读到数字时，直接将其送至**输出队列**中；当读到运算符时，将运算符**栈**中所有优先级高于或等于当前运算符的运算符弹出，送至输出队列中，再将当前运算符入栈；当读入左括号时，将其入运算符栈；当读到右括号时，将运算符栈中从栈顶到靠近栈顶的第一个左括号（“伪栈底”）之间的所有运算符全部依次弹出，送至输出队列中，再删除栈中的左括号。

为了简化算法，可认为扫描到的任何运算符，其优先级都比栈顶的**左括号**优先级高。为了方便边界条件（栈空）判断，提高算法运行效率，在扫描中缀表达式之前，在空栈中预先压入一个‘#’字符作为栈底元素，另外，在中缀表达式的最后增加一个‘#’字符作为中缀表达式的结束标志，当扫描到结束符‘#’时，将栈中从栈顶到‘#’之间的所有运算符全部依次弹出，送至输出队列中，再删除栈中的‘#’，并结束算法。

除了显示算法的运行结果外，最好能在算法运行过程中演示运算符栈和存放后缀表达式的队列的变化情况。如，若输入的中缀表达式字符串：1+2*（3-1+2）-3#，就会得到后缀表达式：1231-2+*+3-，在算法运行过程中，运算符栈和存放后缀表达式的队列变化过程如表 1 所示。

表 1 中缀表达式到后缀表达式的转换过程示例（数据为 1 位数）

转换步骤	中缀表达式的读入	运算符栈	后缀表达式
初始	1+2*（3-1+2）-3#	#	空
1	+2*（3-1+2）-3#	#	1
2	2*（3-1+2）-3#	#+	1
3	*（3-1+2）-3#	#+	12
4	（3-1+2）-3#	#+*	12
5	3-1+2）-3#	#+*（	12
6	-1+2）-3#	#+*（	123
7	1+2）-3#	#+*（-	123
8	+2）-3#	#+*（-	1231
9	2）-3 #	#+*（+	1231-
10	）-3 #	#+*（+	1231-2
11	-3 #	#+*	1231-2+
12	3 #	#-	1231-2+*+
13	#	#-	1231-2+*+3
14		空	1231-2+*+3-

②后缀表达式的计算

● 问题分析

在后缀表达式中，不仅不需要括号，而且还完全免除了运算符优先规则。后缀表达式只需要从左到右顺序计算。

后缀表达式中可能有多个运算符，如 $1231-2+*+3-$ ，因此必须像输入字符一样保存中间结果。在计算后缀表达式时，最后保存的值最先取出参与运算，所以需要用栈来存储操作数及中间结果。

因为在生成的后缀表达式队列中，存放的是字符序列，因此在算法中要有一个数字字符到数值的转换。

● 数据结构与算法设计思想

利用栈（操作数和运算结果栈）计算后缀表达式。顺序扫描后缀表达式，当读到数字时，将其送至栈中；当读到运算符 θ 时，将栈顶字符弹出，将其转换成对应的数值并赋给变量 y ，再将次栈顶字符弹出，将其转换成对应的数值，并赋给变量 x ，之后计算 $x\theta y$ ，将运算结果转换成对应的数字字符送入栈中。

本算法可以对任意输入的后缀表达式进行计算，也可以以前一个中缀表达式到后缀表达式转换算法的输出为输入，计算该后缀表达式。

除了显示算法的运行结果外，最好能在算法运行过程中演示（操作数和运算结果）栈的变化情况。以输入的后缀表达式为 $1231-2+*+3-$ 的情况为例，在算法运行过程中，操作数及运算结果栈变化过程如表 2 所示。

表 2 后缀表达式计算过程示例（数据为 1 位数）

计算步骤	后缀表达式的读入	操作数和运算结果栈
初始	1231-2+*+3-	空
1	231-2+*+3-	1
2	31-2+*+3-	12
3	1-2+*+3-	123
4	-2+*+3-	1231
5	2+*+3-	122
6	+*+3-	1222
7	*+3-	124
8	+3-	18
9	3-	9
10	-	93
11	空	6

③中缀表达式的计算

● 问题分析

要把一个表达式翻译成正确求值的机器指令序列，首先要能够正确解释表达式。要对算术表达式求值，首先要了解算术四则运算规则。即：先乘除，后加减；同级从左到右计算；先括号内，后括号外。

例如， $1+2*(3-1+2)-3=1+2*(2+2)-3=1+2*4-3=1+8-3=9-3=6$

为了简化算法，可认为扫描到的任何运算符，其优先级都比栈顶的左括号优先级高。算

符之间的优先关系如表 3 所示。

表 3 算符之间的优先关系表

	+	-	*	/	()	#
+	>	>	<	<	<	>	>
-	>	>	<	<	<	>	>
*	>	>	>	>	<	>	>
/	>	>	>	>	<	>	>
(<	<	<	<	<	=	x
)	>	>	>	>	x	>	>
#	<	<	<	<	<	x	=

● 数据结构与算法设计思想

算法伪代码：

将输入的算术表达式存储到字符串 S 中；
建立并初始化操作数栈和运算符栈，并将‘#’压入运算符栈；
建立二维数组存储算术运算符的优先关系；
建立并初始化一个临时栈 T 用来处理多位数或小数；
依次（循环）读入表达式中的每个字符，存到字符变量 ch 中；
当读入的字符 ch 是操作数，将其入临时栈；
当读入的字符 ch 是运算符：
 将临时栈中的数据转化为整数或小数（这需要编写个函数），
 并将转化后的数据入操作数栈，清空临时栈；
 根据运算符栈的栈顶元素和 ch 的优先权比较结果，做不同的处理：
 a) 若是小于，则 ch 压入运算符栈，读入下一个字符；
 b) 若是大于，则弹出运算符栈顶的运算符，从操作数栈弹出两个数，进行相应运算，将结果压入操作数栈；
 c) 若是等于，则运算符栈顶元素是‘(’且 ch 是‘)’, 这时弹出运算符栈顶的‘(’, 相当于去括号，然后读入下一个字符；
 d) 当读入的字符 ch 是‘#’, 且运算符栈顶元素也是‘#’时，
 算法结束。

除了显示最终运算结果外，最好能在算法运行过程中演示运算符栈和操作数栈的变化情况。

(4) 考核标准

若本实验项目占 8 分，其中：

- 基本功能占 3.6 分，具体分值分布如下：
 - 中缀表达式到后缀表达式的转换（1.2 分）
 - 后缀表达式的计算（1.2 分）
 - 中缀表达式的计算(1.2 分)
- 附加功能占 1.4 分，具体分值分部如下：
 - 具有语法检查功能（含滤空格、除零异常处理、括号匹配判断等功能）。（0.3 分）
 - 除了用键盘输入数据外，还可以将前一个运算的结果（以指定特定文件形式）作

为下一个运算的输入。(0.3 分)

- 能演示算法运行过程中运算符栈和操作数栈的变化情况。(0.2 分)
- 能处理多位数、负数及小数的运算。(0.2 分)
- 能实现加减乘除运算以外的其他运算。(0.2 分)
- 界面友好，使用方便，最好能有图形界面。(0.2 分)

- 设计报告占 3 分。

2. 哈夫曼编码译码器

(1) 问题描述

设计一个哈夫曼编码译码系统，对某个英文文本文件 (.txt) 中的字符进行哈夫曼编码，并将该 txt 文件生成编码文件 (.cod)；反过来还可将一个编码文件 (.cod) 还原为一个文本文件 (.txt)。

(2) 功能及界面要求

本题可采用 console 控制台或可视化界面，console 控制台参考界面如下：

```
*****
*                                     *
*               哈夫曼编码译码器               *
*      1、选择需要进行编码的文件      *
*      2、建立哈夫曼树                  *
*      3、建立密码本并对文件编码        *
*      4、选择需要进行解码的文件并解码  *
*      5、按位压缩方式对文件进行压缩    *
*                                     *
*****
```

功能说明：

- ① “选择需要进行编码的文件”：选择该选项后，提示用户输入（或选择）要进行编码（加密）的文件（包括路径和文件名）。
- ② “建立哈夫曼树”：选择该选项后，程序根据 1 中确定的文件建立哈夫曼树。
- ③ “建立密码本并对文件编码”：选择该选项后，程序根据 2 中建立好的哈夫曼树为 1 中出现的每个字符建立编码，并对文件进行编码，在进行编码前提示用户将编码文件存放在哪个文件（文件扩展名为 cod）中。
- ④ “选择需要进行解码的文件并解码”：选择该选项后，提示用户输入（或选择）需要进行解码（译码）的文件（文件扩展名为 cod），并输入（或选择）将解码（译码）后的文件存放到哪个文件（文件扩展名为 txt），程序将 cod 文件根据 3 建立的密码本进行解码，解码到 txt 文件中。
- ⑤ “按位压缩方式对文件进行压缩”：对 cod 文件进行压缩，显示压缩比（即压缩后的编码文件字节数/编码前的原 txt 文件字节数），并能对压缩后的 cod 文件进行解码。

(3) 存储要求

- 哈夫曼树采用数组存储。
- 密码本在内存中采用数组存储，也可根据用户选择将密码本存到文件中。
- 编码文件和译码文件都采用文本文件存储。

(4) 数据结构与算法设计指导

①为实现功能 2，首先用对原 txt 文件进行扫描，得到每个字符（包括空格、标点符号和回车换行）出现的次数，并根据教材提供的算法得到哈夫曼树。

②为实现功能 3，首先根据哈夫曼树及教材提供的算法得到每个出现字符的哈夫曼编码（即建立密码本），并对原 txt 文件重新进行扫描，扫描到某个字符时在密码本中找到该字符的哈夫曼编码，写入到编码文件中。

③为实现功能 4，要扫描编码文件，扫描（读）到‘0’或‘1’时，根据哈夫曼树进行相应的处理，直到扫描（读）到某个‘0’或‘1’后，哈夫曼树已经到达某个叶子，将该叶子对应的字符写入到解码文件中。

④为实现功能 5，对编码后的 cod 文件进行处理，将每 8 个（‘0’或‘1’）字符串转化为相应的整数（用 1 个字节存储）并写入压缩文件中，注意对最后一个 01 串（长度 ≤ 8 ）的处理。

(5) 考核标准

若本实验项目占 8 分，其中：

- 基本功能占 4 分，具体分值分布如下：

- 建立哈夫曼树。（1 分）
- 产生字符-哈夫曼编码对照表，并显示每个字符对应的哈夫曼编码。（1 分）
- 对文件或从键盘输入的字符串进行编码，产生发送方的编码电文并显示或存入指定的文件中。（1 分）
- 接收方译码，显示译码结果或将译码结果存入指定文件。（1 分）

- 附加功能占 1 分，具体分值分部如下：

- 发送方按位压缩存储电文，接收方根据压缩电文解压译码，并显示压缩比。（0.5 分）
- 界面友好，使用方便，最好能有可视化界面。（0.5 分）

- 设计报告占 3 分。

实验七 图的应用（2 学时）

一、实验目的

- 1、掌握图的各种存储结构的特点及适用范围。
- 2、掌握建立图的方法。（包括邻接矩阵、邻接表）
- 3、熟练掌握图的深度优先搜索算法和广度优先搜索算法，并能灵活运用这两个算法解决实际问题。

二、实验内容

1. 在图中求一条从顶点 i 到顶点 j 的简单路径。
2. 在图中求从顶点 i 到顶点 j 的最短路径。

三、实验指导

实现上述两个功能时要求图分别用邻接矩阵和邻接表表示。

求简单路径问题，可利用图得深度优先搜索遍历算法实现，从顶点 i 出发，开始遍历，访问到顶点 j 时遍历结束。在遍历的过程中，需要将访问过的顶点压入栈，当在遍历过程中遇到一个访问过的顶点 k 时，需要依次删除栈中元素，直到栈顶为 k。遍历结束后，将栈中的顶点倒着输出即为顶点 i 到顶点 j 的简单路径。

求最短路径问题,可利用图得广度优先搜索遍历算法实现,为实现图得广度优先搜索算法,需要用到队列。

四、考核标准

- 1、完成全部题目,设计合理,结果正确;评定为 A
- 2、完成部分题目或未按时完成,设计比较合理,结果正确;评定为 B 或 C
- 3、未独立完成实验要求;评定为 D

(本单元共 3 分。未按时完成或算法的时间与空间复杂度过高会酌情扣分。)

实验八 综合设计与应用二（团队项目）（4 学时）

本单元为团队项目,系统设计与实现占 3 分, PPT、答辩情况、团队合作、创新性、实用性占 2 分,总计 5 分。

一、实验目的

综合运用所学的数据结构与算法实现一个小系统。

二、实验内容

1. 校园导游系统(求最短路径算法: Dijkstra,Floyd)
2. 最小生成树模拟程序(求最小生成树算法: Prim、Kruskal)
3. 迷宫问题模拟程序（栈、队列、堆的应用,各种树形搜索算法）
4. 内部排序性能分析演示程序（各种排序算法）
5. 通讯录管理系统的设计与实现（各种查找算法）
6. （拓扑排序算法）
7. 自选题目

三、实验指导

1. 校园导游系统

（1）问题描述

设计一个校园导游程序,为来学校的用户提供景点信息及路径查询服务。

（2）功能及界面要求

本题可采用 console 控制台或可视化界面,console 控制台参考界面如下:

```
*****
*                                     *
*                               校园导游程序                               *
*                                     *
*                                     *
*          这段区域显示各顶点的编号、名称并画出校园地图                  *
*                                     *
*          1、景点信息查询                                                    *
*          2、校门口到其他景点的路径查询（单源最短路径）                    *
*          3、校园各景点间的路径查询（各顶点对间最短路径）                  *
*                                     *
*****
```

功能说明:

① “景点信息查询”

系统为来访客人提供图中任意景点相关信息的查询功能。

用户选择该选项后，提示用户输入要查询的景点编号，根据景点编号显示出该景点的名称和简介。

② “校门口到其他景点的路径查询”

系统为来访客人提供从校门口到图中任意景点的问路查询功能。

用户选择该选项后，提示用户输入要到达的景点编号，根据该景点编号显示出从校门口到该景点的最短路径长度及路径信息（即路径上的顶点序列）

③ “校园各景点间的路径查询”

系统为为来访客人提供图中任意景点间的问路查询；

用户选择该选项后，提示用户输入起点编号及终点编号，根据起点和终点编号确定起点到终点的最短路径及长度并直观显示。

（3）数据结构要求

- 校园地图中的景点信息采用一维结构体数组存储，各条边的信息采用邻接矩阵存储。
- 路径采用栈或队列存储。

（4）数据结构与算法设计指导

①设计学校的校园平面图（有向图，任意两点间的去边和回边长度相等），所含景点不少于 10 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利用教材上建立图的邻接矩阵的算法实现图的存储。

③根据查找算法实现功能 1。

④根据迪杰斯特拉算法和教材上的参考程序实现功能 2。

⑤根据弗洛伊德算法和教材上的参考程序实现功能 3。

（5）考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。

2. 最小生成树模拟程序

（1）问题描述

已知一个无向连通网表示 n 个城市以及城市间可能设置的通信线路，其中网的顶点表示城市，边表示两个城市之间的线路，赋予边上的权值表示相应的代价。对于 n 个点的连通网可以建立许多不同的生成树，每一棵生成树都可以是一个通信网。现在，我们要选择这样一棵生成树，使总的耗费最小，即构造连通网的最小生成树。

(2) 功能及界面要求

本题可采用 console 控制台或可视化界面，console 控制台参考界面如下：

```
*****
*                               *
*               最小生成树模拟程序                               *
*                               *
*               这段区域可以显示出一个设计好的无向连通网络                               *
*                               *
*                               *
*               1、普里姆（Prim）算法构造最小生成树                               *
*               2、克鲁斯卡尔（Kruskal）算法构造最小生成树                               *
*                               *
*****
```

功能说明：

- ① 提示用户输入起点编号，根据普里姆（Prim）算法构造出最小生成树，按顺序显示选出的各条边，把构造出的最小生成树直观显示出来，并显示出该最小生成树上各边权值之和。
- ② 根据克鲁斯卡尔（Kruskal）算法构造出最小生成树，按顺序显示选出的各条边，把构造出的最小生成树直观显示出来，并显示出该最小生成树上各边权值之和。

(3) 数据结构要求

- 无向连通图采用邻接矩阵存储。
- 最小生成树采用一维结构体数组存储（边）。
- 实现 Kruskal 算法时，最好用并查集实现，并用树形结构存储并查集。

(4) 数据结构与算法设计指导

- ① 设计无向连通图（顶点数不小于 6）并存储。
- ③ 根据邻接矩阵和教材中相应的算法实现功能 1。
- ③ 根据邻接矩阵和教材中相应的算法实现功能 2。

(5) 考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。

3. 迷宫问题模拟程序

(1) 问题描述

老鼠走迷宫是心理学中的一个经典试验。设有一只无盖大箱，箱中设置一些隔断，形成弯弯曲曲的通道作为迷宫，如图 1。箱子还设有一个入口和一个出口。试验时，先在出口处放一些奶酪之类的东西吸引老鼠，然后将一只老鼠放到入口处，这样，老鼠受到美味的吸引，向着出口走去。心理学家就观察老鼠如何由入口到达出口。

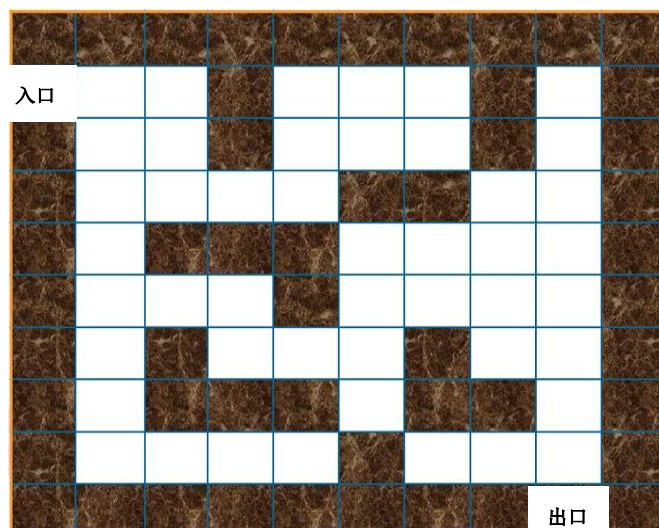


图 1 作为迷宫的箱盒

这个试验可以用来考查老鼠记忆力的强弱。如果它记忆力很好，那么在迷宫中对先前尝试过的失败路径就不会再去尝试。

(2) 功能及界面要求

设计一个模拟小动物走迷宫的程序，为小动物寻找一条从迷宫入口到迷宫出口的通路。最好能以高运行效率找到一条从入口到出口的最短路径。假设老鼠具有稳定记忆力，能记住以前走过的失败路径，而不会重蹈覆辙。

要求：

- 用户可以设置迷宫的行数或列数。
- 随机产生迷宫的状态。
- 用户设置小动物的入口下标和出口下标。
- 根据迷宫状态和入、出口位置直观显示出从入口到出口的通路或“不存在通路”的信息。最好能将所设置的整个迷宫的状态也显示出来。

(3) 数据结构要求

- 迷宫采用二维数组存储。
- 通路上的各个坐标可用队列或栈存储。

(4) 数据结构与算法设计指导

整个算法分为四个子算法：迷宫生成算法；搜索路径算法；输出路径算法；恢复迷宫算法。其中路径搜索算法可以基于广度优先搜索，也可以基于深度优先搜索（即回溯法），也可以基于其它优化搜索策略（如：A*算法、爬山法等），以提高搜索效率。如果按照深度优先搜索（回溯法），找到的路径并不一定是最短的。如果要找到一条从入口到出口的最短路径，就得采用广度优先搜索。

① 基于广度优先搜索迷宫问题求解方法指导

● 迷宫的表示及迷宫的生成算法

在平面图上迷宫是由一些小方格组成的，所以可以用一个 $\text{maze}[m][n]$ 来表示迷宫，数组的每个元素 $\text{maze}[i][j]$ 的值取 0 或 1，取 0 表示此路可通，取 1 表示此路不通。不失一般性，若设迷宫入口是 $\text{maze}[1][1]$ ，出口是 $\text{maze}[m][n]$ ，且 $\text{maze}[1][1]=0$ ， $\text{maze}[m][n]=0$ 。一个迷宫的示意图如图 2 所示。

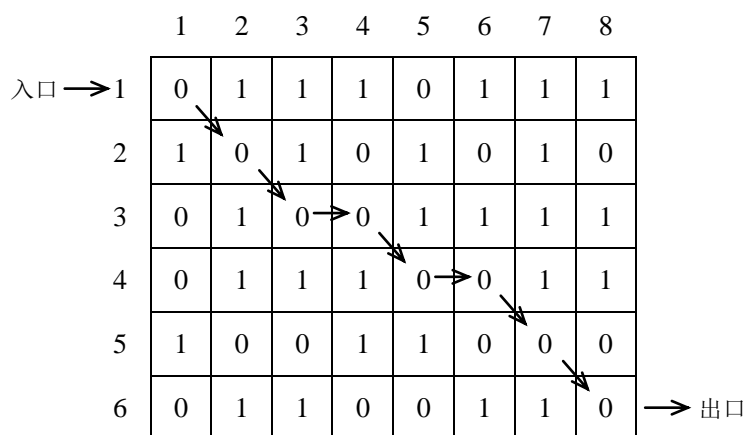


图 2 迷宫及其最短路径示例

若 (x,y) 不是边界点，则与它相邻的点有 8 个，否则与其相邻的点可能只有 3 个或 5 个。为了避免过多的条件判断，可把原来表示迷宫的矩阵扩大一圈，即将迷宫四周各镶上一条取值均为 1 的边，以此表示受阻的墙。这样表示迷宫的二维数组应为 $\text{maze}[m+2][n+2]$ 。于是，对于迷宫中任一点 (x,y) ($1 \leq x \leq m$, $1 \leq y \leq n$)，其搜索方位均有 8 个，如图 3 所示。

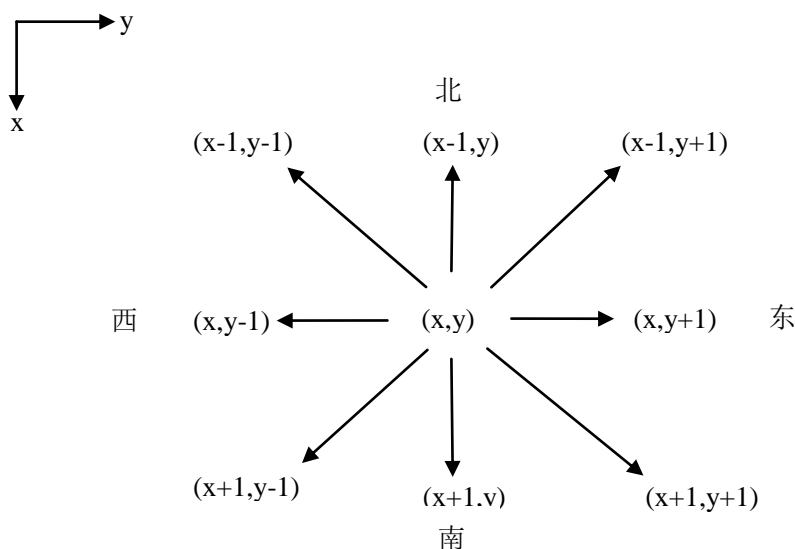


图 3 (x, y) 的八个邻接点的坐标

迷宫类型的定义如下：

```
#define m 6          /*迷宫的实际行数*/
#define n 8          /*迷宫的实际列数*/
int maze[m+2][n+2]; /*迷宫数组*/
```

随机产生“通路”或“墙”时应把“通路”的产生概率比“墙”的产生概率大 2-3 倍。

● 路径搜索算法及相关数据结构

若按广度优先搜索，从迷宫入口出发，向四周搜索，用队列记下所有一步能通达的坐标点 p_{11} , ..., p_{1k_1} ($0 \leq k_1 \leq 3$)；然后依次从 p_{11} , ..., p_{1k_1} 出发，向四周搜索。记下所有经过二

步(从入口算起)能通达的坐标点 $P_{21}, \dots, P_{2k_2} (0 \leq k_2 \leq 4 \text{ 或 } 0 \leq k_2 \leq 7)$; 继续如此进行下去, 直至找到出口点成功结束, 或者当前队列为空以失败告终。

搜索算法涉及搜索路径怎样存储的问题。若采用广度优先搜索, 先到达的点先做下一步搜索的出发点, 故需要引进一个“先进先出”的队列来保存已到达的坐标点。若采用顺序队列, 用一个结构体数组 sq 作为该队列的存储空间。因为迷宫中每个点至多被访问一次, 所以数组 sq 的长度至多为 $m \times n$ 。 sq 的每一个元素有三个域 x, y 和 pre , 其中 x 和 y 分别记下搜索过程中到达的点的行、列坐标, pre 则是一个静态链域(即指针域, 由其构成静态链表), 它记下到达该点的出发点(即搜索路径上的前趋)在 sq 中的下标。

顺序队列的定义如下:

```
typedef struct
{
    int x,y;           /*行、列坐标*/
    int pre;           /*指针域*/
}sqtype;              /*队列元素类型*/
sqtype sq[m*n];      /*队列*/
```

设队列的头尾指针 $front$ 和 $rear$ 分别正好指向队列的队头和队尾元素。开始时, 队列中只有一个元素 $sq[1]$, 其中记录的坐标点是入口, 因为不是从其它点出发到达入口点, 故其 pre 域为 0, $front$ 和 $rear$ 均指向该元素(即将 $front$ 和 $rear$ 均置 1)。此后搜索时, 均是以 $front$ 指向的点作为搜索的出发点, 当搜索到一个可达点时, 将该点的坐标及出发点的下标(即 $front$ 值)入队, 因此, $rear$ 始终指向当前搜索到的可达点。若从 $front$ 所指的点出发搜索完毕, 则出队(注意, 在顺序队列上的出队操作, 只需后移头指针 $front$, 是逻辑上的删除队头元素, 不是物理上的删除, 出队的元素还在数组 sq 中原位置上。这样, 最后可在数组 sq 中找到最短路径), 使 $front$ 指向新的出发点, 继续搜索。搜索过程或者以找到出口点成功结束, 或者以当前队列为空导致失败而告终。图 4 显示了图 2 所示迷宫从入口出发进行搜索的中间状态。

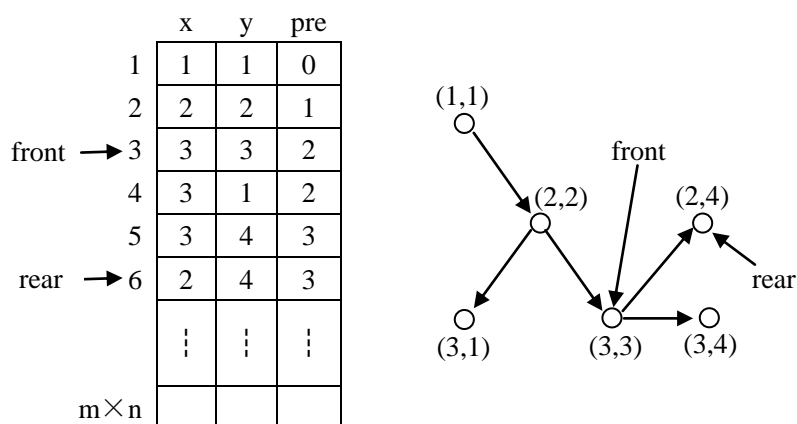


图 4 搜索图 2 所示迷宫的中间状态

不妨设从坐标点 (x, y) 出发搜索的 8 个方位顺序是从正东起沿顺时针方向进行。为了简化问题, 将这 8 个方位上的 x 和 y 坐标的增量预先依次存放在一个结构数组 $move[8]$ 中, 如图 4

所示。该数组的每个元素有两个域 x 和 y 。只要令方向值 v 从 0 增至 7, 便可通过下述计算得到从 (x,y) 出发搜索到的每一个相邻点 (i,j) :

```
i=x+move[v].x;  j=y+move[v].y;
```

	0	1	2	3	4	5	6	7
x	0	+1	+1	+1	0	-1	-1	-1
y	+1	+1	0	-1	-1	-1	0	+1

图 5 move 坐标增量表

在迷宫搜索过程中，为了搜索当前位置周边的 8 个位置，可设置一个坐标增量数组，其定义如下：

```
struct moved
{int x,y;          /*坐标增量，取值-1,0,1*/
}move[8];          /*坐标增量数组*/
```

在搜索算法要解决的一些关键问题是：如何防止重复到达某坐标点？一种方法是，一旦到达某坐标点(x,y)，就将 `maze[x][y]` 由 0 改为 1，这样将破坏原来的迷宫，但是若将 `maze[x][y]` 由 0 改为-1，则搜索过程结束时，可将迷宫中的所有-1 重新置为 0，由此可恢复原来的迷宫。

● 输出路径算法设计

从出口出发, 由静态链表(顺序队列 `sqtype`)回溯直至入口, 将经过的点直接输出(最短路径的逆序)或记录下来后逆着输出(最短路径的正序)。对图 2 所示的迷宫, 图 6 显示了 `sq` 的最终状态, 输出结果以及在迷宫中走过的路径。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	front ↓ 16	17	18	19	rear ↓ 20	...	
X	1	2	3	3	3	2	4	4	1	5	4	5	2	5	6	5	6	6	5	6	...	
y	1	2	3	1	4	4	1	5	5	2	6	6	6	3	1	7	5	4	8	8	...	
pre	0	1	2	2	3	3	4	5	6	7	8	8	9	10	10	11	12	14	16	16	...	

(a) sq 的最后状态

(6,8)
(5,7)
(4,6)
(4,5)
(3,4)
(3,3)
(2,2)
(1,1)

(b) 输出结果

	1	2	3	4	5	6	7	8
1	0	1	1	1	0	1	1	1
2	1	0	1	0	1	0	1	0
3	0	1	0	0	1	1	1	1
4	0	1	1	1	0	0	1	1
5	1	0	0	1	1	0	0	0
6	0	1	1	0	0	1	1	0

(c) 搜索路径

图 6 迷宫的最短路径示例

● 恢复迷宫算法设计

将迷宫中所有-1 值重新改回原来值 0，本算法较为简单。

② 基于深度优先搜索迷宫问题求解方法指导

基于深度优先搜索迷宫问题算法可简单描述如下：

设当前位置的初值为入口位置；

do{

 若当前位置可通，

 则{ 将当前位置插入栈顶；

 若该位置是出口位置，则结束；

 否则切换当前位置的下一个邻接块为新的当前位置；

 }

 否则，

 若栈不空且栈顶位置尚有其他方向未经搜索，

 则设定新的当前位置为沿顺时针方向旋转找到的栈顶位置的下一相邻块；

 若栈不空但栈顶位置的四周均不可通，

 则{ 删去栈顶位置元素；

 若栈不空，则重新测试新的栈顶位置，

 直至找到下一个可通的相邻块或出栈至栈空；

 }

 }while（栈不空）；

(5) 考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。



4. 内部排序性能分析演示程序

(1) 问题描述

分别采用直接插入排序、希尔排序、简单选择排序、堆排序、冒泡排序、快速排序、归并排序算法进行排序，并对比较次数与移动次数进行比较。

(2) 功能与界面要求

- 对随机序列、正序和逆序的进行上述各种排序算法的比较测试和分析。
- 待排序的记录不从键盘输入。随机序列应该是随机生成的 1—N 之间的 n 个整数。
- 要求设置菜单，在菜单提示下，输入 1-7 之间的整数，选择某一种排序方法进行排序，输入 8，进行统计分析(比较上述 7 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)，输入 0，则退出。
- 归并排序算法要求用递归的分治算法实现。
- 当选择功能 1-7 时：
 - 输入：待排序元素个数 n，输入 m（关键字取值范围为 1-m）。
 - 输出：输出自动生成的 n 个整数的初始序列。
 - 输出排序后的序列。
 - 输出初态分别是随机序列、正序和逆序时的比较次数、移动次数。

●当选择功能 8 时：

输入：无。

n 不需要输入，在 1000-3000 之间随机产生，m 也不需要输入，表中关键字在 100-300 之间随机产生。

至少用 5 组不同数据作比较。

比较指标有：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）。

输出：显示比较结果，具体格式如表 4 所示。

表 4 不同初态的各种排序方法的实验比较

n	m	初态	InsertSort		ShellSort		BubbleSort		QuikSort		SelectSort		HeapSort	
			比较	移动	比较	移动	比较	移动	比较	移动	比较	移动	比较	移动
20	100	随机序列	117	98	92	38	169	294	103	50	190	54	109	108
		正序	19	0	62	0	19	0	190	38	190	0	105	100
		逆序	209	190	80	36	190	570	200	38	190	30	121	118
100	1000	随机序列	2351	2252	827	376	4760	6756	865	330	4950	285	1036	787
		正序	99	0	503	0	99	0	4950	198	4950	0	946	711
		逆序	5045	4946	667	259	4950	14838	5000	198	4950	159	1081	838
1000	10000	随机序列	252139	251140	14694	7225	498324	753420	14227	4826	499500	2973	16877	11077
		正序	999	0	8006	0	999	0	499500	1998	499500	0	15960	10307
		逆序	500482	499483	11711	4695	499500	1498449	5000000	1998	499500	1512	17582	11708
1500	10000	随机序列	560594	559095	24384	11685	1123074	1677285	23851	7468	1124250	4467	27139	17553
		正序	1499	0	13507	0	1499	0	1124250	2998	1124250	0	25809	16476
		逆序	1125629	1124130	19075	7053	1124250	3372390	1125000	2998	1124250	2406	28415	18556
3000	10000	随机序列	2278264	2275265	58231	29822	4497015	6825795	52387	16610	4498500	8979	60242	38060
		正序	2999	0	30007	0	2999	0	4498500	5998	4498500	0	57485	35747
		逆序	4501053	4498054	42560	15538	4498500	13494162	4500000	5998	4498500	5100	62892	40260

(3) 数据结构要求

采用顺序存储的线性表。

(4) 算法设计指导

在各种排序算法中增加比较次数和移动次数的统计功能。

需要设计一个逆置算法，将正序表变成逆序表。

需要设计一个算法，产生 n 个随机数(n 是用户指定的元素个数)。

(5) 考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。

5. 通讯录管理系统的设计与实现

(1) 问题描述

设计与实现一个通讯录管理系统，实现按姓名、电话号、QQ 号查找。

(2) 功能及界面要求

要求能够管理通讯录，即具有通讯录的建立、增、删、改记录功能；

要求实现在指定通讯录中按姓名、电话号、QQ 号查找功能。

本题可采用控制台界面，若能用可视化界面更好。控制台参考界面（一级界面）如下：

```
*****
*           通 讯 录 管 理 系 统           *
*                                           *
*          1. 通讯录编辑                   *
*          2. 通讯录查找                   *
*          3. 退出                         *
*****
```

控制台参考界面（通讯录管理子系统二级界面）如下：

```
*****
*           通讯录编辑子系统             *
*                                           *
*          1. 建立通讯录并录入记录         *
*          2. 增加记录                     *
*          3. 删除记录                     *
*          4. 修改记录                     *
*          5. 返回主菜单                   *
*****
```

对应功能说明：

- ① 从无到有建立一个通讯录，以文件形式存储。在建立通讯录的同时允许录入记录。
- ② 在通讯录（文件）尾部增加若干记录。
- ③ 通过姓名找到指定记录，删除指定记录。
- ④ 通过姓名找到指定记录，修改该记录后保存到文件。

控制台参考界面（通讯录查找子系统二级界面）如下：

```
*****
*           通讯录查找子系统             *
*                                           *
*          1. 按姓名查找（散列表查找）     *
*          2. 按电话号查找（二叉排序树查找） *
*          3. 按 QQ 号查找（二分查找）     *
*          4. 返回主菜单                   *
*****
```

对应功能说明：

① 读指定文件内容，以用户名为关键字建立**散列表**，要求完成下列功能：

- 采用一定的方法解决冲突；
- 查找并显示给定用户的记录；

设计不同的散列函数，比较冲突率；在散列函数确定的前提下，尝试各种不同处理冲突的方法，考察平均查找长度的变化。

②读指定文件内容，以电话号码为关键字建立**二叉排序树**。在二叉排序树上查找并显示给定电话号码的记录。

③对从键盘输入的各记录，以 QQ 号为关键字建立**有序表**。在有序表上二分查找并显示给定 QQ 号的记录。

(3) 数据结构要求

可分别用散列表、二叉排序树、有序表等存储通讯录，若能用**键树**、**B 树**、**B+树**等存储通讯录更好。每个记录至少有下列数据项：姓名、手机号、QQ 号、单位、地址。

(4) 数据结构与算法设计指导

具体数据结构与算法设计方法见教材。

(5) 考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。

6. 教学计划编制系统的设计与实现

(1) 问题描述

大学的专业都要指定教学计划。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，而且课程开设时间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。每门课程恰好占 1 学期。在这样的前提下设计与实现一个教学计划编制系统。

(2) 功能与界面要求

输入：学期总数，学分上限，专业开设课程门数，每门课程的编号（整型）、课程名称、学分，先修关系（用课程编号序对给出）。

输出：教学计划。

(3) 数据结构要求

要求用图表示课程及课程间先修关系。

(4) 数据结构与算法设计指导

可设学期总数不超过 10，课程总数不超过 50。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。

可用拓扑排序算法先将该专业的课程进行拓扑排序，然后按拓扑序列将课程分配到各个学期，每学期开设的课程不能超过学分上限。

(4) 考核标准

系统设计与实现占 3 分， PPT、答辩情况、团队合作、创新性、实用性等占 2 分，总计 5 分。

实验九 排序和查找（2 学时）

一、实验目的

- 1、掌握快速排序和二路归并排序的实现方法。
- 2、掌握二叉排序树的建立过程以及二叉排序树上的基本操作。

二、实验内容

- 1、随机生成若干个正整数，分别使用快速排序算法和二路归并排序算法对这些数据进行排序。
- 2、根据随机生成的正整数序列建立二叉排序树，在二叉排序树上进行插入、查找及删除等操作。

三、实验指导

1、题目 1 可以使用一维整型数组存储待排序的数据；要求随机生成 3 组数据，每组数据的整数个数和取值范围由用户输入，对每组数据要求输出两种排序算法的排序结果、比较次数及移动次数。

2、题目 2 的二叉排序树使用二叉链表存储，要求随机生成若干个正整数，正整数的个数和取值范围由用户输入，用户可以根据菜单进行各种操作，例如：

1—建立 2—输出【中序遍历】 3—插入 4—查找【要求输出比较次数】 5—删除 0—退出

3、提示：

（1）题目 1 可参考教材中关于快速排序算法和二路归并排序算法的思想和实现过程加以完善来实现。

（2）题目 2 可参考教材中关于二叉排序树的建立、插入、删除及删除的思想和实现过程加以自己的设计和整合并实现。

四、考核标准

- 1、完成全部题目，设计合理，结果正确；评定为 A
- 2、完成部分题目或未按时完成，设计比较合理，结果正确；评定为 B 或 C
- 3、未独立完成实验要求；评定为 D

（本实验共 2 分。一个实验题目 1 分，未按时完成或算法的时间与空间复杂度过高也会酌情扣分。）

第三部分：选用教材及参考书

1. 实验教材：

《数据结构与算法实验讲义》，黑龙江大学计算机科学技术学院、软件学院编著，2020

2. 参考书：

严蔚敏，李冬梅，吴伟民. 数据结构（C 语言版）（第 2 版）[M]. 北京:人民邮电出版社，2015

第四部分：设计报告模板

一、设计报告大纲

1. 系统概述

开发系统的目的与意义。

概要介绍系统。

2. 系统需求分析

(1) 描述系统工作流程、系统运行环境要求等。

可以分系统管理员、图书管理员、读者三个角色描述系统工作流程。(可用文字描述)

(2) 输入数据的类型、值的范围及输入的形式；输出数据的形式、值的范围及输出的形式；若是会话式输入，结束标志是什么？是否接受非法输入？对非法输入的回答方式是什么等。

3. 系统概要设计

3.1 数据结构设计

给出文字说明及数据类型定义。

3.2 软件结构设计

可用文字叙述结合软件结构图描述。

给出函数列表和函数之间调用关系。

4. 系统详细设计与实现

给出模块的过程描述，可以选择 3-4 个有代表意义的模块完成详细设计，详细设计的表达工具可用文字叙述配合程序流程图或伪代码完成。

给出实现效果截图。

5. 系统测试

系统测试的目的是尽可能的检查出程序中的错误，提高软件系统的可靠性。本文进行了功能测试，测试的任务是通过测试用例在计算机上执行程序暴露程序中潜在的错误。要求选两个模块进行测试。

每个测试用例包括：测试内容 实际输入、输出结果、预期结果。

6. 结论

(1) 完成的主要工作。

(2) 系统优势、存在的问题与不足。

(3) 调试过程中主要遇到的主要问题及解决方法。

(4) 感想与收获。

参考文献

附录

编码风格良好、带注释的源代码。

二、样例

样例 1：音像信息管理系统

3.1 系统概要设计

3.1.1 软件结构设计

音像信息管理系统根据需求分析中的功能分析，可以提炼出主要音像信息、音像类型、媒体类型、地区分类、音像评论五个子模块。音像信息管理模块主要是提供给后台管理员使用，管理员登陆系统后可以完成对音像类型、媒体类型、地区分类的添加、修改和删除操作。由于前后台的相互关联，当管理员对后台信息完成修改时，前台用户浏览页面就会有所体现。对音像信息的管理还包括对音像商品评论的管理，管理员拥有对评论信息审核的权利，若该条评论未通过后台的审核，管理员可以执行删除的操作。系统软件结构如图 7 所示。

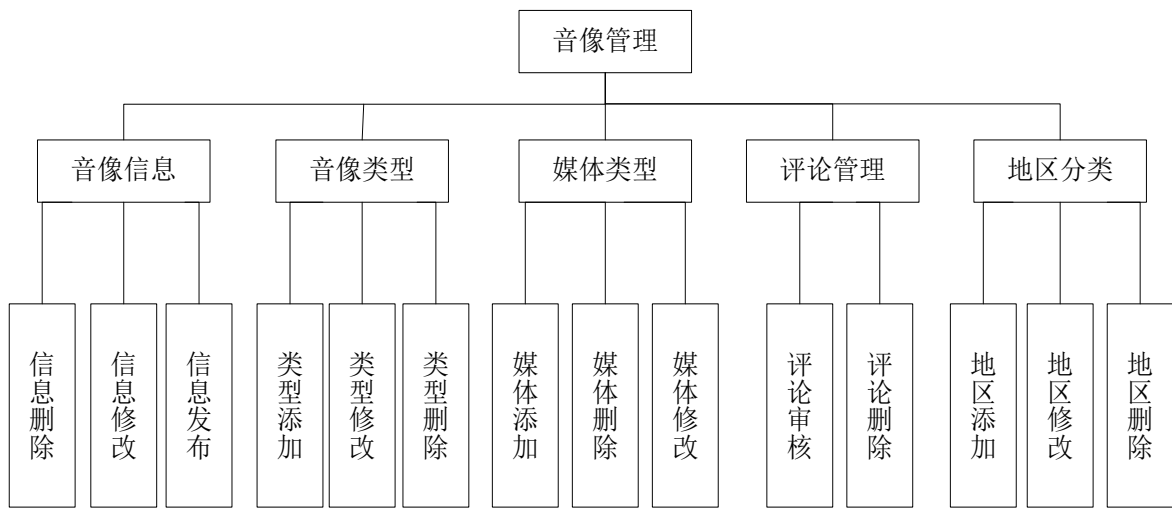


图 7 音像信息管理系统软件模块结构图

样例 2：图书管理系统

4. 借书处理模块详细设计

借书处理模块详细设计阶段算法的基本流程如下：

首先图书管理员输入“读者号”，系统检查读者是否为合法读者，若不是，显示提示信息，并返回；若是，则进一步检查读者借书数量是否超出限额，若超出，给出提示信息，并返回；若未超出，图书管理员输入“书号”、“借书日期”检查书号是否合法，若书号非法，给出提示信息，并返回；否则，继续检查该号图书是否已全部借出，若是，给出提示信息，并返回，否则，将借书文件记录数加 1，借书文件追加 1 条相关记录，读者“已借书数”加 1，图书“借出数”加 1，并提示借书成功。该模块算法流程图如图 8 所示。

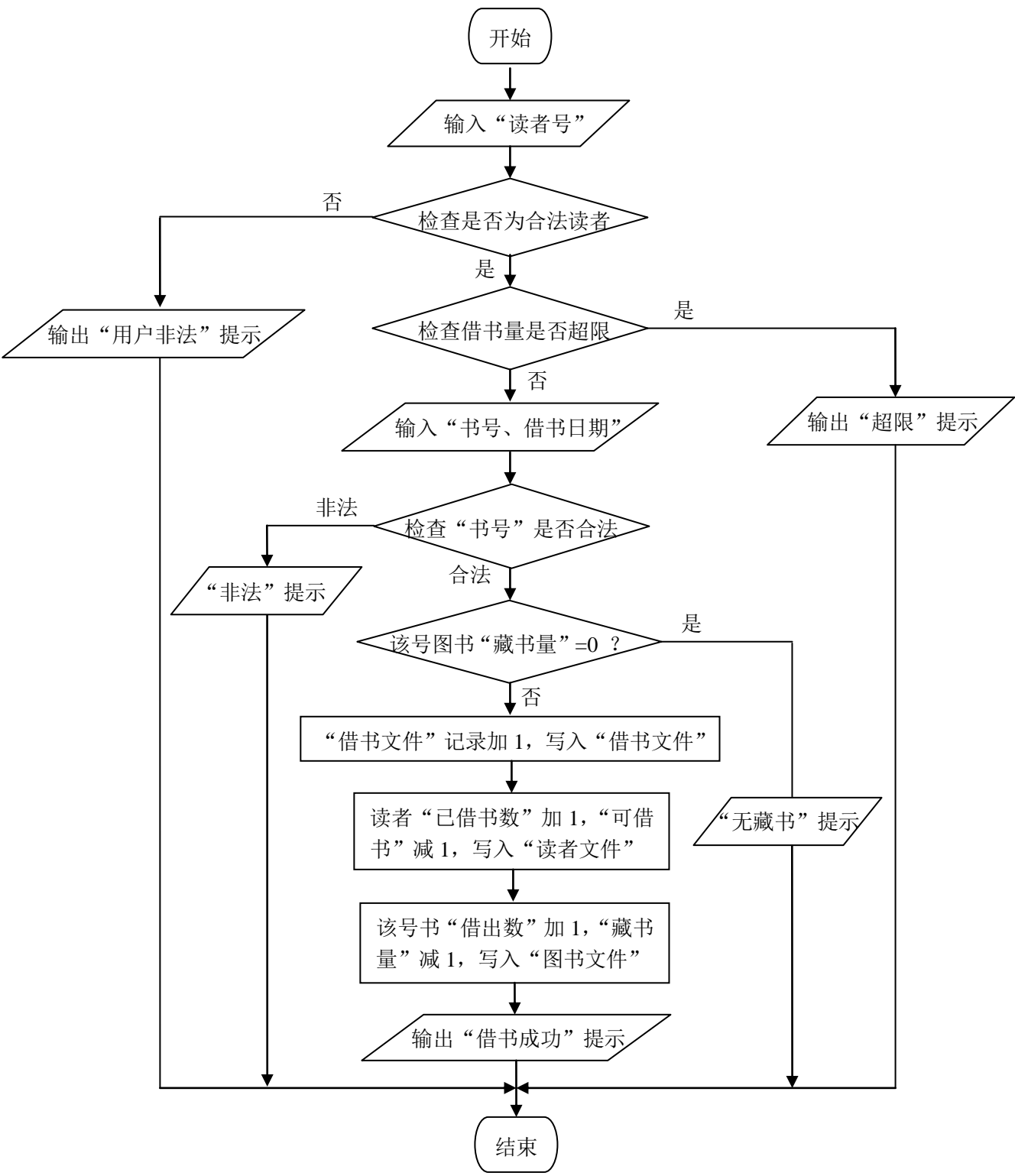


图 8 借书处理模块算法流程图

样例 3：音像出租销售管理系统的设计与实现

5. 系统测试

系统测试的目的是尽可能的检查出程序中的错误，提高软件系统的可靠性。本文进行了功能测试，测试的任务是通过测试用例在计算机上执行程序暴露程序中潜在的错误。

5.1 用户注册测试

在用户注册页面中，用户所要填写的注册信息包括用户名、密码、密码确认、性别、真实姓名、性别、地址、邮箱地址和移动电话等。针对用户注册的功能进行测试，对于用户注册功能模块的测试其测试用例如表 1 所示。

表 1 用户注册的测试用例

测试内容	测试用例	预期结果	实际结果
用户名	1.输入由字母数字（6—16 位）“123aabbcc”	输入成功	与预期相同
	2.输入汉字“你好 1234dfg”	输入失败	
	3.输入字符少于 6 个或多于 16 个“12de”	输入失败	
	4.不输入	输入失败	
密码	1.输入由字母数字（6—16 位）“fheurh123”	输入成功	与预期相同
	2.输入汉字“再见 bbfber3456”	输入失败	
	3.输入字符少于 6 个或多于 16 个“fg5”	输入失败	
	4.不输入	输入失败	
密码确认	1.再次输入一致	输入成功	与预期相同
	2.再次输入不一致	输入失败	

三、设计报告格式要求

1. 封面

- (1) 封面格式见附录 1。
- (2) 题目：小一号，宋体。
- (3) 学院、专业、年级、学号、姓名、日期：用四号宋体；英文、数字用四号新罗马字。

2. 正文中字体

- (1) 一级标题（三号、黑体）
- (2) 二级标题（四号，黑体）
- (3) 在正文中，中文要求用小四宋体字，英文与数字要求用小四新罗马字。

3. 正文中行距

- (1) 通篇 1.5 倍行距。
- (2) 各级标题：段前 0.5；段后 0.5。

4. 插图

- (1) 图题由图号和图名组成。
- (2) 图题置于图下，采用黑体五号字，居中排写。
- (3) 图按出现次序编号，如图 1，图 2，…。
- (4) 图名在图号之后空一格排写。
- (5) 插图与其图题为一个整体，不得拆开排写于两页。
- (6) 论文中插图应该与文字紧密配合。插图应该编排在正文提及之后，任何一个插图在正文中都应该有类似于“……如图 5 所示”之类的文字相配合。插图处的该页空白不够时，则将其后文字部分提前排写，将图移到下一页最前面。

5. 插表

- (1) 表格要求用三线制，不加左右边线，上下底为粗实线（1.5 磅），中间为细实线（1 磅）。比较复杂的表格可适当增加横线与竖线。
- (2) 表题由表号和表名组成。
- (3) 表题置于表上，采用黑体五号字，居中排写。
- (4) 表按出现次序编号，如表 1，表 2，…。
- (5) 表名在表号之后空一格排写，表名中不允许使用标点符号。
- (6) 插表与其表题为一个整体，不得拆开排写于两页。
- (7) 插表中文字，中文要求采用宋体 5 号字，英文采用新罗马 5 号字。
- (8) 论文中插表应该与文字紧密配合。插表应该编排在正文提及之后，任何一个插表在正文中都应该有类似于“……如表 5 所示。”之类的文字相配合。插表处的该页空白不够时，则将其后文字部分提前排写，将表移到下一页最前面。

6. 参考文献

- (1) 文献标识

论文正文中须按出现顺序标识参考文献标号，文献编号用阿拉伯数字置于方括号“[]”中，置于所引内容最末端，按上标处理。如：XXXXX^[1]。

论文中引用文献原文应加引号，若引用愿意，文前用冒号或逗号，不用引号。

(2) 书写格式

(a) 参考文献题头采用小二号黑体字居中书写，段前 0.5，段后 0.5。

(b) 参考文献中的中文采用宋体 5 号字，英文采用新罗马 5 号字编写。

(c) 参考文献按论文引用顺序编排，文献编号顶格书写，加括号“[]”，其后空一格写作者名等内容。

● 连续出版物

[序号] 作者名. 文献题名[J]. 刊名. 出版年份, 卷号 (期号): 起止页码.

例:

[1] 孙继红, 刘奇付. 基于 ASP.NET 的项目任务管理系统[J]. 电脑知识与技术. 2013, 9(21) : 4855-4856.

● 专著

[序号] 作者名. 文献题名[M]. 出版地: 出版者, 出版年: 起止页码.

例:

[2] 何玉洁, 梁琦. 数据库原理与应用[M]. 北京: 机械工业出版社, 2011:25-41.

● 论文集

[序号] 主要责任者.文献题名[C].主编.论文集名.出版地: 出版者,出版年:起止页码.

例:

[3] Ffrederrik Maes, Dirk Vandermeulen, Paul Suetrens. Medical image registration using mutual information[C]. Proceedings of the IEEE, 2003, 91(10):1699-1721.

● 学位论文

[序号] 主要责任者.文献题名[D].保存地:保存单位,年份.

例:

[4] 刘洋. 基于 IOS 的移动视频监控客户端软件的设计与实现[D]. 武汉:武汉邮电科学研究院, 2016.

(5) 报告

[序号] 主要责任者.文献题名[R].报告地:报告会主办单位,年份.

(6) 专利文献

[序号] 专利所有者.专利题名[P].专利国别:专利号,发布日期.

(7) 国际、国家标准

[序号] 标准代号,标准名称[S].出版地:出版者,出版年.

(8) 报纸文章

[序号] 主要责任者.文献题名[N].报纸名,出版日期(版次).

(9) 电子文献

[序号] 主要责任者.电子文献题名[文献类型/载体类型].电子文献的出版或可获得地址,发

表或更新的期/引用日期.

主要责任者为多人时,一般只列出 3 名作者,不同作者姓名间用逗号相隔。外文姓名按国际惯例,将作者名的缩写置前,作者姓置后。学术刊物文献无卷号的可略去此项,直接写“年,(期)”。

英文文献标题除第一个词首字母大写外,其余为小写;俄文文献名第一个词和专用名词的第一个字母大写,其余为小写;日文文献,人名的汉字须用日文汉字,不使用中文汉字及其简化汉字。

附录 1：设计报告封面格式



“*****” 课程设计报告

题目：

学 院	
年 级	
专 业	
学 号	
姓 名	
日 期	
成 绩	
评 语	

黑龙江大学

计算机科学技术学院 黑龙江大学软件学院

附录 2：编程规范

参见华为编程规范。