

# 数据结构实训

ICT数据结构课程组

# ▶▶▶ 实验授课方式与实验要求总说明

## 1、授课方式

采用统一授课和线下答疑相结合的方式进行。

平时：在QQ群里直接问、老师适时答疑。

每次实验需要到实验室进行，主要是教师指导答疑。

## 2、考核方式

- 1) 实验报告25%+实验单测25%+总测验（所有实验）50%。
- 2) 各专业单测由任课老师单独组织进行。注：为保证公平性，单测可能不在实验课内进行，另行安排时间采用雨课堂统一测试。
- 3) 测试内容为实训课相关程序设计题。
- 4) 总测采用课程组（自愿加入）按期末考试要求统一进行的方式进行，时长为2学时。

## 实验统一要求

- 1) 学生需要根据数据结构实训大纲和任课老师要求，预习并完成主要算法设计与实现；
- 2) 每次实验课都要按照给的实验报告格式提交实验报告**纸质**版，提交日期为下一次实验课，最后一次实验报告提交日为课后的一周（学委负责收并按选课序号排好，交给对应任课教师）。
- 3) **电子版和程序**包命名方式：PDS实验序号（二位0左补）选课序号（三位0左补）学号姓名.rar 如：PDS010232220193216马澍.rar
- 4) 学委统一收齐压缩成一个文件发给对应老师，压缩文件名为：2021PDS实验序号（两位0左补）.rar，如2021PDS01.rar
- 5) 提交日期20点前发给学委，学委24点前发给任课教师。

## ▶▶▶ 实验总体要求

1. 遵守工程伦理和学术规范;
2. 每个实验要解决的问题都要来源于实际工程问题的抽象, 鼓励对国家重大工程中的一个具体点进行抽象建模求解;
3. **课前编写实现实验内容的程序;**
4. 所设计的程序需要包含一个测试主函数, 用于运行验证所设计程序的正确性;
5. 实验结果的分析必须基于**100个以上**数据样本点, 每个样本点的维数不能低于2;
6. 测试程序的输入输出数据以文件形式提供, 具体形式不限定。但为了便于观看, 在不影响性能的前提下可以为文本文件形式, 如各个数据点之间以回车换行分开, 每个数据点的各维之间以逗号分隔开。
7. 不能基于已有的模板类库和算法库实现上述功能, 但可参考开源代码。
8. 按时提交实验报告并且要符合写作规范。

# 实验01

## 多维数字信号的线性表表示及操作

## ▶▶▶ 实验目标

1. 深入理解线性表的存储结构及操作的基本原理和方法；
2. 掌握线性表的各种操作（建立、插入、删除等）的实现方法；
3. 能够实现多维信号的简单运算和操作，如均值滤波、上采样和下采样等。

## ▶▶▶ 实验内容

1. 多维信号要来源于实际工程：要对日常生活中的一个具体应用进行抽象建模，鼓励对国家重大工程中的一个具体点进行抽象建模。

声音

生物医学信号,如心电图等

视频

公交车的行驶轨迹

....

## 实验内容

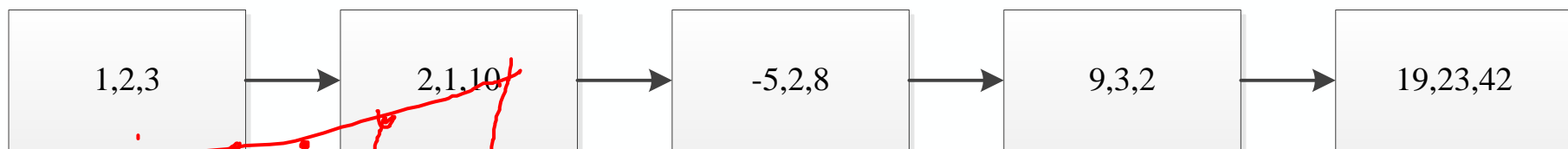
2. 为一个不定长多维数字信号设计一个数据结构（形如下图所示），并实现对其进行基本操作，包括初始化、求数据元素个数操作、插入、删除、取数据元素、判非空操作。





## 实验内容

3.能够实现多维信号的简单运算和操作（数字信号处理课讲授的基本方法），如均值滤波、上采样和下采样等。



**均值滤波：**滤波每个结点的值等于前后结点的均值，如滤波器长度为3，则第二结点滤波后的值为： $([1\ 2\ 3] + [2\ 1\ 10] + [-5\ 2\ 8])/3$ 。

**上采样：**增加结点数目，如上采样后的链表中的结点数目是原链表中结点数目的2倍，对于新增的结点的值可以用插值方法计算，如新增结点的值可以为相邻两个结点的均值等。



## 1. 多维信号由实际问题抽象

在实验目的部分要写明抽象建模的问题。

在算法描述与流程部分写明建模原理和算法

## ▶▶▶ 实验要求

2.深入理解典型存储结构及操作的基本原理和方法，要对各种方案进行比较分析；

在算法描述与流程部分给出各个方案的原理，并作必要的理论分析。

在算法实现部分给出实现方法，包括核心代码。

## ▶▶▶ 实验要求

3.要对实验结果进行和解释，并通过信息综合得到合理有效地结论.

在实验结果与分析部分进行。

## 实验要求

4. 所设计的程序需要包含一个测试主函数，用于运行验证所设计程序的正确性。

测试程序形式为：

main()

{

//初始化现场;

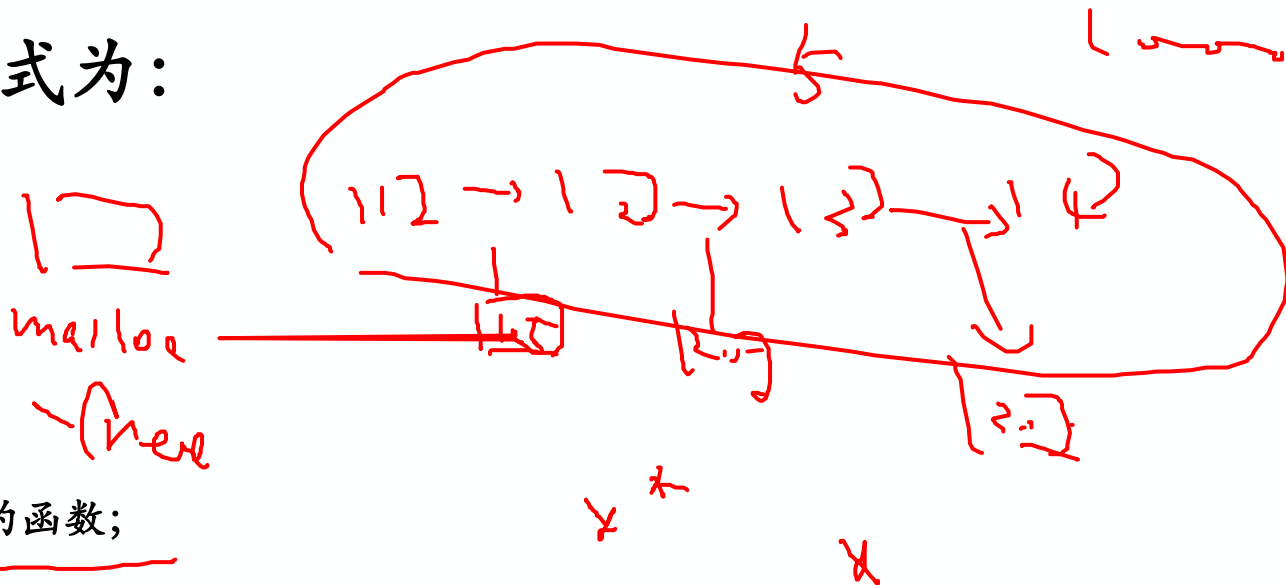
//调用读取数据的函数;

//显示一维信号

//调用线性表操作函数并显示两个类型结果 (真实结果、实际运算的结果)

//信号简单运算函数并显示两个类型结果 (真实结果、实际运算的结果)

//清理现场



## 实验要求

5. 测试用例：简单到复杂、边界条件。

1) 简单输入输出测试用例：很容易知道真实答案的用例

数据：1,2,3,5等

输入：固定数组、scanf、printf

2) 复杂输入输出测试用例：读文件。

测试程序的输入数据以文本文件形式提供，输出亦为文本文件形式；各个数据点之间以回车换行分开，每个数据点的各维之间以逗号分隔开。例如数据点是三维数据，数据文件格式如下：

1,2,3

2,1,10





-5,2,8

9,3,2

19,23,42

1, 2, 3 →

## 3) 边界条件

- (1) 如果输入条件规定了值的范围 (或是规定了值的个数), 则应取刚达到这个范围的边界的值, 以及刚刚超越这个范围边界的值作为测试输入数据;  

- (2) 如果输入条件规定了值的个数, 则用最大个数, 最小个数, 比最小个数少一, 比最大个数多一的数作为测试数据;  

- (3) 如果程序的规格说明给出的输入域或输出域是有序集合, 则应选取集合的第一个元素和最后一个元素作为测试用例;  
  

- (4) 如果程序中使用了一个内部数据结构, 则应当选择这个内部数据结构的边界上的值作为测试用例;
- (5) 分析规格说明, 找出其它可能的边界条件。

## ▶▶▶ 实验报告格式说明



# ▶ ▶ ▶ Academic Honesty Violations in Data Structures (DS)

**In DS, the MINIMUM penalty recommended for a violation of the Academic Honesty Policy will be a ZERO ON THE ASSIGNMENT, PROJECT or EXAM and a LOWERING OF YOUR FINAL GRADE for below what is otherwise earned. You may NOT withdraw from the course if found guilty. Some examples of academic misconduct in DS include but are not limited to the following actions:**

- 1. Picking up and using or discarding another student's written or computer output;**
- 2. Using the computer account of another student;**
- 3. Representing as one's own the work of another on assignments, quizzes, and projects;**
- 4. Giving another student a copy of one's work on an assignment before the due date.**
- 5. Copying work from online resources (Baidu, Chegg, google forums, etc.)**
- 6. Posting work to online resources where other students can view your work**

**All submissions will be checked for similarity. This code will check each submission for similarity between other student submissions, past student submissions, the solution manual, and online resources and postings. If your submission is flagged for a high level of similarity, it will be turned in for an academic honesty violation if deemed appropriate.**

**NOTE: Changing variable names, adding comments, or spacing will result in a violation.**

## ▶▶▶ I/O 操作：输入输出

**在操作系统中，为了统一对各种硬件的操作，简化接口，不同的硬件设备也都被看成一个文件。**

文件是指存储在外部介质上的数据的**序列**。

操作系统是以文件为单位对数据进行管理的。

根据数据的组织形式，可分为ASCII文件和二进制文件。ASCII文件又称文本文件，它的每一个字节放一个ASCII代码，代表一个字符。二进制文件是把内存中的数据按其在内存中的存储形式原样输出到磁盘上存放

在操作系统中，为了统一对各种硬件的操作，简化接口，**不同的硬件设备也都被看成一个文件**。对这些文件的操作，等同于对磁盘上普通文件的操作。例如，通常把显示器称为标准输出文件，`printf`就是向这个文件输出，把键盘称为标准输入文件，`scanf`就是从这个文件获取数据。

参考资料：<https://www.cnblogs.com/lanhaicode/p/10320097.html>

# ▶▶▶ 标准设备输入输出

文件	硬件设备
stdin	标准输入文件，一般指键盘；scanf()、getchar() 等函数默认从 stdin 获取输入。 char cFlag;float fWeight= 0.0;int arr[20]; scanf( "%f%d" ,&fWeight,arr);//地址，改变实参内容 cFlag=getchar(); );//返回值改变变量内容
Stdout	标准输出文件，一般指显示器；printf()、putchar() 等函数默认向 stdout 输出数据。 printf( "%f%d" ,fWeight,arr[0]); putchar(cFlag);
stderr	标准错误文件，一般指显示器；perror() 等函数默认向 stderr 输出数据（后续会讲到）。
stdprn	标准打印文件，一般指打印机。

## 文件的读写

- ✓ 定义文件变量: `FILE *fpTxt, *fpBin;`
- ✓ 打开文件: `FILE *fopen(char *filename, char *mode);`

```
fpTxt = ( "d:\\data.txt" , "r"); fpBin = ( "d:\\data2.bin" , "wb");  
if (fpTxt==NULL) printf("Error on open");
```

- ✓ 遍历: `while (!feof(fpTxt)) { ... }`

### ✓ 读写

```
fprintf(fp, "%d , %s\n", i, str);  
//1,ab[换行]  
fscanf(fp, "%d, %s\n", &i, str);  
//文本文件内容: 1,ab[换行]  
ch=fgetc(fpTxt);fputc(ch,fpTxt);  
fgets(str, N, fpTxt);//读一行  
fputs(str,fpTxt);
```

- ✓ 关闭: `fclose(fpTxt); fclose(fpBin);`

```
//size_t fwrite(const void* buffer, size_t  
size, size_t count, FILE* stream);  
//fwrite(数据类型指针, 数据类型大小,  
写入的数据类型个数, 文件指针);  
fwrite(a,sizeof(Node),1,fp);//将结构体直  
接写入文件  
//size_t fread (void *buffer,size_t  
size,size_t count,FILE *stream);  
fread(a,sizeof(Node),1,fp);//从文件中读取  
一个类型的结构
```

## ▶▶▶ fread和fwrite示例

```
typedef struct car{
    char car_model[10];
    int car_number;
    int car_sales_price;
    int car_purchasing_price;
    int car_sales_number;
}Car;
typedef struct Node{
    struct car car_data;
    struct Node *next;}Node;
```

```
void initdata(Node* head){
    FILE *fp = fopen("car.txt","rt");
    Node *temp = head;
    Node *p = (Node*)malloc(sizeof(Node));
    while(fread(&p->car_data,sizeof(p->car_data),1,fp) ==
    1){
        temp->next = p;
        p->next = NULL;
        temp = temp->next;
        p = (Node*)malloc(sizeof(Node));
    }
    free(p);
    fclose(fp);}
```

```
void savedata(Node* head)
{   FILE *fp = fopen("car.txt","w"); head = head->next;
    while(head){
        fwrite(&head->car_data,sizeof(head->car_data),1,fp);
        head = head->next;    }
    fclose(fp);}
```

## ▶▶▶ C++文件的读写

```
#include <fstream>
```

```
ifstream inFile("book.txt");//读文件
```

```
inFile>>BK_head1>>BK_head2>>BK_head3;
```

```
while(!inFile.eof()) //逐行依次读取所有图书数据
```

```
    inFile>>L.BK[i].no >>L.BK[i].name>>L.BK[i].price;
```

```
inFile.close();
```

```
ofstream outFile("book_new.txt"); //写文件
```

```
for( i=0;i<L.length;i++)
```

```
    outFile<<setw(15)<<L.BK[i].no<<"\t"<<setw(50)<<L.BK[i].
```

```
    name<<"\t"<< setw(5)<<L.BK[i].price<<endl;
```