

**实 验 报 告**

**（2019 / 2020 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 数据结构 | | | | | |
| 实验名称 | 各种内排序算法的实现及性能比较 | | | | | |
| 实验时间 | 2019 | 年 | 12 | 月 | 13 | 日 |
| 指导单位 | 计算机学院 | | | | | |
| 指导教师 | 邹志强 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 张颖 | 班级学号 | B18030406 |
| 学院(系) | 计算机学院 | 专 业 | 计算机科学与技术 |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | **各种内排序算法的实现及性能比较** | | | **指导教师** | **邹志强** |
| **实验类型** | **验证** | **实验学时** | **2** | **实验时间** | **2019-12-13** |
| 1. **实验目的和要求**   1.掌握各种内排序算法的实现方法。  2.学会分析各种内排序算法的时间复杂度。 | | | | | |
| 二、**实验环境(实验设备)**  硬件：微型计算机  软件：Windows 操作系统、Dev-C++ 5.11 | | | | | |
| **三、实验原理及内容**  1、算法设计  简单选择排序、直接选择排序、冒泡排序、快速排序、两路合并排序以及堆排序。  2、算法实现与分析  #include<iostream>  using namespace std;  #include<time.h>  #include<algorithm>  #include<stdlib.h>  #include <cstdio>  #include <cstring>  #include<windows.h>  #include<Mmsystem.h>  #define random(x) (rand()%x)  clock\_t start,stop;  double duration;  //简单选择排序  void SelectSort(int A[],int n)  {  int small;  for(int i = 0;i < n-1;i ++)  {  small = i;  for(int j = i+1;j < n;j ++)  {  if(A[j]<A[small])  small = j;  }  if(small != i){  swap(A[i],A[small]);  }  }  }  //直接插入排序  void InsertSort(int A[],int n)  {  for(int i=1;i<n;i++)  {  int j=i;  int temp=A[j];  while(j>0&&temp<A[j-1])  {  A[j]=A[j-1];  j--;  }  A[j]=temp;  }  }  //冒泡排序  void BubbleSort(int A[],int n)  {  int i,j,last;  i=n-1;  while(i>0)  {  last=0;  for(j=0;j<i;j++)  if(A[j+1]<A[j])  {  swap(A[j],A[j+1]);  last=j;  }  i=last;  }  }  void QSort(int A[],int left,int right)  {  int i,j;  if(left<right)  {  i=left;  j=right+1;  do  {  do i++;while (A[i]<A[left]);  do j--;while (A[j]>A[left]);  if(i<j)  swap(A[i],A[j]);  }while(i<j);  swap(A[left],A[j]);  QSort(A,left,j-1);  QSort(A,j+1,right);  }  }  //快速排序  void QuickSort(int A[],int n)  {  QSort(A,0,n-1);  }  void GQSort(int A[],int left,int right)  {  int i,j;  if(right>=9)  {  if(left<right)  {  i=left;  j=right+1;  do  {  do i++;while (A[i]<A[left]);  do j--;while (A[j]>A[left]);  if(i<j)  swap(A[i],A[j]);  }while(i<j);  swap(A[left],A[j]);  QSort(A,left,j-1);  QSort(A,j+1,right);  }  }  else  {  InsertSort(A,right-left+1);  return ;  }  }  //改进的快速排序  void GQuickSort(int A[],int n)  {  GQSort(A,0,n-1);  }  //两路合并排序  void Merge(int A[],int i1,int j1,int i2,int j2)  {  int\* Temp=new int[j2-i1+1];  int i=i1,j=i2,k=0;  while(i<=j1&&j<=j2)  {  if(A[i]<=A[j])  Temp[k++]=A[i++];  else Temp[k++]=A[j++];  }  while (i<=j1)  Temp[k++]=A[i++];  while(j<=j2)  Temp[k++]=A[j++];  for(i=0;i<k;i++)  A[i1++]=Temp[i];  delete[] Temp;  }  void MergeSort(int A[],int n)  {  int i1,j1,i2,j2;  int size=1;  while(size<n)  {  i1=0;  while(i1+size<n)  {  i2=i1+size;  j1=i2-1;  if(i2+size-1>n-1)  j2=n-1;  else  j2=i2+size-1;  Merge(A,i1,j1,i2,j2);  i1=j2+1;  }  size\*=2;  }  }  //堆排序  void AdjustDown(int A[],int r,int j)  {  int child=2\*r+1;  int temp=A[r];  while (child<=j)  {  if((child<j)&&(A[child]<A[child+1]))  child++;  if(temp>=A[child])  break;  A[(child-1)/2]=A[child];  child=2\*child+1;  }  A[(child-1)/2]=temp;  }  void HeapSort(int A[],int n)  {  for(int i=(n-2)/2;i>-1;i--)  AdjustDown(A,i,n-1);  for(int i=n-1;i>0;i--)  {  swap(A[0],A[i]);  AdjustDown(A,0,i-1);  }  }  void Output(int A[],int n){  for(int i = 0;i < n;i ++){  printf("%d ",A[i]);  }  printf("\n\n");  }  void Menu(){  printf("拥有的排序方法有:\n\n");  printf("1. 选择排序\n");  printf("2. 插入排序\n");  printf("3. 冒泡排序\n");  printf("4. 快速排序\n");  printf("5. 合并排序\n");  printf("0. Exit\n");  printf("\n\n请选择要采用的排序方法:");  }  int main()  {  int \*a,\*b,\*c,\*d,\*e,\*f,\*g,NUM,choose;  printf("请输入要随机生成的数组成的列表长度:");  scanf("%d",&NUM);  a = (int\*)malloc(NUM \* sizeof(int));  b = (int\*)malloc(NUM \* sizeof(int));  c = (int\*)malloc(NUM \* sizeof(int));  d = (int\*)malloc(NUM \* sizeof(int));  e = (int\*)malloc(NUM \* sizeof(int));  f = (int\*)malloc(NUM \* sizeof(int));  g = (int\*)malloc(NUM \* sizeof(int));  memset(a,0,sizeof(a));  memset(b,0,sizeof(b));  memset(c,0,sizeof(c));  memset(d,0,sizeof(d));  memset(e,0,sizeof(e));  memset(f,0,sizeof(f));  memset(g,0,sizeof(g));  srand((int)time(0));  for(int x = 0;x < NUM;x ++){  a[x]=random(NUM);  b[x]=a[x];  c[x]=a[x];  d[x]=a[x];  e[x]=a[x];  f[x]=a[x];  g[x]=a[x];  }  printf("\n原列表是:\n");  Output(a,NUM);  Menu();  scanf("%d",&choose);  switch(choose){  case 1:SelectSort(a,NUM);Output(a,NUM);break;  case 2:InsertSort(a,NUM);Output(a,NUM);break;  case 3:BubbleSort(a,NUM);Output(a,NUM);break;  case 4:QuickSort(a,NUM);Output(a,NUM);break;  case 5:MergeSort(a,NUM);Output(a,NUM);break;  case 0:printf("错误输入\n");break;  }  return 0;  }  3、实验结果与结论            选择排序和冒泡排序的时间复杂度都是O()，快速排序、归并排序和选择和堆排序的时间复杂度都是都是O(nlogn)。 | | | | | |

**实 验 报 告**

|  |
| --- |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等）  (一)实验中遇到的主要问题及解决方法  在实验中遇到的问题主要是对各类排序的算法还掌握不熟练，导致出现大量编译错误，后来对照教材进行一定程度的修改过后解决了问题。  （二）实验心得  一直只看教材去学习编程不如实际敲一下进行编译处理更能让自己学到东西。  （三）意见与建议（没有可省略） |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **支撑毕业要求指标点**   《数据结构》课程支撑毕业要求的指标点为：  1.2-M掌握计算机软硬件相关工程基础知识，能将其用于分析计算机及应用领域的相关工程问题。  3.2-H能够根据用户需求，选取适当的研究方法和技术手段，确定复杂工程问题的解决方案。  4.2-H能够根据实验方案，配置实验环境、开展实验，使用定性或定量分析方法进行数据分析与处理，综合实验结果以获得合理有效的结论。   |  |  |  |  | | --- | --- | --- | --- | | 实验内容 | 支撑点1.2 | 支撑点3.2 | 支撑点4.2 | | 线性表及多项式的运算 | √ |  |  | | 二叉树的基本操作及哈夫曼编码译码系统的实现 |  | √ | √ | | 图的基本运算及智能交通中的最佳路径选择问题 |  | √ | √ | | 各种内排序算法的实现及性能比较 | √ |  | √ | | | | | | |
| **六、指导教师评语 (含学生能力达成度的评价)**  如评分细则所示 | | | | | |
| **成 绩** | XX | **批阅人** | **XX** | **日 期** | **XXX** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **回答问题准确度** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  |  |  |  |  |