

**程 序 设 计 专 题**

**大 程 序 报 告**



大程名称： 排序算法可视化

小组成员：

1. **姓名 ： 学号： 电话：**
2. **姓名 ： 学号： 电话：**
3. **姓名 ： 学号： 电话：**

**指导老师： 许端清**

**2020~2021春夏学期 2021 年 6 月 16 日**

**报告撰写注意事项**

1. 图文并茂。文字通顺，语言流畅，无错别字。
2. 书写格式规范，排版良好，内容完整。
3. 存在拼凑、剽窃等现象一律认定为抄袭；0分
4. 蓝色文字为说明，在最后提交的终稿版本，请删除这些文字。

目录

[1 大程序简介 4](#_Toc74940066)

[1.1 选题背景及意义 4](#_Toc74940067)

[1.2 目标要求 4](#_Toc74940068)

[1.3 术语说明 4](#_Toc74940069)

[2 需求分析 4](#_Toc74940070)

[2.1 业务需求 4](#_Toc74940071)

[2.2 功能需求 5](#_Toc74940072)

[2.3 数据需求 5](#_Toc74940073)

[3 程序开发设计 6](#_Toc74940074)

[3.1 总体架构设计 6](#_Toc74940075)

[3.2 功能模块设计 6](#_Toc74940076)

[3.3 数据结构设计 6](#_Toc74940077)

[3.4 源代码文件组织设计 7](#_Toc74940078)

[3.5 函数设计描述 9](#_Toc74940079)

[4 部署运行和使用说明 22](#_Toc74940080)

[4.1 编译安装 22](#_Toc74940081)

[4.2 运行测试 22](#_Toc74940082)

[4.3 使用操作 23](#_Toc74940083)

[5 团队合作 26](#_Toc74940084)

[5.1 任务分工 26](#_Toc74940085)

[5.2 开发计划 26](#_Toc74940086)

[5.3 编码规范 27](#_Toc74940087)

[5.4 合作总结 27](#_Toc74940088)

[开发亮点 27](#_Toc74940089)

[挑战点 27](#_Toc74940090)

[应用知识点 27](#_Toc74940091)

[讨论记录 28](#_Toc74940092)

[5.5 收获感言 28](#_Toc74940093)

[6 参考文献资料 28](#_Toc74940094)

排序算法可视化大程序设计项目

# 大程序简介

## 选题背景及意义

排序算法是大多数人学习算法时最先接触到的内容，多样的算法实现体现出了程序设计的精妙性。然而对于初学者，也许还不大熟练，理解起算法的实现方式或多或少有一些困难，用一种直观的方法去解释或许更能让人轻松接受。

因此本次我组选题为排序算法可视化。其一是该项目包含了大多数本学期所学的知识点；其二是该项目分工明确，既有后端实现可视化的具体算法，也有前端复杂的界面设计，同时网上各种其它语言实现的可视化也具有一定的启发性。

我组希望通过此次项目，巩固本学期有关结构、动态内存分配、图形化编程以及文件操作的知识点，更加了解模块化程序设计。

## 目标要求

\* 能更熟悉运用 `libgraphics` 函数库

\* 能更熟悉文件在C语言中的实现及处理

\* 理解程序运行时与Windows系统的交互

\* 学习如何与他人分工协作

## 术语说明

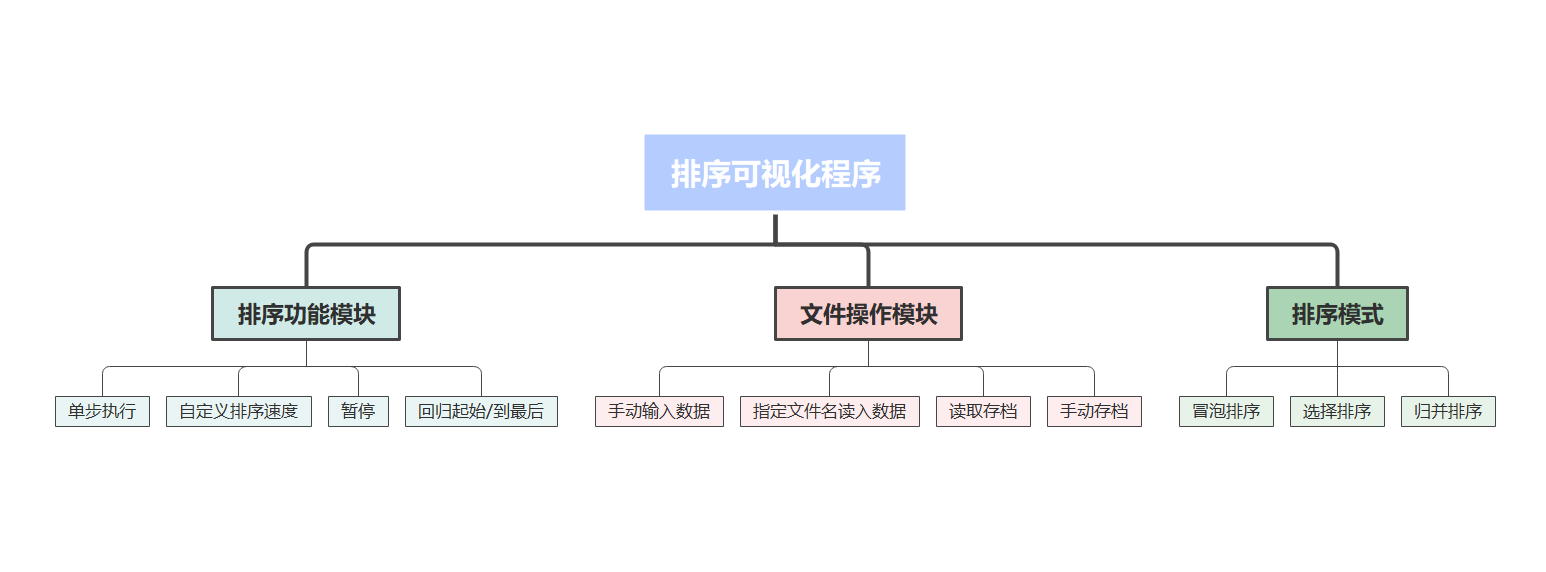
无

# 需求分析

## 业务需求

为算法初学者提供直观的排序算法实现过程，向大众展示计算机算法的精美，可供学龄前儿童娱乐。

## 功能需求

该可视化程序提供冒泡排序、选择排序、归并排序三种可视化选项，每个选项将对应的排序算法可视化。文件操作方面，支持手动输入数据类型或者输入指定文件名来读取数据。同时能够在运行排序时手动存档，在关闭重新打开程序后能读取存档继续未完成的步骤。功能方面，我们制作了手动进行排序的单步执行功能、自定义排序速度功能、暂停功能以及直接到起始/到结尾。

## 数据需求

**静态数据**

无

**动态输入数据**

读取文件名

待排序数据

排序速度

**动态输出数据**

可视化柱状图高度

可视化柱状图起始坐标

**内部生成数据**

可视化柱状图单位高度

可视化柱状图宽度

排序算法走的步数

**数据约定**

文件内容必须符合格式，格式详情见testfiles中的文件

美观起见，浮点类型数据将保留小数点后两位

手动输入数据当严格符合世界现行标准的十进制数字表达方式

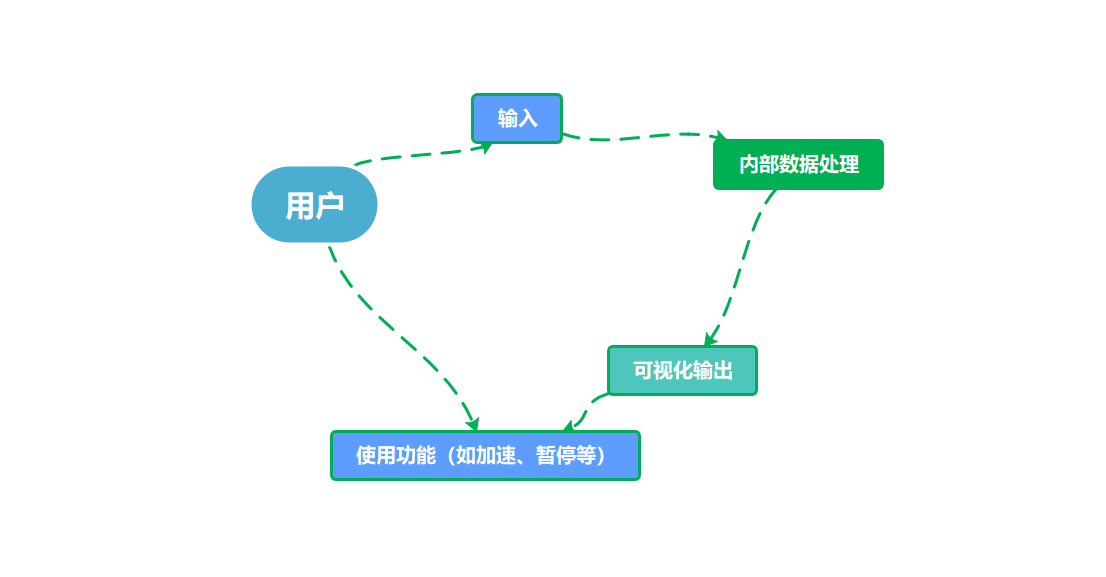
**数据采集要求和范围**

输入的待排序数据个数不宜过大，最好不超过50个，防止界面空间不足造成的挤占以及时间复杂度过高造成的卡死

输入的待排序数据不可相差过大，最好最大数不超过最小数的100倍

# 程序开发设计

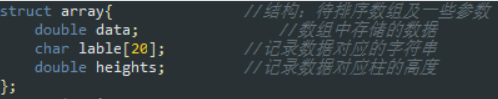
## 总体架构设计



## 功能模块设计

在2.2功能需求中已有介绍

## 数据结构设计



结构说明：代表待排序数组可视化过程需要的参数

结构描述：data——数组元素的数值

lable[20]——该元素数值转化的字符串

heights——该元素对应可视化柱状图的高度参数

InitGraphics.c进行整体框架的展现部分

Visualization.c进行数据、文件的处理以及排序可视化

MergeSort.c进行归并排序部分

Preparation.c进行一些内置数据的判断，为显示弹窗作准备

## 源代码文件组织设计

<文件目录结构>

1）文件函数结构

以文件树形式列出：

.

├── testfiles(放置测试数据文本，供用户参考格式)

└── Source(主要源代码)

├──排序可视化.dev

├── libgraphics

│ ├── exceptio.c

│ ├── imgui.c

│ ├── linkedlist.c

│ ├── random.c

│ ├── genlib.c

│ ├── simpio.c

│ ├── graphics.c

│ ├── strlib.c

├── include

│ ├── ExternVariables.h

**| |** ├── struct array

**| |** ├── isInt

**| |** ├── isFloat

**| |** ├── isDouble

**| |** ├── len

**| |** ├── rch[7]

**| |** ├── struct array a[100000]

**| |** ├── pause\_flag

**| |**  ├── step\_flag

**| |**  ├── toend

**| |** ├── toset

**| |** ├── struct array arrayset[100000]

**| |** ├── step

**| |** ├── isBubble

**| |** ├── isSelection

**| |**  ├── isMerge

**| |** ├── cnt

**| |** ├── endcnt

**| |** ├── stepcnt

**| |** ├── isshow

**| |** ├── end\_flag

**| |** ├── sortspeed

**| |** ├── tmpx

**| |** ├── tmpi

**| |** ├── tmpj

**| |** ├── showstep

**| |** ├── sortstep

**| |** ├── endstep

**| |** ├── stepmode

**| |** ├── pausemode

│ ├── exception.h

│ ├── extgraph.h

│ ├── gcalloc.h

│ ├── genlib.h

│ ├── graphics.h

│ ├── imgui.h

│ ├── linkedlist.h

│ ├── random.h

│ ├── simpio.h

│ ├── strlib.h

├── Main.c（主文件）

├── InitGraphics.c

│ ├── DrawBox()

│ ├── InitWindowGraphics()

│ ├── RedisplaySortbuttons()

│ ├── RedisplayTextStrings()

│ ├── MouseEventProcess()

│ ├── KeyboardEventProcess()

│ ├── is\_inbox()

├── Visualization.c

│ ├── InitArray()

│ ├── Display()

│ ├── Visualize()

│ ├── SelectionSort()

│ ├── BubbleSort()

│ ├── DoNextSelectionSort()

│ ├── DoNextBubbleSort()

│ ├── GetSI()

│ ├── SwapStruct()

│ ├── ReadArray()

├── MergeSort.c

│ ├── DoFromTheEnd()

│ ├── IsToShow()

│ ├── DoNextMergeStep()

│ ├── UpColumn()

│ ├── DownColumn()

│ ├── DrawOneColumn()

│ ├── StressColumn()

│ ├── DrawAllTheColumns()

│ ├── ColorChange()

│ ├── ColorList()

│ ├── DoMergeSort()

│ ├── merge()

│ ├── merge\_sort()

├── Preparation.c

│ ├── ShowInstruction()

│ ├── ShowWelcomeTop()

│ ├── CleanDesktop()

│ ├── WhetherPauseChange()

│ ├── CleanNumbers()

│ ├── Is\_numberset()

│ ├── Is\_numbersave()

└── saves.dat(用于存档的二进制文件)

2）多文件构成机制

将所有外部变量归于ExternVariables.h文件中，在写的.c文件中将其包含

主文件Main.c的Main函数中调用InitGraphics.c文件中的InitWindowGraphics()函数完成多文件交互

## 函数设计描述

**Visualization.c**

**GetSI**

·函数原型

void GetSI(void);

·功能描述

由于画布有限，必须根据数据的值来确定柱状图的单位高度和单位宽度，防止图越界

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

首先得到最大数据，然后将此高度记为4.0，其余依比例而定

**InitAray**

·函数原型

void InitArray();

·功能描述

获得每个数据对应的柱状图的起始左下角坐标，并将数据根据类型转化成字符串存入结构数组中

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

首先获取窗口中心位置坐标，由根据数据集长度定义的柱状图单位宽度计算出第一个数据的柱状图左下角坐标，此后每个坐标为相应的前一个坐标加单位宽度

**Visualize**

·函数原型

void Visualize(void);

·功能描述

将初始化数据可视化为柱状图

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

由柱状图左下角坐标，用循环画出柱状图

**SwapStruct**

·函数原型

void SwapStruct(struct array \*x, struct array \*y);

·功能描述

在排序中交换两者的结构，从而同时交换数值、对应字符串、对应高度

·参数描述

x, y：两个指向要交换的两个结构变量的指针

·返回值描述

无

·重要局部变量定义

struct array temp;

·重要局部变量用途描述

用于交换两个结构的中间变量

·函数算法描述

简单的交换

**ReadArray**

·函数原型

void ReadArray();

·功能描述

用于读入数据

·参数描述

无

·返回值描述

无

·重要局部变量定义

FILE \*fp;

int mode = 0;

isInt; isFloat; isDouble;

char \*fname = NULL;

char num[7] = {0};

·重要局部变量用途描述

mode：判断是手动输入数据还是从文件读入数据

fp：文件指针

isInt, isFloat, isDouble：判断读入数据类型

fname：用于动态内存分配来记录要读入的文件名

num：记录读入的字符

·函数算法描述

用户输入mode值，若为1则从控制窗口读入，若为2则让用户再输入文件名，从文件读入。首先读入第一个空格前的字符，与字符串常量”int”,”float”,”double”比较来确定数据类型。

文件读入时，每次读到空格或换行符或文件末停止，将字符转化成数值。如果读到的是空格，则继续循环，否则结束循环。

**BubbleSort**

·函数原型

void BubbleSort(struct array a[], int len);

·功能描述

进行冒泡排序的可视化

·参数描述

a：要排序的结构数组

len：数组长度

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

在冒泡排序算法的嵌套循环中，加入柱状图移位的代码。具体执行：每次比较两数时，将柱状图变色以展现过程，并将上一轮变色的柱子涂回原色。将原本排序中比大小交换数值改为交换所属结构。利用Pause()函数展现出排序的每一步过程

**SelctionSort**

·函数原型

void SelectionSort(struct array a[], int len);

·功能描述

进行选择排序的可视化

·参数描述

a：要排序的结构数组

len：数组长度

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

与冒泡排序类似

**InitGraphics.c**

**InitWindowGraphics**

·函数原型

void InitWindowGraphics();

·功能描述

在画布上初始化整个图形界面

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

即用各种函数进行绘图

**DrawBox**

·函数原型

void DrawBox(double x, double y, double width, double height);

·功能描述

画一个长方形

·参数描述

x,y：画笔起始位置

width,hieght：长方形的长宽

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

画笔移至(x,y)，依次画四条线组成长方形

**RedisplaySortbuttons**

·函数原型

void RedisplaySortbuttons();

·功能描述

按下排序按钮时将按钮颜色变深

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

利用图形库的绘制

**MouseEventProcess**

·函数原型

void MouseEventProcess(int x, int y, int button, int event)

·功能描述

鼠标回调函数

·参数描述

x,y：鼠标位置坐标

button：鼠标按钮

event：鼠标按钮的动作（按下、提起等）

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

获取用户关于鼠标的事件，刷新页面显示。

**KeyboardEventProcess**

·函数原型

void KeyboardEventProcess(int key, int event)

·功能描述

键盘消息回调函数

·参数描述

key：键盘按键

event：键盘按键的动作（按下、抬起等）

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

获取用户关于键盘的事件，控制部分功能。

**is\_inbox**

·函数原型

int is\_inbox(double x,double y,int Bu);

·功能描述

判断鼠标点击位置是否在按钮内

·参数描述

x,y：鼠标位置坐标

Bu：button代号

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

比较鼠标点击位置和按钮区域来判断

**RedisplayTextStrings**

·函数原型

void RedisplayTextStrings(int n);

·功能描述

按钮上写上名称

·参数描述

n：

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

利用图形库函数画字符串

**MergeSort.c**

**DoFromTheEnd**

·函数原型

void DoFromTheEnd(int where);

·功能描述

直接到开头或直接到结尾

·参数描述

int where：判断是到开头还是到结尾的参数

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

where == 1，到开头，并将所有数据标为起始状态；where == 2，到结尾，并将所有数据标为已排序完成状态

**IsToShow**

·函数原型

int IsToShow();

·功能描述

判断是单步还是自动

·参数描述

无

·返回值描述

单步返回1，否则返回0

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

if判断几个参数值

**DoNextMergeStep**

·函数原型

void DoNextMergeStep();

·功能描述

单步执行归并排序函数

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

把判断单步执行变量设为1，调用DoMergeSort函数进行一次归并排序，然后把判断单步执行变量重置为0

**UpColumn**

·函数原型

void UpColumn(int order,double amount);

·功能描述

为体现归并排序特点，需要将选定柱移至下方再移至上方，此函数把第order个数移上去

·参数描述

order：要移的数的序号

amount：要移的数的数值

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

利用libgraphics库函数实现

**DownColumn**

·函数原型

void DownColumn(int begin,int order,double amount);

·功能描述

为体现归并排序特点，需要将选定柱移至下方再移至上方，此函数把第order个数移上去

·参数描述

begin：

order：

amount：

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

利用libgraphics库函数实现

**DrawOneColumn**

·函数原型

void DrawOneColumn(int num,int positionY,double amount,char \* colorname);

·功能描述

画一个柱

·参数描述

num：标号

positionY：

amount：数值

colorname：要画的颜色

·返回值描述

无

·重要局部变量定义

double showpercent;

int max=0;

·重要局部变量用途描述

showpercent：同GetSI函数，得到单位长度

max：用于找数值最大数的下标

·函数算法描述

根据下标选定起始位置、计算单位长度，画出柱子

**StressColumn**

·函数原型

void StressColumn(int num,double amount,char\* colorname);

·功能描述

调用DrawOneColumn函数

·参数描述

num：标号

amount：数值

colorname：颜色

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

调用DrawOneColumn函数，使positionY固定为1

**DrawAllTheColumns**

·函数原型

void DrawAllTheColumns(struct array a[]);

·功能描述

画出所有柱

·参数描述

a：保存数据的结构数组

·返回值描述

无

·重要局部变量定义

double leftlocaX,leftlocaY;

double showpercent;

·重要局部变量用途描述

showpercent：单位高度，同之前的SI函数

leftlocaX，leftlocaY：起始坐标

·函数算法描述

利用for循环对DrawOneColumn函数进行的扩展

**ColorChange**

·函数原型

double ColorChange(int n);

·功能描述

变换颜色

·参数描述

n：颜色参数

·返回值描述

return (double)n/255;返回颜色参数值

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

直接返回n/255

**ColorList**

·函数原型

char\* ColorList(int color);

·功能描述

颜色表

·参数描述

color：颜色参数

·返回值描述

return colorname;

·重要局部变量定义

static char colorname[3];

·重要局部变量用途描述

colorname：将整形值color转为字符串

·函数算法描述

利用graphics.c库文件中的DefineColor函数，根据输入的color参数输出相应的颜色参数

**DoMergeSort**

·函数原型

void DoMergeSort(struct array a[]);

·功能描述

调用merge\_sort函数进行归并排序

·参数描述

a：保存数据的结构数组

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

调用merge\_sort函数

**merge**

·函数原型

void merge(struct array arr[], int low, int mid, int high);

·功能描述

调用之前的函数进行归并排序可视化的操作

·参数描述

arr：保存数据的结构数组

low：归并排序中的最低位

mid：归并排序中的中间位

high：归并排序中的最高位

·返回值描述

无

·重要局部变量定义

double \*tmp = (double \*)malloc((high-low+1)\*sizeof(double));

·重要局部变量用途描述

tmp：申请空间，使其大小为两个

·函数算法描述

将归并排序每一步要排的一串柱下移，进行排序，再与上方比较，比较后上移至相应位置

**merge\_sort**

·函数原型

void merge\_sort(struct array arr[], unsigned int first, unsigned int last);

·功能描述

归并排序，同时每一次递归变一次色

·参数描述

arr：保存数据的结构数组

first,last：归并排序所需的参数

·返回值描述

利用return结束

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

归并排序算法，颜色参数自增来变色

**Preparation.c**

**ShowInstruction**

·函数原型

void ShowInstruction();

·功能描述

展现help弹窗

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

使用libgraphics库中的MessageBox函数

**ShowWelcomeTop**

·函数原型

void ShowWelcomeTop();

·功能描述

在起始界面画出“欢迎来到Visual Sorting”界面

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

重复画两次语句来实现粗体效果

**CleanDesktop**

·函数原型

void CleanDesktop();

·功能描述

清屏函数

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

先保存原先画笔颜色，然后用白笔erase，再切回保存的颜色

**WhetherPauseChange**

·函数原型

void WhetherPauseChange();

·功能描述

播放暂停键的切换

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

根据全局参数pause\_flag来确定是播放状态还是暂停状态，从而切换样式

**CleanNumbers**

·函数原型

void CleanNumbers();

·功能描述

将结构数组的数据全部清零（包括字符串）

·参数描述

无

·返回值描述

无

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

简单的for循环

**Is\_numberset**

·函数原型

int Is\_numberset();

·功能描述

判断结构数组中有无数据

·参数描述

无

·返回值描述

若有，返回1；不然，返回0

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

遍历判断有无

**Is\_numbersave**

·函数原型

int Is\_numbersave();

·功能描述

判断是否存过档

·参数描述

无

·返回值描述

存过，返回1；不然，返回0

·重要局部变量定义

无

·重要局部变量用途描述

无

·函数算法描述

逐个读入字符。如果开头就是空格，说明未存档，返回0。否则返回1。

# 部署运行和使用说明

## 编译安装

1. 打开文件“排序可视化.dev”，右键，选择打开方式：Dev C++
2. 按下F9键
3. 在同一文件夹中找到“排序可视化.exe”

## 运行测试

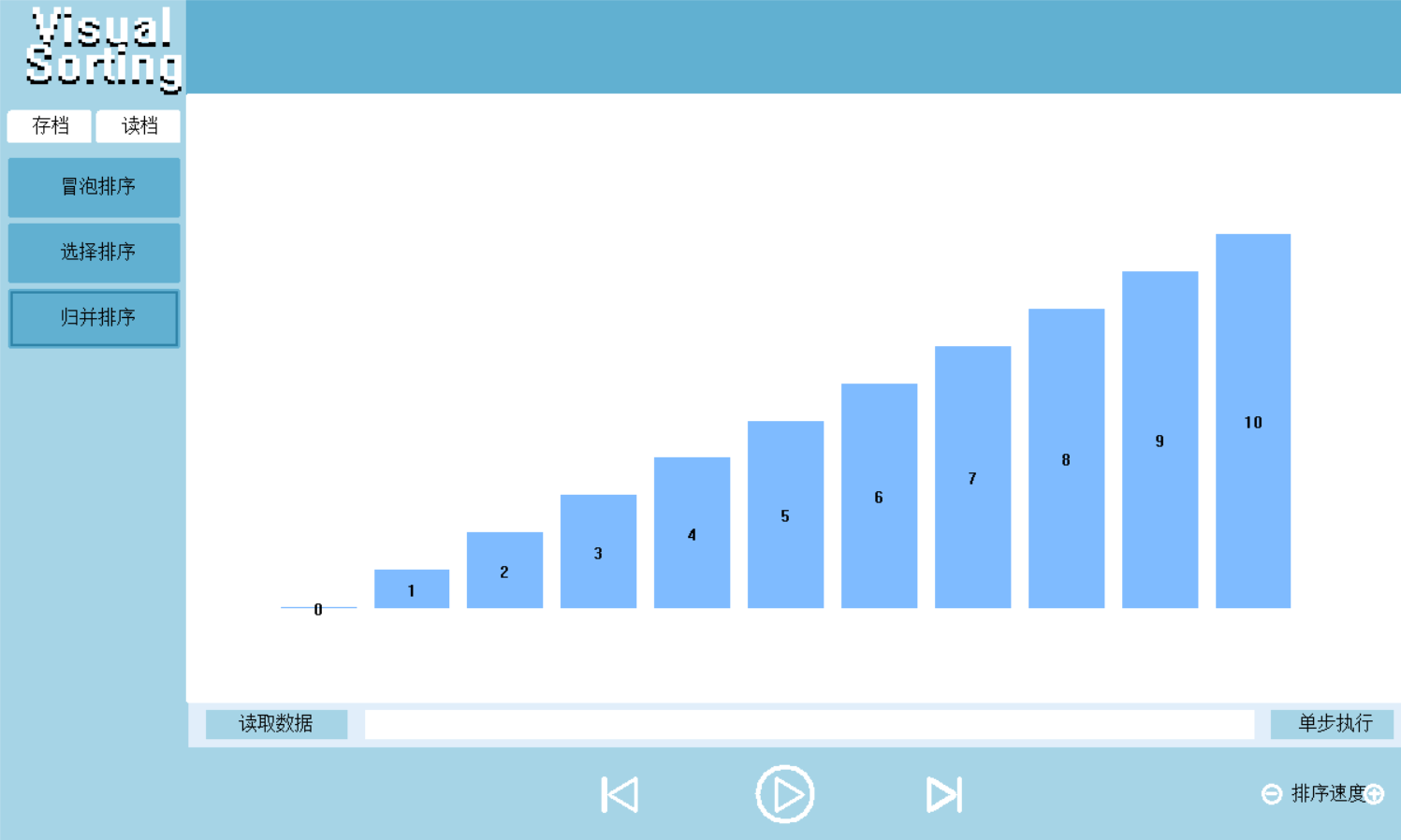
案例1：读取文件函数读入乱码

经过断点调试，我们发现函数内未考虑到文件数据结束后有换行符的情况。通过加入对换行符的判断，函数读入问题消除。

案例2：保留小数点问题

对于float和double类型，为了可视化的美观，我们决定在数值转换成字符串时保留小数点后二位。起初我们让字符串中小数点后第一位字符等于原数减去整数部分后乘以10再取整，即(int)((n – (int)n) \* 10) + 48,小数点后第二位同理。按照语法是完全正确的，然而在实际运行中，会出现4.30被转换成字符串4.29。通过使用其他ide我们排除了语法问题。进而我们在其它步骤的调试中发现Dev-C++存储数据类似于无限循环，即4.30会被存储为4.29999999，因此出错。于是我们决定干脆取数(int)((n – (int)n) \* 100)，然后让字符串中字符取这个十位数和个位数，成功解决问题。

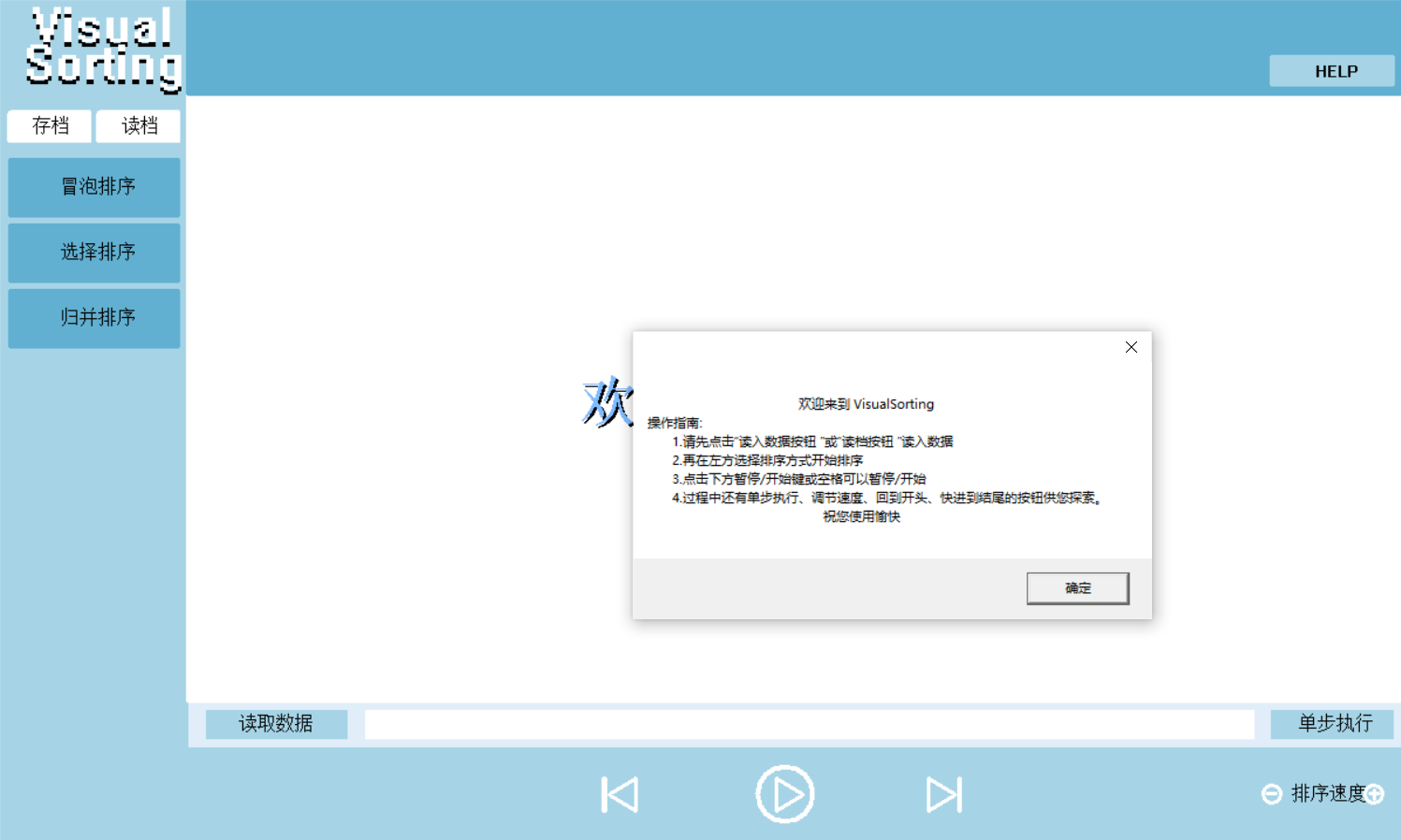
案例3：单步执行bug



起初单步执行按钮并无法实现预想中的功能。对此我们对代码进行了大修，对冒泡排序、选择排序也采取了计步的方法，并加入一些全局变量进行判断。最终实现了单步执行功能。

## 使用操作

1. 在4.1编译安装后，打开得到的**“排序可视化.exe”**文件，出现如下图所示



1. 弹窗部分有简易的操作指南，之后随时点**“HELP”**键都可出现

要想输入数据，有位于左上的**“读档”**按钮和位于右下的**“读取数据”**按钮可供选择



点击“读取数据”出现控制台

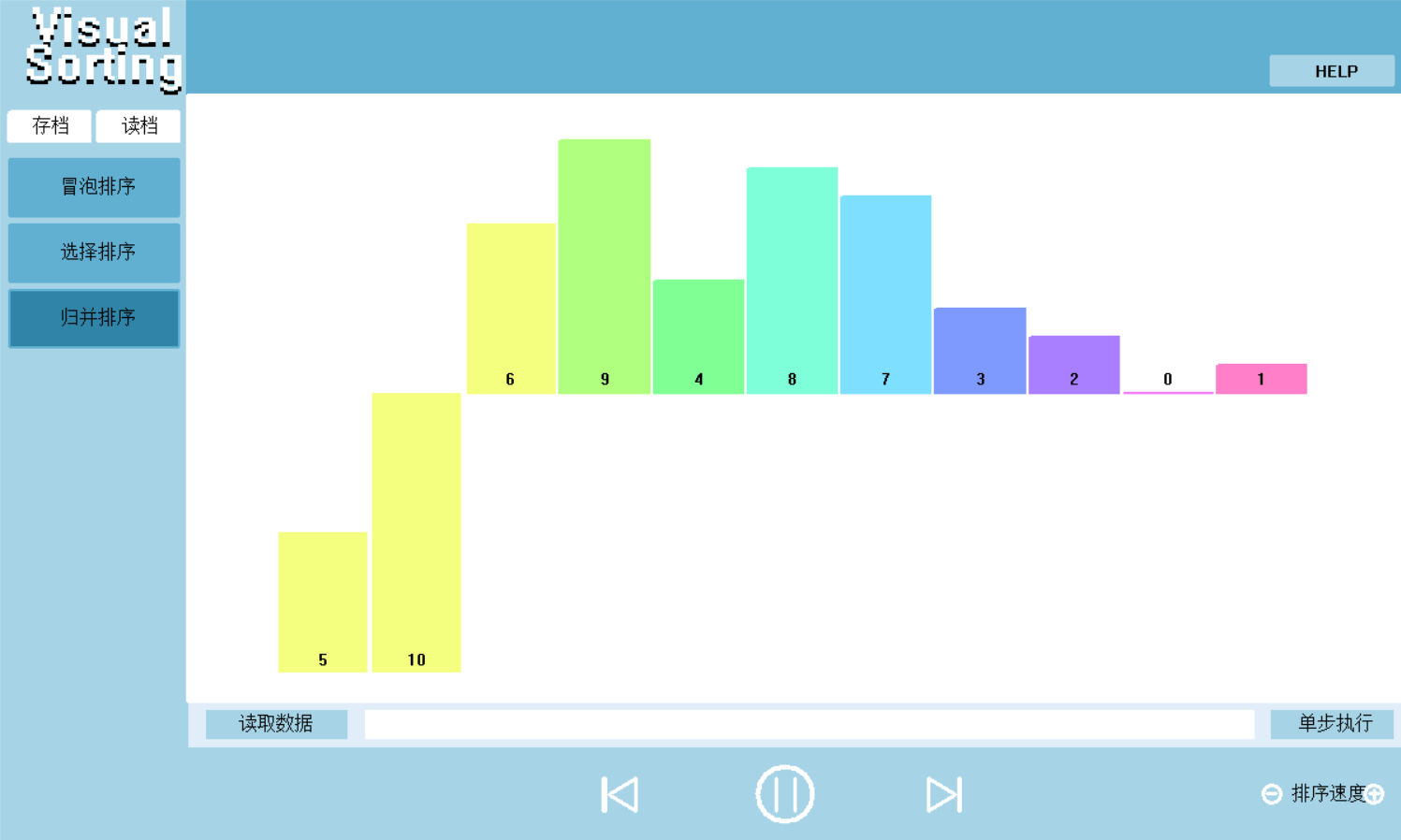
控制台内根据提示信息进行即可**（注：读取的文件内容必须符合格式，格式见testfiles中的测试数据）**



点击“读档”，程序会自动读入保有存档信息的二进制文件

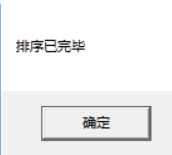
若要存档，在有数据时可以随时点击**“读档”**旁的**“存档”**按钮进行存档

1. 输入数据后，可以看到左边分别有**“冒泡排序”**、**“选择排序”**和**“归并排序”**按钮，点击想要程序呈现的即可



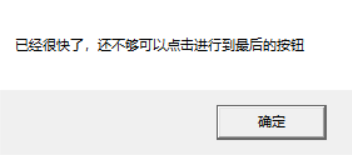
如图，点击归并排序，程序开始自动执行

开始后，如果想要手动一步步执行，点击右下角的**“单步执行”**按钮，每点一次，排序走一步，直到排序结束



排序结束时会有弹窗提醒，

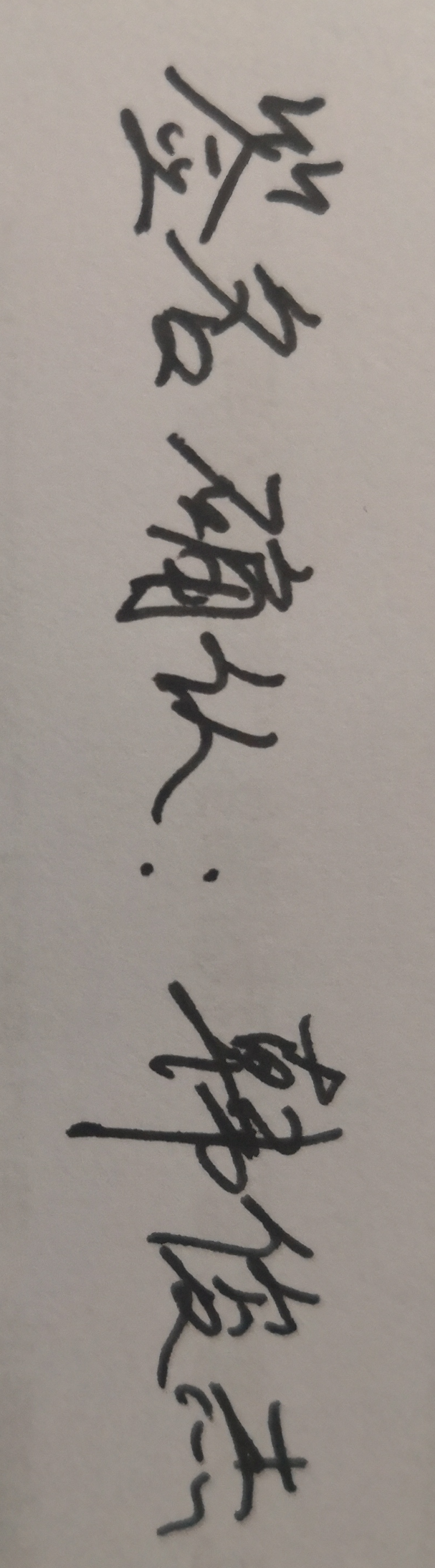
1. 最下方正中间三个按钮从左到右分别为：**“回归初始状态”**、**“暂停/开始”**、**“直接排序完毕（进行到最后）”**，根据需要点击即可**（注：单步执行时不可点击最下方三个按钮）**
2. 右下角排序速度两侧的“-”和“+”可以改变自动排序的执行速度



速度加到太快会出现如图所示的弹窗，请点击确定将所有弹窗关闭再继续程序运行

# 团队合作

## 任务分工



**匿名版略去**

## 开发计划

5.9-5.12 选题+分工

5.12-5.25 初步完成项目雏形

5.25-6.9 基本完成

6.9-6.18 全部完成

## 编码规范

·文件统一采用 UTF-8 with BOM 编码，LF 换行

·使用C89规范

代码风格参考谷歌的 C++ 风格指南（[中文版](https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/contents/)/[英文版](https://google.github.io/styleguide/cppguide.html)）（删去其中C++特性的部分）。下面节选部分内容展示，具体内容请前往上述网站

* **不允许的行为**: 不要描述显而易见的现象, *永远不要*用自然语言翻译代码作为注释, 除非即使对深入理解 C++ 的读者来说代码的行为都是不明显的。要假设读代码的人 C++ 水平比你高, 即便他/她可能不知道你的用意: 你所提供的注释应当解释代码**为什么**要这么做和代码的目的, 或者最好是让代码自文档化
* 类型名称的每个单词首字母均大写, 不包含下划线。变量 (包括函数参数) 和数据成员名一律小写, 单词之间用下划线连接。枚举的命名应当和常量或宏一致: kEnumName 或是 ENUM\_NAME
* 尽量不使用非 ASCII 字符, 使用时必须使用 UTF-8 编码
* 只使用空格, 每次缩进 2 个空格
* 函数调用如果同一行放不下, 可断为多行, 后面每一行都和第一个实参对齐, 左圆括号后和右圆括号前不要留空格

## 合作总结

### 开发亮点

界面十分美观

使用windows.h头文件内的函数来实现部分操作

在图形库中修改参数从而能够绘制出更多种类的颜色

功能较多，考虑范围广，各项功能可以完美无bug嵌合

### 挑战点

单步执行的实现比较困难，传统的死循环暂停循环在图形交互中无法使用

多人合作难免会出现理解错误，导致加入函数会出现一些定义之类的小bug

### 应用知识点

文本/二进制文件读写: 支持文件读取、保存

图形库函数：整个 UI 都是通过 libgraphics 库进行设计的，也用到了鼠标消息回调函数和键盘消息回调函数

指针：部分函数应用了指针

结构数组：数据用结构数组保存（由于对数据的要求以查找为重，本学期所学的链表显得有点多余，故不予以使用）

动态内存分配：一些地方利用动态内存分配代替数组，节约了空间

### 讨论记录

基本上大部分内容都是线下讨论一起完成的，所以线上记录并不多

**匿名版略去**

## 收获感言

**匿名版略去**

# 参考文献资料

https://visualgo.net/en/sorting?slide=1