

**实验报告封面**

**课程名称： 算法课程设计 课程代码： GF2001**

**任课老师： 李俊琴 实验指导老师: 李俊琴**

# 实验报告名称： 实验1 排序算法及其应用

学生姓名： 许景智

学号： 2220631128 教学班： BIA

递交日期： 2022.11.29

签收人：

我申明，本报告内的实验已按要求完成，报告完全是由我个人完成，并没有抄袭行为。我已经保留了这份实验报告的副本。

申明人(签名): 许景智

**实验报告评语与评分：**

**评阅老师签名：**

**内页写作格式**

**一、实验名称：**要用最简练的语言反映实验的内容。

**二、实验日期：**写明做实验的具体年、月、日及组別。

**三、实验目的：**使用简洁的文字或关键字来敘述，是以怎样的目的作此实验的。

**四、实验环境：**实验的操作系统和软件等。

**五、实验的步骤和方法：**这是实验报告极其重要的内容。这部分要写明经过哪几个步骤。

**六、数据记录和计算：**指从实验中测到的数据以及计算结果。

**七、实验结果或结论：**即根据实验过程中所见到的现象和测得的数据，得出结论。

**八、备注或说明**：可写上实验成功或失败的原因，实验后的心得体会、建议等。

|  |  |
| --- | --- |
| **1、实验名称** | **实验1 排序算法及其应用** |
| **2、实验日期** | **2022.11.29** |
| **3、实验目的** | |
| 1. 掌握算法设计概念及流程 2. 掌握算法评价的方法 | |
| **4、实验环境** | |
| 1、硬件环境：i386兼容机，2G剩余硬盘空间；  2、软件环境：Windows XP以上兼容系统；  3、使用资源：开发环境VC++2017。 | |
| **5、实验的步骤和方法** | |
| 1. **给定n个整数组成的序列a1, a2, …, an, 用快速排序法将该序列排序。**  * 说明：快速排序是一种交换排序，它由C. A. R. Hoare在1962年提出。快速排序(quick sort)的采用了分而治之(divide and conquer)的策略：将原问题分解为若干个规模更小但结构与原问题相似的子问题。递归地解这些子问题，然后将这些子问题的解组合为原问题的解。 * 算法思想：快速排序的基本思想是：通过一趟排序将要排序的数据分割成独立的两部分：分割点左边都是比它小的数，右边都是比它大的数。然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。 * 图解：  1. **根据上述理解，设计算法，实现对序列的排序算法**   **注意：首先画出设计算法的流程图，然后再根据流程图进行编程实现。**   1. **描述游戏中的英雄，每英雄有自己的属性：姓名，生命值，防御力，攻击力，魔法值等。编程实现召唤英雄组成战队的功能。根据英雄的某个属性将英雄排序，然后输出该英雄的信息。要求：**  * **控制台程序** * **英雄的存储空间是动态增长的** (使用melloc、realloc函数) * **不能使用stl或其它第三方库** * **使用快速排序法**  1. **拓展练习：**  * **使用堆排序和桶排序两种排序算法实现第（3）步的功能**   完成上面的练习后，填写实验报告，将**实验报告word、运行结果录屏，分别上交智慧树**。 | |
| **6、数据记录和计算** | |
| （快排算法的流程图，1、2题关键代码请贴在此处）  流程图：  第三题排序算法，使用类做数组：  void QuickSort(int L, int R, Type type)  {  if(L >= R) return;  int left = L;  int right = R;  Hero pivot = this->Heros[left];  while(left < right)  {  while(left < right)  {  bool condition = false;  switch(type)  {  case Name:  {  int cp = this->Heros[right].Name.compare(pivot.Name);  condition = (cp == 0 || cp == 1) ? true : false;  }  break;  case HP:  condition = this->Heros[right].HP >= pivot.HP ? true : false;  break;  case DV:  condition = this->Heros[right].DV >= pivot.DV ? true : false;  break;  case OC:  condition = this->Heros[right].OC >= pivot.OC ? true : false;  break;  case MV:  condition = this->Heros[right].MV >= pivot.MV ? true : false;  break;  }  if(condition)  {  right--;  }  else  {  this->Heros[left] = this->Heros[right];  break;  }  }  while(left < right)  {  bool condition;  switch(type)  {  case Name:  {  int cp = this->Heros[left].Name.compare(pivot.Name);  condition = (cp == 0 || cp == -1) ? true : false;  }  break;  case HP:  condition = this->Heros[left].HP <= pivot.HP ? true : false;  break;  case DV:  condition = this->Heros[left].DV <= pivot.DV ? true : false;  break;  case OC:  condition = this->Heros[left].OC <= pivot.OC ? true : false;  break;  case MV:  condition = this->Heros[left].MV <= pivot.MV ? true : false;  break;  }  if(condition)  {  left++;  }  else  {  this->Heros[right] = this->Heros[left];  break;  }  }  if(left >= right)  {  this->Heros[left] = pivot;  }  }  this->QuickSort(L, right - 1, type);  this->QuickSort(right + 1, R, type);  } | |
| **7、实验结果或结论**  **）** | |
| **我对“算法”的理解**：  算法用来解决一些特定条件的问题，一种问题可以有不同的算法来解决，最优的算法可以用最少的复杂度来解决问题。  **我对排序算法的了解：（种类？区别？应用？）**  **插本考了选择排序**  **快速排序、希尔排序、插入排序、冒泡排序、堆排序** | |
| **8、备注或说明** | |
| **用类来解决动态数组创建、尝试使用结构体代码有点乱。**  **类里只写了添加跟列举、快速排序。**  **代码还是很乱的，c++没有JavaScript方便。** | |