

上机综合大作业说明

上机综合大作业的题目具有综合应用的特点，编程环境和编程语言不限，建议 C 语言实现。程序界面可以采用字符界面，但更提倡图形界面。

完成统一的上机综合大作业作为上机考核，每组一般为 3 人，以组为单位提交报告，每组根据各位成员的实际工作情况自行申报上机考核成绩。

报告格式要求如下：

1. 选择的题目；
2. 题目功能描述；
3. 问题分析与算法设计思路；
4. 数据结构定义（注意定义上方添加注释）、各个函数功能描述（给出主要函数列表：每条记录应包括函数名及功能简要说明）、程序流程图；
5. 程序源代码，各个函数上方及主要操作语句均添加注释；
6. 含有数据输入/输出的截图，并结合数据的存储结构和所输入的数据进行分析；
7. 如题目中未限定存储结构，则对所采用的存储结构的优缺点加以说明，及采用该存储结构的理由；
8. 若因程序含有错误未完成该题目，需说明原因；
9. 实验心得体会：如解决了什么问题，还存在什么问题，哪些方面得到了提高或受到了启发；在设计该算法时，你遇到的最大困难是什么？你是如何解决的？
10. 在文档末尾列表格给出每个人的详细分工以及成绩申报（每组总分控制在 255 分），注意说明每人承担的具体任务和完成情况。

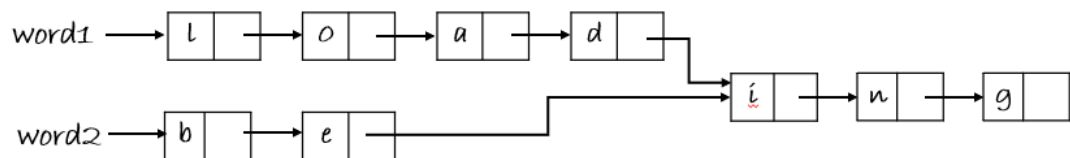
规则说明

1. 完成统一的上机综合大作业作为上机考核，每组一般为 3 人，以组为单位提交报告，要求完成 2-3 个题目（1 高+1 低，或 2 中+1 低），每组根据各位成员的实际工作情况自行申报上机考核成绩；
2. 如果发现某组报告出现抄袭现象均做降分处理。

题目 1:

[问题描述] (难易程度: 低)

利用单链表存储带有同样后缀的单词时为了节约空间, 常常把同样的后缀用一个链表存储, 即共享后缀链表。完成 2 个链表是否含有共同后缀的判断, 如果含有则输出。如下图所示, 给出了单词" loading" 和" being" 的存储形式。



[实验目的]

1. 掌握线性表的链表存储结构。
2. 掌握在单链表上基本操作的实现。
3. 在掌握单链表的基本操作上进行综合题的实现。

[实验内容及要求]

1. 要求求出各个测试用例的 2 个链表的公共后缀是什么, 并输出公共后缀。
2. 给出算法的基本思想, 用 C 实现;
3. 对所设计算法的时间复杂度和空间复杂度进行分析;
4. 选做: 最好能给出另一种不同的算法设计思路 (可仅分析其时间、空间复杂度, 不实现), 对两种算法从算法的时间、空间复杂度、逻辑设计难易等方面进行优缺点对比分析。

[测试数据]

L1	L2	公共后缀
Loading	being	ing
abcdef	wdeegf	无
dagdgkg	yurtkadkg	kg
hhhtt	空表	无
same	same	same

题目 2:

[问题描述] (难易程度: 低)

利用带头结点的单链表实现两个集合的并、交、差运算。

[实验目的]

- 1、掌握线性表的链表存储结构。
- 2、掌握在单链表上基本操作的实现。
- 3、在掌握单链表的基本操作上进行综合题的实现。

[实验内容及要求]

1. 要求用带头结点的单链表存储两个集合中的元素和最终的结果。
2. 集合的元素限定为十进制数, 程序应对出现重复的数据进行过滤, 即使得链表中没有重复数据。
3. 显示两个集合的内容及其并集、交集和差集的内容。
4. 要求不改变原来的集合, 并集、交集和差集分别另外存放。
5. 对所设计算法的时间复杂度和空间复杂度进行分析;
6. 选做: 最好能给出另一种不同的算法设计思路(可仅分析其时间、空间复杂度, 不实现), 对两种算法从算法的时间、空间复杂度、逻辑设计难易等方面进行优缺点对比分析。

[测试数据]

- 1、set1={3, 8, 5, 9, 11}, set2={22, 6, 8, 3, 15, 11, 20}

set1 \cup set2 =

set1 \cap set2 =

set1-set2 =

- 2、其中一个集合为空集
- 3、两个集合都是空集
- 4、创建集合时有重复数据的情况

题目 3:

[问题描述] (难易程度: 中)

已知有一个正整数构成的集合 $A = \{a_i\} (0 \leq i < n, n \geq 2)$, 对其进行集合划分, 得到 2 个不相交的子集合 A_1 和 A_2 。 A_1 和 A_2 中元素个数分别为 n_1 个和 n_2 个, A_1 中所有元素的和为 S_1 , A_2 中所有元素的和为 S_2 。试设计一个尽可能高效的算法, 使得 $\min(|n_1 - n_2|)$ 且 $\max(|S_1 - S_2|)$ 成立。

[实验目的]

掌握线性表的顺序存储结构及其相关操作, 考察该课程所学算法的灵活应用。

[实验内容及要求]

1. 测试至少 3 组测试数据集, 数据集的选取自己设定, 尽可能考虑各种情况, n 要求大于 20。
2. 对所设计算法的时间复杂度和空间复杂度进行分析;
3. 选做: 最好能给出另一种不同的算法设计思路 (可仅分析其时间、空间复杂度, 不实现), 对两种算法从算法的时间、空间复杂度、逻辑设计难易等方面进行优缺点对比分析。

题目 4

[问题描述] （难易程度：中）

利用中缀表达式实现算术表达式的求值。

[实验目的]

- 1、掌握栈的应用。
- 2、掌握算符优先表达式求值的算法。
- 3、掌握字符串处理和数值的转换。

[实验内容及要求]

1、表达式以字符串形式输入，并以‘#’开始和结束（‘#’也作为算法来处理）。表达式中操作数为整数，含有+、-、*、/、（、）等运算符。

如输入： $\#6+3*(9-7)-8/2\#$

2、能够有效判别表达式的输入格式是否有误（如缺失操作数、非法算符、括号不匹配等错误），若输入格式错误，输出错误提示。

[测试数据]

- 1、 $\#6+3*(9-7)-8/2\#$
- 2、 $\#(8-2)/(3-1)*(9-6)\#$
- 3、 $\#5+a*(8-4)/2\#$
- 4、 $\#5+(7-3)*6\#$

题目 5

[问题描述]（难易程度：低）

利用后缀表达式实现算术表达式的求值。

[实验目的]

- 1、掌握栈的应用。
- 2、掌握算符优先表达式求值的算法。
- 3、掌握字符串处理和数值的转换。

[实验内容及要求]

1、表达式以字符串形式输入，并以‘#’开始和结束（‘#’也作为算法来处理）。表达式中操作数为整数，含有+、-、*、/、（、）等运算符。

如输入：#6+3*(9-7)-8/2#

2、能够有效判别表达式的输入格式是否有误（如缺失操作数、非法算符、括号不匹配等错误），若输入格式错误，输出错误提示。

[测试数据]

- 1、#6+3*(9-7)-8/2#
- 2、#(8-2)/(3-1)*(9-6)#
- 3、#5+a*(8-4)/2#
- 4、#5+(7-3)*6#

题目 6

[问题描述]（难易程度：中）

利用前缀表达式实现算术表达式的求值。

[实验目的]

- 1、掌握栈的应用。
- 2、掌握算符优先表达式求值的算法。
- 3、掌握字符串处理和数值的转换。

[实验内容及要求]

1、表达式以字符串形式输入，并以‘#’开始和结束（‘#’也作为算法来处理）。表达式中操作数为整数，含有+、-、*、/、（、）等运算符。

如输入：#6+3*(9-7)-8/2#

2、能够有效判别表达式的输入格式是否有误（如缺失操作数、非法算符、括号不匹配等错误），若输入格式错误，输出错误提示。

[测试数据]

- 1、#6+3*(9-7)-8/2#
- 2、#(8-2)/(3-1)*(9-6)#
- 3、#5+a*(8-4)/2#
- 4、#5+(7-3)*6#

题目 7

[问题描述] （难易程度：低）

根据后序序列和中序序列输出前序遍历序列。

[实验目的]

- 1、熟练掌握二叉树的存储结构；
- 2、熟练掌握二叉树的遍历及应用
- 3、熟练掌握递归算法

[实验内容及要求]

1. 输入说明：第一行给出正在正整数 N，为树中结点个数。随后 2 行，每行给出 n 个字符，分别对应后序遍历和中序遍历结果，题目保证输入正确对应唯一一棵二叉树；
2. 输出说明：在一行中输出 “Preorder:” 及前序遍历序列结果。
3. 对所设计算法的时间复杂度和空间复杂度进行分析；

[测试数据]

输入：

```
7
BCAEGFD
ABCDEFG
```

输出：

```
Preorder: DACBFEG
```

输入：

```
5
CBEDA
CBADE
```

输出：

```
Preorder: ABCDE
```

输入：

```
7
```

GFEDCBA

ACEGFDB

输出：

Preorder: ABCDEF

输入：

1

A

A

输出：

Preorder: A

题目 8

[问题描述] （难易程度：低）

判断一棵树是否是一棵二叉排序树。

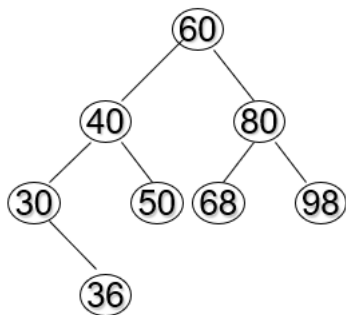
[实验目的]

- 1、熟练掌握二叉排序树的定义和操作；
- 2、熟练掌握递归算法

[实验内容及要求]

- (1) 构建一棵二叉树；
- (2) 输出二叉树的先序、中序、后序遍历序列
- (3) 判断是否一棵二叉排序树

[测试数据]

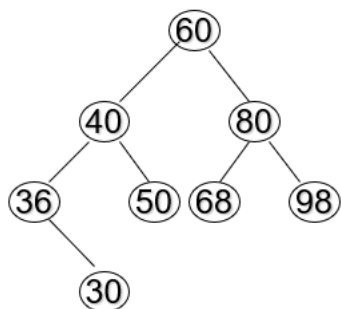


先序：60, 40, 30, 36, 50, 80, 68, 98

中序：30, 36, 40, 50, 60, 68, 80, 98

后序：36, 30, 50, 40, 68, 98, 80, 60

该树是一棵二叉排序树



先序：60, 40, 36, 30, 50, 80, 68, 98

中序：36, 30, 40, 50, 60, 68, 80, 98

后序：30, 36, 50, 40, 68, 98, 80, 60

该树不是一棵二叉排序树

题目 9：停车场管理系统模拟

[问题描述] 设停车场是一个可以停放 n 辆汽车的南北方向的狭长通道, 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, 依次由北向南排列 (大门在最南端, 最先到达的第一辆车停放在车场的最北端), 若车场内已停满 n 辆车, 那么后来的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先退出车场为它让路, 待该辆车开出大门外, 其它车辆再按原次序进入车场, 每辆停放在车场的车在它离开停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。要求程序输出每辆车到达后的停车位置 (停车场或便道上), 以及某辆车离开停车场时应缴纳的费用和它在停车场内停留的时间。(难易程度: 高)

【提示】:

建议用栈模拟停车场, 以队列模拟车场外的便道, 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”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: 若是车辆到达, 则输出汽车在停车场内或便道上的停车位置; 若是车辆离去, 则输出汽车在停车场内停留的时间和应交纳的费用。

题目 10

[问题描述]

利用二叉排序树（又称为二叉查找树、二叉搜索树）实现对输入的英文单词进行搜索，同时可给出单词出现的次数。（难易程度：高）

[实验目的]

- 1、掌握二叉链表的存储结构。
- 2、掌握搜索和过滤的方法。
- 3、掌握二叉排序树的插入和删除操作。

[实验内容及要求]

1、构造二叉排序树

（1）从文本文件中读入文本内容，能够分离出单词，过滤掉阿拉伯数字和标点符号，并将英文字母的大写形式全部转换成小写形式。

（2）按照英文字母表的顺序构造英文单词的二叉查找树。当两个英文单词的首字母相同时，按第二个字母进行排序，依次类推。

（3）当待插入的单词已在二叉查找树中，则将该单词的出现次数增 1。

2、遍历二叉排序树

（1）搜索：输入一个待检索单词，在二叉查找树中进行查找，如果能找到该单词，则输出该单词及其出现次数；

（2）实现二叉排序树的中序遍历，并将遍历结果输出到屏幕上，包括单词和单词出现的位置。

3、删除结点：给定一个停用词列表（停用词是指对搜索没有作用的词，如：of, and, a, an, the 等等），将二叉查找树中的属于停用词表中的单词依次删除。

4、可以显示菜单，在菜单中可以进行如下四项操作（但并不局限这些操作）：

（1）读入文本内容，包含若干英文单词、标点符号以及阿拉伯数字，用于构建二叉查找树。

（2）输入停用词，每个停用词占一行。对于每个停用词，都需要删除二叉查找树中的相应结点，即：每输入一个停用词，执行一次删除结点的操作。

（3）中序遍历二叉排序树，遍历结果中的每个单词占一行，先输出该单词，然后输出一个空格，再输出该单词出现的次数。

(4) 输入查询词。对每个查询词, 都需要在二叉查找树中的搜索相应结点, 如果找到, 则输出该单词及其出现次数; 如果未找到, 则输出相应的信息。每个查询词的查询结果占一行, 先输出该单词, 然后输出一个空格, 再输出该单词出现的次数。

(5) 分析算法的时间、空间复杂度。

[测试数据]

- 1、输入的文本含有大小写字母、阿拉伯数字、标点符号及其它字符。
- 2、单词的数量应当在 30 个以上, 并有一定量的相同单词。、

题目 11

[问题描述]

实现一个简单银行叫号模拟系统。银行有三个窗口可以同时办理业务，当有用户到达银行时，首先选择自己要办理的业务，可以选择一种或多种。系统计算办理此业务所需的时间并显示给用户，然后系统查看有无空闲的窗口，如果有，通知用户到一个空闲窗口办理，如果没有空闲窗口，则需安排用户到某个窗口等候，系统先计算每个队列中用户办理业务的总时间，将用户安排到时间最短的队列等候。模拟输出多个用户办理业务的过程。（难易程度：高）

[实验目的]

- 1、深入理解队列的特性。
- 2、掌握使用队列实现某些问题。

[实验内容及要求]

- 1、建立 3 个队列存放在三个窗口等待的用户。
- 2、建立业务表，描述银行能够办理的各项业务，以及办理业务所需时间。
- 3、建立用户表，描述用户办理的业务，用户的状态等
- 4、可以随机产生用户进入银行的时间，让用户输入所需办理的业务。
- 5、由于输入的数据量较大，可以采用文本文件存放需要输入的数据。
- 6、分析算法的时间、空间复杂度。

[测试数据]

以下数据供参考：

用户 1 在时间 1 到达银行，在 1 号窗口办理业务，需要 1 分钟

用户 1 在时间 2 结束，离开

用户 2 在时间 3 达到。在 1 号窗口开始办理，办理业务需要 4 分钟

用户 3 在时间 3 到达，在 2 号窗口开始办理，办理业务需要 5 分钟

用户 4 在时间 5 到达，在 3 号窗口开始办理，办理需要 8 分钟

用户 5 在时间 6 到达，在 1 号窗口等待，办理业务需要 4 分钟

用户 2 在时间 8 办理完业务，离开

用户 5 在时间 8 在 1 号窗口，办理业务需要 4 分钟

用户 6 在时间 8 到达，在 1 号窗口等待，办理业务需要 6 分钟

用户 7 在时间 8 到达，在 2 号窗口等待，办理业务需要 10 分钟

题目 12（二叉树）

[问题描述]

树和二叉树是最常用的非线性结构(树型结构)，其中以二叉树最为常见。遍历二叉树是二叉树各种操作的基础，它分为先序、中序和后序。（难易程度：中）

[实验目的]

- 1、熟练掌握二叉树的结构特性。
- 2、掌握二叉树的各种存储结构的特点及使用范围。
- 3、通过二叉树的基本操作的实现，从而思考一般树的基本操作的实现。
- 4、熟练掌握各种遍历二叉树的递归和非递归算法。

[实验内容及要求]

- 1、创建二叉树时可以根据前序遍历序列进行创建，或者以其他方式创建二叉树。创建好之后将结点的值写入文本文件。
- 2、创建好二叉链表之后实现以下功能：
 - （1）结点的值从文本文件读出，在屏幕上以树的形式或层次的形式显示。
 - （2）输入一个结点值，输出其双亲及其所有子女，以及兄弟结点值。
- 3、对二叉树进行递归和非递归遍历（先序、中序、后序）。

[测试数据]

- 1、用二叉链表表示一个大家族的家谱，即采用孩子兄弟表示法存储数。根为祖先结点，每个结点的左子树是其第一个孩子，右子树是其下一个兄弟。
- 2、输入的结点值最好能够体现结点之间的双亲、孩子、兄弟之间的关系，以便于测试。

题目 13（哈夫曼树的编/译码器模拟）

[问题描述]

利用哈夫曼编码进行通讯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将待传数据进行预先编码；在接受端将传来的数据进行解码（复原）。对于可以双向传输的信道，每端都要有一个完整的编/译码系统。试为这样的信息收发站写一个模拟哈夫曼的编译码系统。（难易程度：高）

[实验目的]

- 1、通过哈夫曼树的定义，掌握构造哈夫曼树的意义。
- 2、掌握构造哈夫曼树的算法思想。
- 3、通过具体构造哈夫曼树，进一步理解构造哈夫曼树编码的意义。

[实验内容及要求]

1. 从终端读入字符集大小为 n （即字符的个数），逐一输入 n 个字符和相应的 n 个权值（即字符出现的频度），建立哈夫曼树，将它存于文件 `hfmtree` 中。并将建立好的哈夫曼树以树或凹入法形式输出；对每个字符进行编码并且输出。
2. 利用已建好的哈夫曼编码文件 `hfmtree`，对已编码的正文进行译码，输出译码后的正文。
3. 采用文本文件存放文本，先统计文本中的每个字符出现的频率，然后再建立哈夫曼树，并进行编码和译码。

[测试数据]

- 1、用下表给出的字符集和频度的实际统计数据建立哈夫曼树。

字符	A	B	C	D	E	F	G	H	I	J	K	L	M	N
频度	64	13	22	32	103	21	15	47	57	1	5	32	20	57
字符	O	P	Q	R	S	T	U	V	W	X	Y	Z	空格	
频度	63	15	1	48	51	80	23	8	18	1	16	1	168	

并实现以下报文的译码和输出：THIS PROGRAM IS MY FAVORITE

2、采用文本文件存放文本，测试实验内容及要求的第 3 项。

题目 14（图）

[问题描述]

给定一个无向图或有向图，利用深度优先遍历和广度优先遍历对给定图进行遍历，并判断图中是否存在环（回路），如果存在环，输出所有存在的环路。（难易程度：高）

[实验目的]

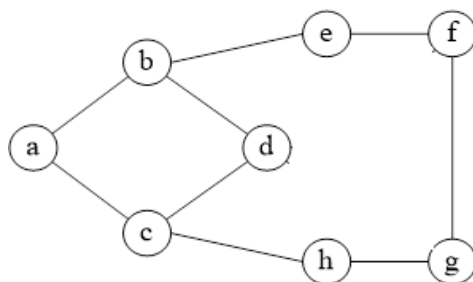
- 1、熟悉图的两种常用的存储结构。
- 2、掌握对图的两种遍历方法，即深度优先遍历和广度优先遍历。
- 3、进一步掌握利用递归或队列结构进行算法设计方法。
- 4、掌握深度优先遍历的应用。

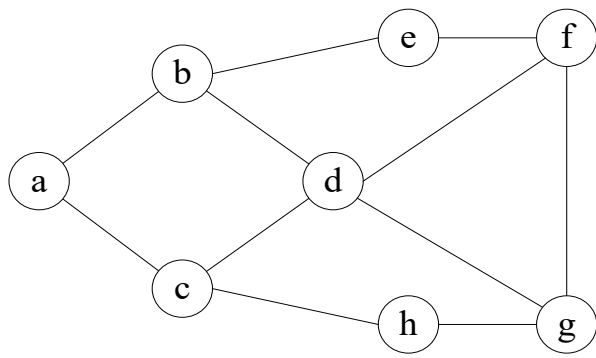
[实验内容及要求]

- 1、构造一个具有 n 个顶点的无向图和有向图。
- 2、输出以深度优先遍历和广度优先遍历图的顶点序列。
- 3、把递归实现的 3 种深度优先遍历算法改为非递归算法。
- 4、基于深度优先遍历判断是否存在环，存在时输出所有的环。
- 5、对各个算法的时间复杂度、空间复杂度进行分析。
- 6、针对有向图判断回路的算法及无向图判断回路的算法进行分析对比，有什么异同。

[测试数据]

以下图 a、图 b 作为测试数据：





题目 15（求无向网的最小生成树）

[问题描述]

如果要在 n 个城市之间建设通信网络，只需架设 $n-1$ 条线路即可。如何以最低的经济代价建设这个通信网，是求一个无向网的最小生成树问题。（难易程度：低）

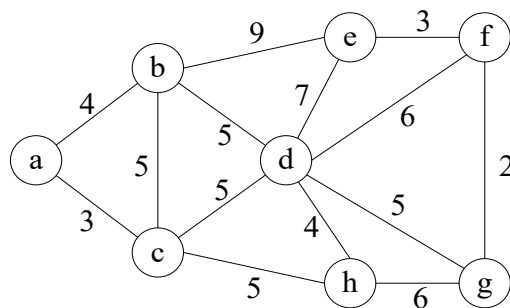
[实验目的]

- 1、掌握图的各种存储结构和基本操作。
- 2、对于实际问题的求解能够选用合适的存储结构。
- 3、通过 Prim 算法理解如何求无向网的最小生成树。
- 4、通过 kruskal 算法理解如何求无向网的最小生成树。
- 5、分析对比两种算法优劣。

[实验内容及要求]

- 1、构造具有 n 个顶点的无向网，并利用 Prim 算法求网的最小生成树。
- 2、以文本形式输出所求得的最小生成树中各条边以及它们的权值。

[测试数据] 以下图作为测试数据：



题目 16（串、模式匹配 KMP 算法）

[问题描述]

给定一个文本文件，完成以下操作：

- 1、统计英文文章中单词个数、空格个数和标点符号的个数，统计给定单词在文本中出现的总次数；
- 2、查找指定的单词，输出其出现在文本中的行号、在该行中出现的次数以及位置
- 3、指定词查到后可以替换为新的单词。（难易程度：高）

[实验目的]

- 1、掌握串的各种存储结构。
- 2、根据实际问题选用合适的存储结构。
- 3、掌握串的运算和模式匹配算法。

[实验内容及要求]

- 1、通过模式匹配算法及串的运算完成单词的查找和替换。
- 2、以文本文件的形式存放英文文章，现将文章读入到串中，读入时需使用 EOF 函数判断文件结束标记。

[测试数据]

英文文章的长度不能过短，至少满 1 页内容，必须能够说明问题。

题目 17（排序）

[问题描述]

排序是对一组记录按照记录的关键字有序进行重新排列,通过几种典型的排序算法,并对各种算法的特点、使用范围和效率进一步地了解。(难易程度:低)

[实验目的]

- 1、深刻理解排序的定义和各类排序的算法思想,并能够根据实际情况灵活应用。
- 2、掌握各类排序的时间复杂度的分析方法,能从“关键字间的比较次数”分析算法的平均情况、最好情况和最坏情况。
- 3、掌握不同排序方法的空间复杂度的计算。
- 4、理解排序方法“稳定”和“不稳定”的含义。
- 5、对结果做出简单的分析,诸如稳定性、最好情况、最坏情况、平均情况等。

[实验内容及要求]

- 1、实现直接插入排序、希尔排序、快速排序、归并排序这 4 种算法。
- 2、根据给定的输入数据实现数据的非递减排序,统计关键字的比较次数和元素移动次数。
- 3、程序中可以采用菜单的方式选择不同排序算法。
- 4、根据测试结果分析几种算法的优劣及适用场合。

[测试数据]

数据由输入或随机函数产生(还应考虑正序、逆序和随机序列)。

题目 18 通讯录管理系统的设计与实现

【问题描述】

通讯录管理系统一般包含通讯者信息的插入、删除、增加、更新、排序等功能。

通讯者信息包括编号、姓名、性别、电话、地址等。（难易程度：中）

【实验目的】 掌握单链表的基本操作上进行综合题的实现。

【实验内容及要求】

- 1、链表定义时每个信息结点仅包含一个数据域和一个指针域。首先将数据域用一个结构体表示，各成员项（编号、姓名、性别、电话、地址等）可用字符数组定义。用该结构体作为链表数据域 data 的数据类型。

- 2、设计菜单界面内容：程序运行后显示 7 个菜单项的内容和输入提示：

通讯录管理系统

=====

<1>通讯录链表的建立；

<2>新的通讯者信息插入；

<3>通讯者信息的查询；

<4>通讯者信息的修改；

<5>通讯者信息的删除；

<6>通讯者信息的全部显示；

<7>退出通讯管理系统；

请选择功能键 1-7：

选 1 后按回车，显示：

通讯录链表的建立

请顺序输入编号、姓名、性别、电话和地址

1002 李小伟 男 14578024323 安徽工业大学人事处，（按下回车后系统提示：）

继续输入吗？（y/n）:y

请顺序输入编号、姓名、性别、电话和地址

o o o o o o o

3、输入 20 人左右的名单，每一个功能均测试到位，并截图放报告里。

题目 19

交通问题

【问题描述】设计一个交通咨询系统，在系统能让旅客查询从任一个城市顶点到另一个城市顶点之间的最短路径问题。采用图来构造各个城市之间的联系，图中顶点表示城市，边表示各个城市之间的交通关系，所带权值包括两个城市间的交通费用、距离、时间。（难易程度：中）

【实验目的】掌握图的最短路径算法实现。

【实验内容及要求：】这个交通咨询系统可以回答旅客提出的各种问题，例如：如何选择一条路径使得从 A 城到 B 城途中中转次数最少；如何选择一条路径使得从 A 城到 B 城里程最短；如何选择一条路径使得从 A 城到 B 城花费最低等等的一系列问题。

- 1、构造具有 10 个以上顶点的交通网络图；
- 2、求从 A 城市出发到其他城市的最短路径问题；
- 3、求两两城市顶点之间的最短路径；
- 4、清楚描述算法实现的核心思想，对第 3 个问题有无不同的算法实现，如果有，在实现的基础上对比分析优劣。
- 5、以文本形式输出所求得的最短路径中各条边以及它们的权值。

[测试数据]

按照题目的问题自行设计 10 个结点的交通网络无向网作为测试数据，报告中画出该带权值的网络图。