

浙江大学

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： 刘逸洋 学号： 200104898 电话： 15757168776
2. 姓名： 吴创 学号： 3200102123 电话： 19883143610
3. 姓名： 章仁翔 学号： 3200104748 电话： 17300989171

指导老师： 张引、侯宇轩

2020~2021 春夏学期 2021 年 6 月 15 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
1.1	选题背景及意义	4
1.2	目标要求	4
1.3	术语说明	4
2	需求分析	5
2.1	业务需求	5
2.2	功能需求	5
2.3	数据需求	5
3	程序开发设计	6
3.1	总体架构设计	6
3.2	功能模块设计	6
3.3	数据结构设计	7
3.4	源代码文件组织设计	7
3.5	函数设计描述	10
4	部署运行和使用说明	18
4.1	编译安装	18
4.2	运行测试	18
4.3	使用操作	19
5	团队合作	21
5.1	任务分工	21
5.2	开发计划	21
5.3	编码规范	22
5.4	合作总结	22
5.5	收获感言	22
6	参考文献资料	24

排序算法可视化大程序设计项目

1 大程序简介

1.1 选题背景及意义

在上个学期的 C 语言的学习中，我们基本掌握了简单程序的写法，也了解了一些排序的原理和算法。在这个学期中，我们的学习更进一步，了解了图形编程，并且有能力实现简单的动画和一些简单的图形绘制等操作，在此基础上，老师就给我们布置了一个大程序的作业，里面包含三个程序，我们小组就选择了其中实现起来相对简单一点的“排序算法可视化”，其意义便是希望利用动态展示，让一些没有编程经验的初学者也能快速的明白一些排序的原理，并且必要时，也可以将该程序当作排序的工具，为使用者服务。

1.2 目标要求

我们的程序主体包含三个排序算法，分别是冒泡排序、选择排序、归并排序，我们的目标不仅是要实现三种排序的动态演示，具体而言还要能够自动执行和单步执行、能够从文件中读取待排序的数据、支持过程的存档和调取、实时添加或删除数据并进行修改数据后的新排序。

1.3 术语说明

该程序会包含许多头文件，其中最为重要的就是 `libgraphics` 和 `simpleGUI` 里面的相关头文件，除此之外还有我们小组自己定义的一些头文件，同时我们除了调用了库函数，还编写了许多供我们自己使用的函数，以及定义了各种各样的全

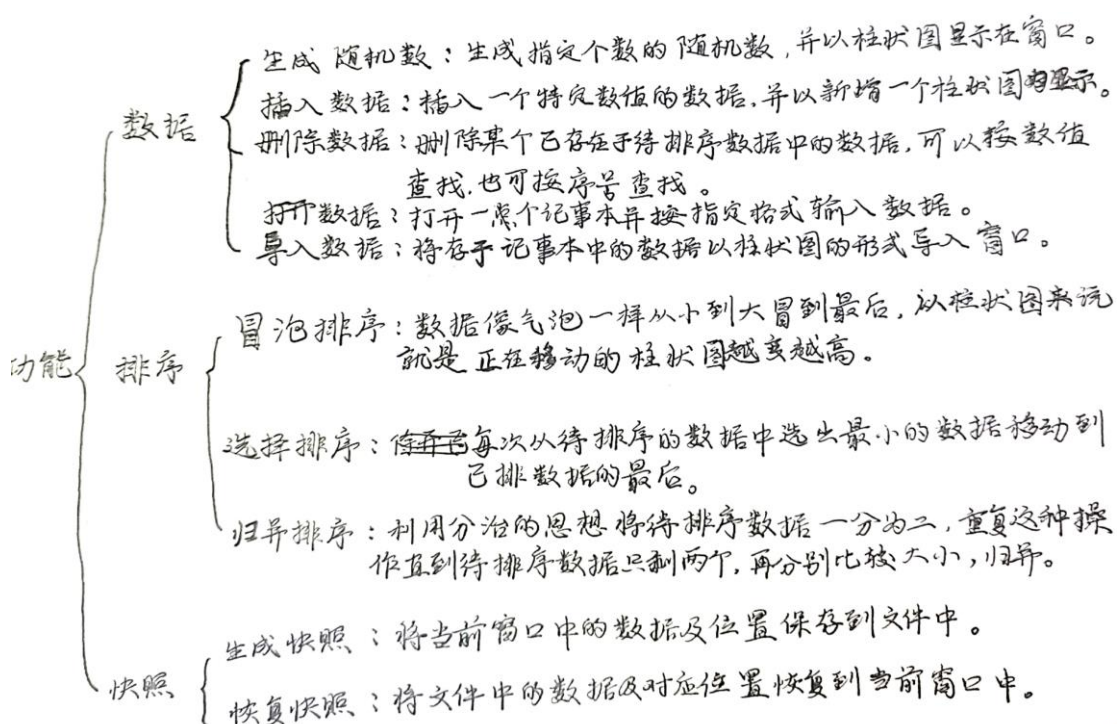
局变量，都是为方便我们程序的编写而服务的，具体内容将在接下来的板块中提及。

2 需求分析

2.1 业务需求

此程序旨在用于教学演示冒泡排序、选择排序、归并排序三种排序方式的排序原理。使用者多为信息技术课程教师。

2.2 功能需求



2.3 数据需求

被排序的数据必须是数，不能是字符，支持 int 型，float 型和 double 型，同时要能够从文件中读入被排序的数据，数据格式为<类型><数据表>，数据之间

用空格隔开，还应支持过程的存档和调取，将必要的数据保存在二进制文件中，可读入二进制文件中保存的数据，继续排序算法的演示。

3 程序开发设计

3.1 总体架构设计

总体
架构

全局变量及数组：double num[100]、int location[100]、
double x-histogram[100]、int preLocation[100]
double winwidth, winheight; int count;
int enable-editing=1, modelID=0;
int bubble-flag, select-flag, Merge-flag;
int finished; int movingTimer, moving-finished;
int data-status=0; double interval;
int firstEnterRanking; int fuc=0; int iMax=1;
int restart-bubble=0, restart-select=0, n-Merge=1;

文件：main.c、main.h、our-histogram.h、our-histogram.c、our-option.h、
our-option.c、our-data.h、our-data.c

函数：int Main(void)、void CharEventProcess(char ch)、
void KeyboardEventProcess(int key, int event)
void MouseEventProcess(int x, int y, int button, int event)
void TimerEventProcess(int timerID)
void display(void)、void setLocation(void)、void drawHistogram()
void drawButtons()、void rankBubble()、void rankSelect()、
void rankMerge()、void saveData()、void restoreData()、
void openData()。

功能表

1. 数据编辑：①数据的插入与删除、②数据的批量导入与导出。
2. 排序可视化：①单步执行、②自动单步执行、③全部自动执行。
3. 其他：存档与调取(二进制文件)

3.2 功能模块设计

数据模块：生成随机数据、插入数据、删除数据、清空数据、打开数据、导入数据。

排序模块：冒泡排序、选择排序、归并排序。

过程存取与调度模块：生成快照、恢复快照。

3.3 数据结构设计

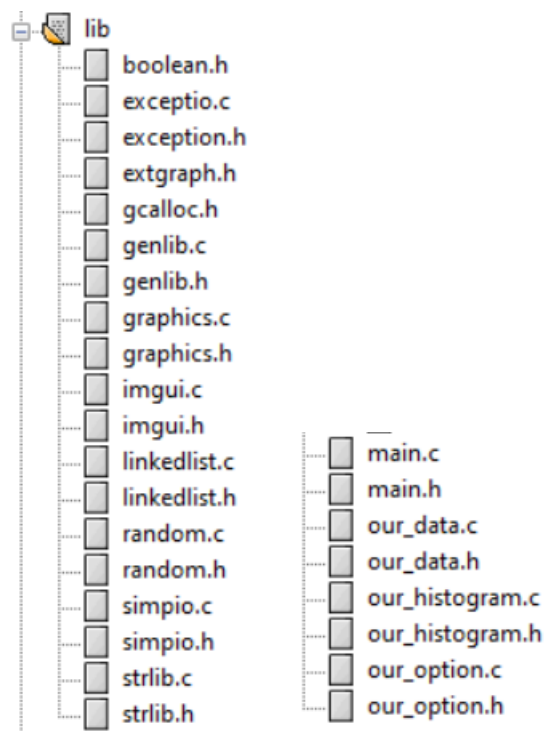
在数据结构方面，我们主要运用了数组这一数据结构来为我们的程序服务。

比如表示存储具体数值的 `num[100]` 数组和表示 `num[100]` 数组中数据位置的 `location[100]` 数组，可以帮助我们单次执行的排序函数作数据之间的交换，即柱状图之间的交换。

3.4 源代码文件组织设计

1) 文件函数结构

程序包含文件：



由于 `lib` 文件夹内的各文件中内容是固定的，这里就不再赘述。接下来就介

绍我们自己编写的文件里的内容。

main.c:

清屏函数、计时器启动函数、用户的显示函数、用户的字符事件响应函数、用户的键盘事件响应函数、用户的鼠标事件响应函数、用户的计时器事件响应函数、用户主程序入口函数。

main.h:

```
// 全局变量
extern double winwidth, winheight; // 窗口尺寸

extern double num[100]; // 存储需要排序的数据, 运行过程中数值不交换;
extern int location[100]; // 对应存储数据当前所在的位置, 运行过程中数值交换;
extern int count; // 数据个数;
extern int enable_editing; // 判断是否允许操作 (正在处理数据... ..);
```

our_data.c:

导出二进制文件函数、导入二进制文件函数、打开 txt 数据库函数。

our_data.h:

```
void saveData(void);

void restoreData(void);

|
void openData(void);
```

our_histogram.c:

初始化函数、设定位置函数、画遮罩函数、柱状图移动函数、添加数据按钮函数、最大值函数、画柱状图函数。

our_histogram.h:

```
void moveHistogram(void);

void setLocation(void);

void drawHistogram(void);
```

our_option.c:

画菜单函数、画按钮函数、冒泡排序函数、选择排序函数、归并排序函数。

our_option.h:

```
void drawMenu(void);  
void drawButtons(void);  
  
int rankBubble(void);  
int rankSelect(void);  
|  
int rankMerge(void);
```

2) 多文件构成机制

外部变量:

在 our_data.c、our_histogram.c、our_option.c 中均利用 extern 调用了在 main.c 中定义的 num[100]、location[100]、count、enable_editing。

在 main.c 中调用了 our_option.c 中的 bubble_flag、finished、Select_flag、Merge_flag，和 our_histogram.c 中的 x_histogram[100]、movingTimer、moving_finished。

在 our_histogram.c 中还调用了 main.c 中的 winwidth、winheight、modelID，和 our_option.c 中的 firstEnterRanking、bubble_flag、Select_flag、Merge_flag。

在 our_option.c 中还调用了 main.c 中的 our_histogram.c 中的 winwidth、winheight、modelID，和 our_histogram.c 中的 movingTimer。

外部函数:

在 main.c 中直接声明调用了 libgraphics 中的 void DisplayClear(void)、和 void startTimer(int id,int timeinterval)，还利用 include 引用了 main.h、our_option.h、our_histogram.h 这几个头文件来调用他们里面定义的函数。

在 our_data.c 中引用了 main.h、our_option.h、our_histogram.h。

在 our_histogram.c 中引用了 main.h、our_option.h。

在 `our_option.c` 中引用了 `main.h`、`our_histogram.h`、`our_data.h`。

Define 保护机制：

所有头文件均添加 `define` 保护机制。

3.5 函数设计描述

`main.c:`

所含文件中的函数所有大程大致统一，在此不过多赘述，仅展示 `void display(void)` 函数所含子函数的具体功能，如下。

```
void display(void)
{
    DisplayClear();           //清屏函数，库中定义

    drawBackground();         //画灰色背景函数，main.c 中定义

    drawButtons();            //画控件定义函数，our_option.c 中定义

    drawHistogram();          //画柱状图函数，out_histogram.c 中定义
}
```

`our_data.c:`

1. 生成二进制快照文件函数

函数原型：	<code>void saveData(void)</code>
功能描述：	生成二进制快照文件
重要局部变量用途描述：	文件指针 <code>output</code>
函数算法描述：	使用文件指针 <code>output</code> 用只写方式打开/新建目录下的二进制文件，将 <code>count</code> 、 <code>num</code> 、 <code>location</code> 等全局变量依次用 <code>fprintf</code> 存入文件中。存入数据完毕后关闭文件。

2. 恢复二进制快照文件函数

函数原型:	void restoreData(void)
功能描述:	恢复二进制快照文件
重要局部变量用途描述:	文件指针 input
函数算法描述:	使用文件指针 input 用只读方式打开目录下的二进制文件，从文件中依次用 fscanf 将 count、num、location 等全局变量读取。读取数据完毕后关闭文件。

3. 导入 txt 文件中的数据函数

函数原型:	void openData (void)
功能描述:	导入 txt 文件中的数据
重要局部变量用途描述:	文件指针 fp
函数算法描述:	使用文件指针 fp 用只读方式打开目录下的二进制文件，从文件中依次用 fscanf 读取字符串，在遇到空格后暂停，在遇到文件末尾后停止。第一个字符串读取字符的数据类型，其余的字符串读取数据。

our_histogram.c:

1. 初始化函数

函数原型:	static void initialization(void)
功能描述:	进入排序功能的初始化
重要全局变量定义:	firstEnterRanking 第一次进入排序的标记 preLocation[]记录上次一每个数据的序号
重要全局变量用途描述:	第一次进入排序 (firstEnterRanking==1) 则 将

	firstEnterRanking=1, 且将 preLocation[]依次赋值为 location[]的值。
函数算法描述:	同上

2. 画遮罩函数

函数原型:	static void drawHider(void)
功能描述:	在排序动画进行时在控件区域画上遮罩, 禁止用户继续操作
函数算法描述:	<p>1. drawBox(0, winheight-2.8, winwidth, 2.8, 1, "数据处理中...", 'C', "wordBlack")</p> <p>2. button(GenUIID(0), 5, winheight-1.6, w, h, "停止")</p>

3. 移动柱状图横坐标函数


函数原型:	static void moveHistogram(void)
功能描述:	排序动态展示界面下的柱状图坐标
重要全局变量定义:	<p>movingTimer 计时器</p> <p>x_histogram[] 每个数据对应的横坐标</p>
重要全局变量用途描述:	movingTimer 为 int 型, 大小总 0 到 100, 当程序检测到 location 的值发生了变化, movingTimer 使柱状图的横坐标 x_histogram[]能够缓慢地从原来的位置变化到指定位置。
函数算法描述:	movingTimer 计时器在 main 文件中随着计时器的触发而每次加一。若 movingTimer<100, 则 x_histogram[]根据 movingTimer 的大小确定它在原坐标与新坐标之

	间的比例；若 <code>movingTimer<100</code> ，则 <code>x_histogram[]</code> 根据最新的 <code>location</code> 确定 <code>x_histogram[]</code> 。
--	--

4. 设置柱状图的横坐标函数

函数原型：	<code>static void setLocation(void)</code>
功能描述：	若 <code>modelID==0</code> 则确定 <code>x_histogram[]</code> 值，若 <code>modelID==1</code> 则使用 <code>static void moveHistogram(void)</code> 函数
函数算法描述：	if 分支结构实现

5. 添加数据按钮函数

函数原型：	<code>static void AddNumButton(void)</code>
功能描述：	在数据编辑界面添加数据按钮 
重要局部变量定义：	<code>r h</code>
重要局部变量用途描述：	<code>r</code> 圆的半径 <code>h</code> 中心位置相对于窗口底端的距离
函数算法描述：	使用： <code>SetPenSize(5);</code> <code>SetPenColor("Cyan");</code> <code>DrawArc(r,0,360);</code> <code>SetPenColor("Dark Gray");</code> <code>MovePen(interval*(count+1)-0.45*r,h+r);</code> <code>DrawLine(0.9*r,0);</code>

	等函数达到理想的效果。
--	-------------

6. 最大值函数

函数原型:	static double maxnum(void)
功能描述:	计算 count 个数据中的最大值
返回值描述:	double 返回最大值
函数算法描述:	for 循环

7. 画柱状图函数

函数原型:	void drawHistogram(void)
功能描述:	根据前面的函数获得的柱状图横坐标, 和 num 与最大值的相对大小, 画出柱状图
重要局部变量定义:	<pre>double histogram_width=interval*0.4;//柱状图的宽度 double hh=5.3;//柱形图高度比例系数 double lowheight=0.25;//柱形图基准线的高度 char* color[]={ "Orange","Blue","Dark Gray","Magenta","Red","Yellow", "Cyan","Green"};//便于后续调用不同颜色</pre>
函数算法描述:	<pre>for 循环 drawRectangle(...); drawLabel(...);</pre>

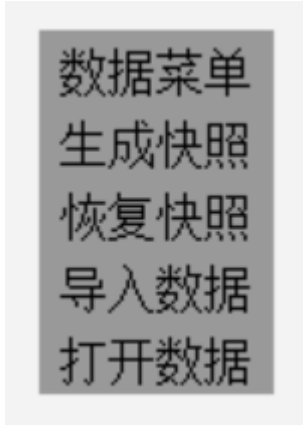
our_option.c:

1. 重新启动排序函数


函数原型:	void restart(void)
-------	--------------------

功能描述:	将排序相关的全局变量变回初始值
函数算法描述:	若干个赋值运算

2. 菜单函数

函数原型:	void drawMenu()
功能描述:	<p>画出数据导入导出模块的菜单</p> 
重要局部变量定义:	<pre>static char * menuListAbout[] = {"数据菜单", "生成快照", "恢复快照", "导入数据", "打开数据" }; int selection;</pre>
重要局部变量用途描述:	前者用于集中存储数据菜单中的文字, selection 判断点击的具体功能
函数算法描述:	使用 menuList 函数, 返回值赋给 selection。再用 selection 判断具体需要执行哪一项操作。

3. 按钮函数

函数原型:	void drawButtons(void)
功能描述:	<p>画控件、实现控件的功能，具体功能请参考下图</p> 
重要局部变量定义:	<pre>double fH = GetFontHeight(); double h = fH*2; // 控件的高度 double x = winwidth/2.5; double y = winheight/2-h; double w = winwidth/7; // 控件的宽度</pre>
函数算法描述:	if 分支结构

4. 单步冒泡函数

函数原型:	int rankBubble(void)
功能描述:	实现单次点击后的单步冒泡排序
参数描述:	无参数
返回值描述:	无需返回值，以全局变量 finished 判断排序是否结束
重要全局变量定义:	<pre>int i_rankBubble=0; int j_rankBubble=0; int restart_rankBubble=1; int func=0;</pre>
重要全局变量用途描述:	<pre>int i_rankBubble=0; int j_rankBubble=0;//记录冒泡排序当前位置 int restart_rankBubble=1;//判断冒泡排序是否需要重新 开始</pre>

	<code>int func=0; //标记目前正在使用何种排序方式</code>
函数算法描述:	调用一次函数后，从原有的 <code>i_rankBubble=0</code> 和 <code>j_rankBubble</code> 中读取数据，对指向的两个数据进行比较和交换，并更改 <code>i_rankBubble=0</code> 和 <code>j_rankBubble</code> 的值。当排序完成是，将 <code>finished</code> 赋值为 1。

5. 单步选择函数

函数原型:	<code>int rankSelect(void)</code>
功能描述:	实现单次点击后的单步选择排序
参数描述:	无参数
返回值描述:	无需返回值，以全局变量 <code>finished</code> 判断排序是否结束
重要局部变量定义:	<code>int i, k, min, p1=0, p2=0;</code>
重要局部变量用途描述:	执行 <code>num[]</code> 数组的比较，同时进行 <code>location[]</code> 的暂存和交换
函数算法描述:	单步执行的思路：调用一次函数，定位到需比较的第一个数组位置，同时遍历所有未比较数组值，进行逐一比较，最后执行单步交换。

6. 单步递归函数

函数原型:	<code>int rankMerge(void)</code>
功能描述:	实现单次点击后的二路归并排序
参数描述:	无参数
返回值描述:	无需返回值，以全局变量 <code>finished</code> 的值来判断是否排序结束

重要局部变量定义:	<code>int temk = 0;int p1=0,p2=0;int mid=(low+high)/2;</code>
重要局部变量用途描述:	记录当前所指定数据的位置，记录比较次数
函数算法描述:	利用分治法和递归的思想，将数据分治处理后再归并起来

【注:】单步递归函数由递归函数改编而来，通过判断递归调用的层数来实现每次用函数只排序一次，达到单步执行的效果。

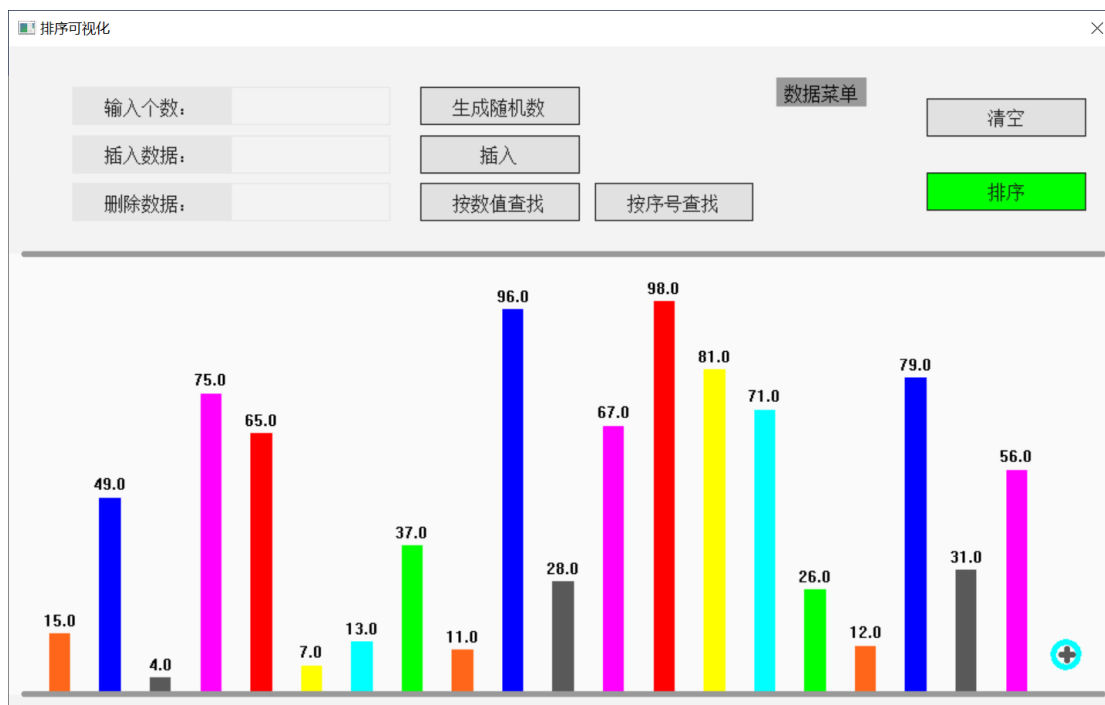
4 部署运行和使用说明

4.1 编译安装

如上文所示，压缩包解压后，打开文件夹，打开 `tutorialsDevC` 文件夹，打开 `0-emptywindow` 文件夹，打开 `step0.dev` 工程文件，选择适合电脑配置的编译器以后，选择“运行”-“全部重新编译”，编译完成后，选择“运行”，即可成功运行。

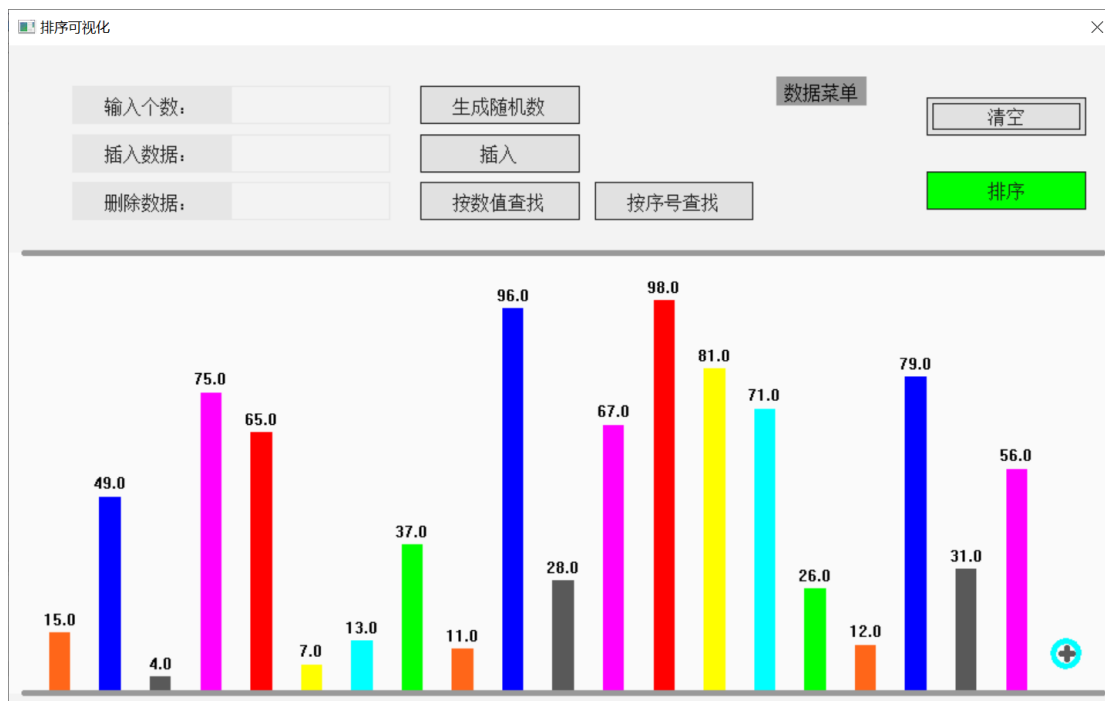
4.2 运行测试

运行程序后，出现以下界面，表示程序启动正常。测试各个程序功能，若均无误，则程序正确。



4.3 使用操作

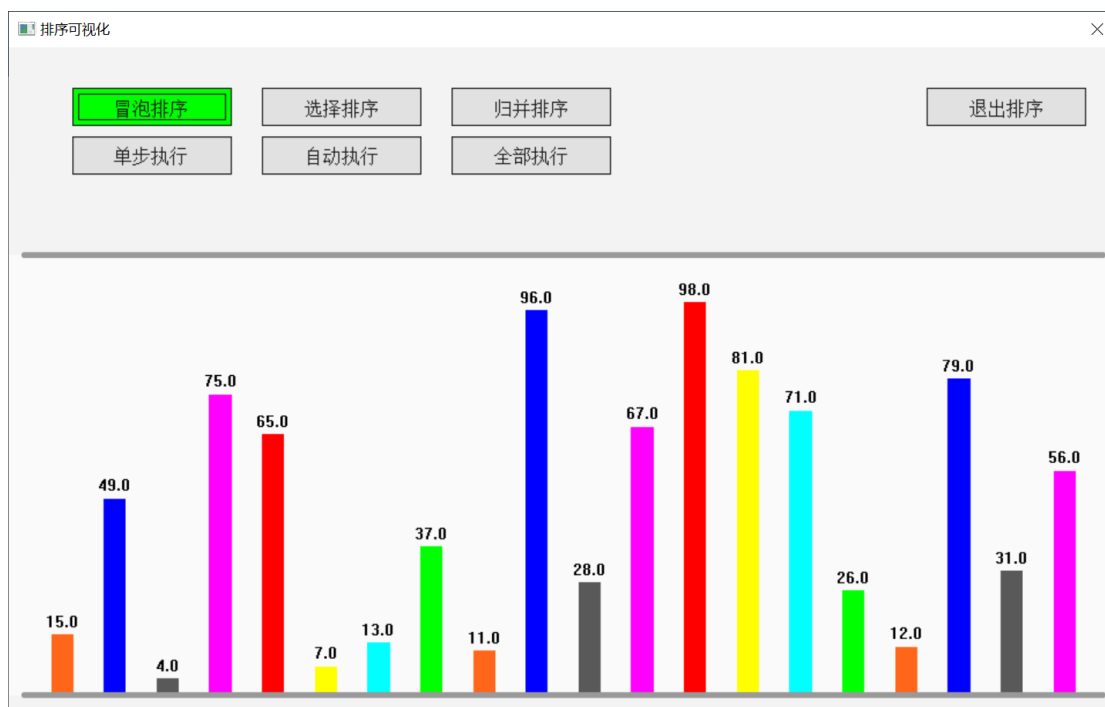
打开 exe 文件后，调出数据编辑界面。



默认生成 20 个随机数并且生成柱状图。首先，从左上角开始介绍。第一个功能是生成指定个数的随机数，可以输入数据数目并且进行一键生成。第二个功

能是在柱状图末尾添加新的指定数据，输入后即可添加。第三个功能是删除指定数据。目前实现了两个不同的删除方法：可以输入数据的值，删除指定数值的数据；也可以输入位次，删除指定位置的数据。

下面来介绍一下右上角。在菜单栏里，生成快照功能可以暂时保存当前数据，使程序重启后还可以通过恢复快照功能快速恢复数据；导出数据功能可以打开一个指定名字的 txt 文件并且写入数据类型和所有数据数值；导入数据功能则可以基于这个 txt 文件进行数据导入。右上角的清除按钮可以一键清除所有数据；而排序按钮可以加入动态排序展示界面。



进入动态排序展示界面后，数据不可修改，只可排序。我们实现了三种排序方式：冒泡排序、选择排序和归并排序。每个排序都实现了单步执行、自动执行和全部执行功能。单步执行是指，进行一次动态排序操作；自动执行是指，自动进行所有动态排序操作；归并排序是指，一次性完成所有排序，显示最终结果。当动态排序进行时，界面会显示“排序中”并且禁止进行操作，该状态可以通过取消按钮退出。排序完成后会显示“排序已完成”提示。在该界面下，可以随时

通过退出排序按钮回到数据编辑界面，当前数据全部保留。

5 团队合作

5.1 任务分工

我们对任务进行按函数和对应功能分工。

刘逸洋同学负责冒泡排序的具体实现、数据编辑功能和排序按钮的控制和响应的具体实现、柱状图画图功能的具体实现。

吴创同学负责归并排序的具体实现、程序参数设置与调试。

章任翔同学负责选择排序的具体实现、数据的批量导入功能以及二进制文件的导入导出功能的具体实现。

在实际编写中，部分任务由多位组员合作完成，并非完全独立。

5.2 开发计划

我们于第一次讨论时制订了开发计划。开发计划如下。

4-16 立项；

4-17 编写预备和知识复习开始；

5-8 进行多次讨论后最终确定选题；

5-14 统一线下会议，最终讨论通过了分工表和函数功能表；知识准备阶段结束，编程第一阶段启动；

5-19 主程序以及所有函数功能文件搭建完毕，同时同步了第一阶段进度；

5-20 统一线上会议，规范统一了编写所使用的全局变量；

5-21 统一线下会议，稍晚时候又进行了统一线上会议，对第一阶段的高难度

点进行集体攻关和讨论；

5-27 第一阶段编译通过，模拟运行达到功能要求，开始编写第二阶段；

5-28 统一线下会议，确定了第二阶段编写任务和具体时限；

5-29 程序代码进行优化，解决了部分已知 bug。

6-2 程序代码继续优化，解决部分已知 bug。

6-6 统一线下会议，第二阶段程序重难点攻关讨论，重新对任务进行分配。

6-11 所有代码编写完成并且通过编译，模拟运行达到功能要求。

5.3 编码规范

我们采用的是统一的 C 语言编程规范。

5.4 合作总结

2021 年 4 月 21 日：确定文件、函数、全局变量、分工、开始各自编写排序函数

2021 年 5 月 15 日第二次研讨 发现对于变量的理解有误 组长解释 组员修改程序

2021 年 6 月 1 日第三次研讨 归并函数集体攻关

2021 年 6 月 14 日：展示 ppt 视频录制

5.5 收获感言

吴创：

在整个大程序的合作过程中，我体会到了团队合作的不容易和其重要性，比如和队员之间的配合沟通在刚开始的时候可能会出现问题，但当你遇到一个人无

法解决的问题时，三个人一起可能就能解决这个问题。在整个过程中，我更加清楚地认识到自己本学期的学习状况，发现了很多知识盲区，也体会到了自己的无力，我还认识到实践才能出真知，课堂上讲的很多理论知识只是听一听的话肯定是不行的，要舍得花时间在课后自己实践一番才能真正掌握。整个大程序的编写中，我所做的工作是最少的，因为自己所掌握的知识实在太少，很多编写工作都不能胜任，对此我也心感愧疚，但又无可奈何，在这里也是非常感谢两位组员对我的关照，尤其是组长，为我们的这个项目花费了很多心思，不仅要完成自己的工作，还要帮我们善后，感谢所有队员的付出，才换来了这个项目的成功。

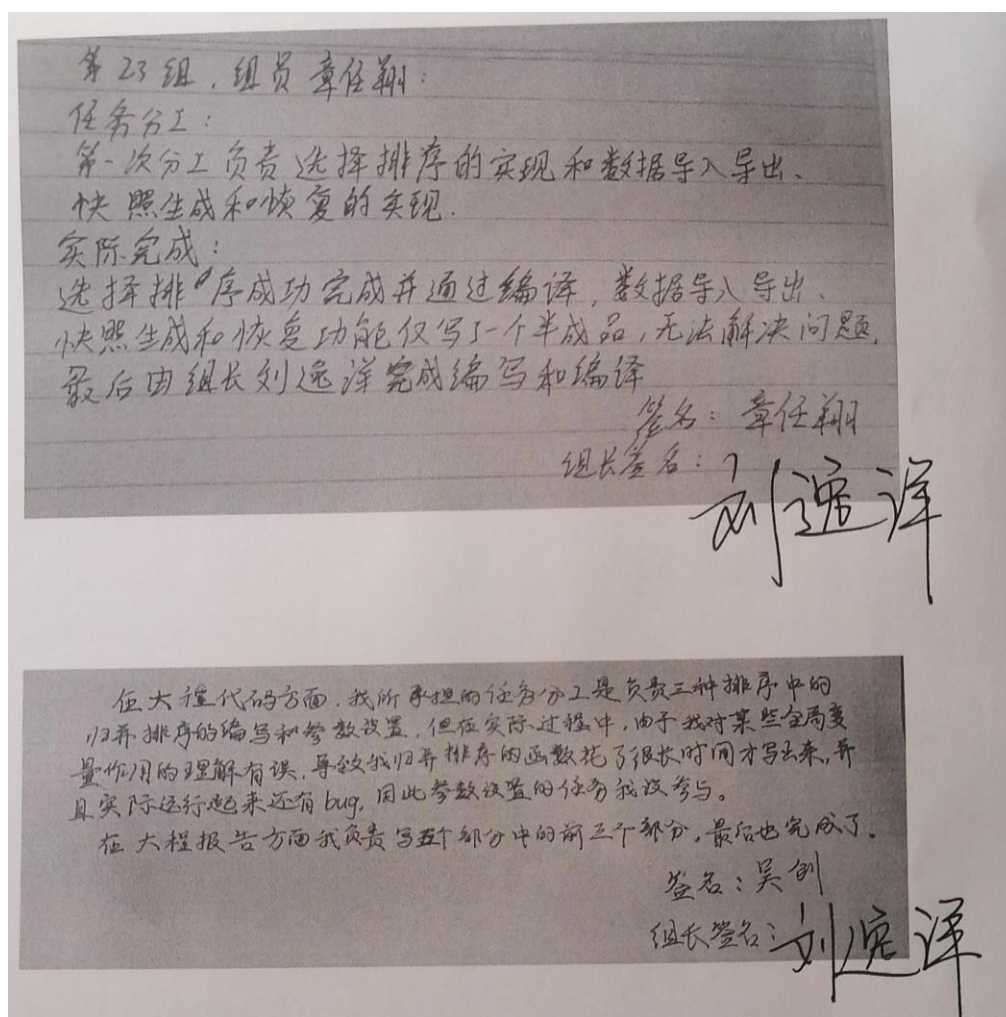
刘逸洋：

“纸上得来终觉浅，绝知此事要躬行。”课本上的知识到实际的运用还有具有非常大的差距。虽然我 C 语言的编程已经掌握了一定的基础，但当引入一个全新的库，对我而言还是具有挑战的。编程的时间有限，需要快速地熟悉运用库中的函数，我采用了两种方式：第一种方式是阅读开源的原函数，这种方式可以比较全面的了解到库中包含的函数，但是这种方式因为短时间输入的信息过大，相对重要的函数容易遗忘；第二种方式是阅读现有的例程，这种方式可以较快的获取到重要函数的用法，但是对于获取的函数具有局限。例如，在这次 C 大程的编写，我通过看例程快速地学会一个函数的使用，而当例程中获得的知识受限之后，我还会遍历原函数，来针对性地获取知识。总之，这次的程序设计任务开阔了我的计算机思维、锻炼了我的团队合作能力和交流能力，收获颇多！

章任翔：

首先感谢组员吴创同学和组长刘逸洋同学的帮助和支持。在整个程序编写过程中，很多内容由于需要和其他人做的内容对接，所以会出现对接不上导致自己

所在任务难以推进的情况，而我们组内进行多次沟通交流后这个问题有所改善，最终也成功编写出了产品。吴创同学在编写自己的内容时非常尽力，经常进行组内讨论，最后成功完成了自己的任务；刘逸洋同学作为组长，不仅承担了很大一部分任务，搭建了程序主框架，还在我未能及时完成二阶段任务的时候把锅接了过去，保证了计划的顺利进行。非常感谢两位的帮助。



6 参考文献资料

[1] 何钦铭,颜晖.C 语言程序设计[M].北京:高等教育出版社,2015

[2] Stephen Prata.C Primer Plus[M].北京:人民邮电出版社,2019