数据结构实训

ICT数据结构课程组

实验四 多维数字信号的排序

▶▶▶ 实验目标

- 1) 深入理解各种排序的算法思想、方法、稳定性及时间和空间复杂度;
- 2) 掌握插入排序、交换排序、选择排序的算法实现;
- 3) 能够实现多维信号的排序操作;
- 4) 提高实际动手进行程序设计的能力。

解决的问题要来源于实际工程:要对日常生活中的一个具体应用进行抽象建模与求解,鼓励对国家重大工程中的一个具体点进行抽象建模与求解。

学生信息管理:按指定关键字排序

图书管理系统: 按指定关键字排序

图像匹配: 京东淘宝的以图搜图

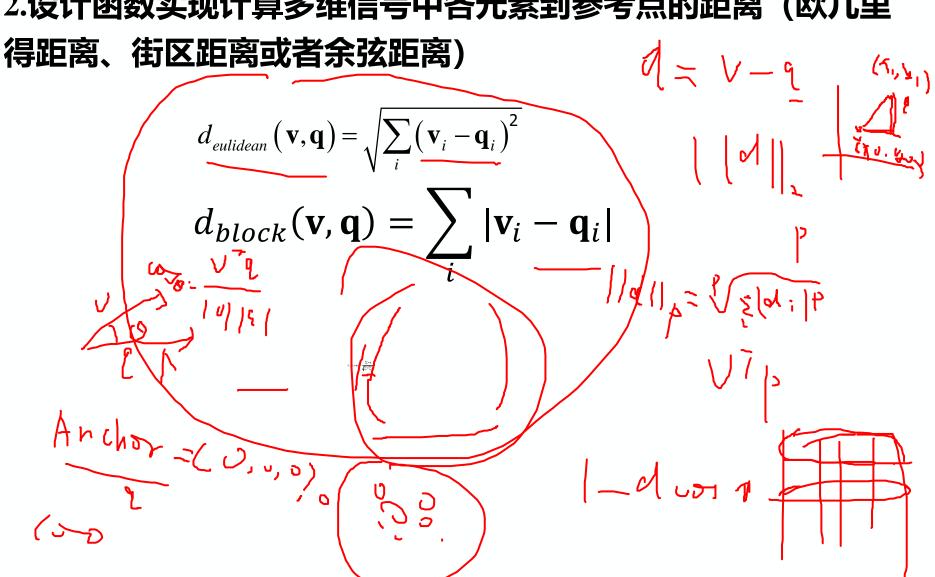
文本匹配: 网页匹配、文件匹配等

声音匹配;计算与查询声音的距离并排序

• • • •

▶ 实验内容

2.设计函数实现计算多维信号中各元素到参考点的距离





```
KeyType AnchorPoint[3];
AnchorPoint[0] = 0; AnchorPoint[1] = 0; AnchorPoint[2] = 0;
while(tmpp!=NULL)//计算欧几里得距离
       tmpD = (AnchorPoint[0] - tmpp->Item1)
                * (AnchorPoint[0] - tmpp->Item1)+
                (AnchorPoint[1] - tmpp->Item2)
                * (AnchorPoint[1] - tmpp->Item2)+
                (AnchorPoint[2] - tmpp->Item3)
               * (AnchorPoint[2] - tmpp->Item3);
        signal[k].key = sqrt(tmpD);
       tmpp=tmpp->next;
```

3. 实现对给信号中各元素依据1)中计算的距离进行排序,排序算法分别从插入排序、交换排序、选择排序三类算法中各选择一个即可(红色标记为期望实现的算法);

| | 平均 | 比较次数 | | 移动次数 | | 稳 | 附力口 |
|-------|--------------------------|----------------------|-------|---------------------------|--------|-----------|----------|
| 排序方法 | 时间 | 最好 | 最差 | 最好 | 最 差 | 定 性 | 存储 |
| 直接插入 | n^2 | n | n^2 | 0 | n^2 | $\sqrt{}$ | 1 |
| 折半插入 | n^2 | n log ₂ n | | 0 | n^2 | $\sqrt{}$ | 1 |
| 希尔排序、 | $n^{1.3}$ | | | 0 | | × | 1 |
| 起泡排序 | n^2 | n | n^2 | 0 | n^2 | $\sqrt{}$ | 1 |
| 快速排序 | nlog ₂ n | $nlog_2n$ | n^2 | $n \log_2 n$ | n^2 | × | log_2n |
| 简单选择 | n^2 | n^2 | | 0 | n | $\sqrt{}$ | 1 |
| 堆排序 | <u>nlog₂n</u> | <u>n log_n</u> | | <u>n log₂n</u> | | × | <u>1</u> |

4. 比较分析所选择三种算法的优缺点和适用范围。

1) 理论分析

| -) -10/5 // | | 1 | | | | | l e |
|-------------|----------------|-----------------|-------|----------------|-------|-------------------------|----------------|
| | 平均 | 比较次数 | | 移动次数 | | 稳 | <i>7/</i> +++□ |
| 排序方法 | 时间 | 最好 | 最差 | 最好 | 最差 | 定性 | 附加 存储 |
| 直接插入 | n^2 | n | n^2 | 0 | n^2 | $\left(\sqrt{}\right)$ | 1 |
| 折半插入 | n^2 | $n \log_2 n$ | | 0 | n^2 | $\left(\sqrt{1}\right)$ | 1 |
| 希尔排序 | $n^{1.3}$ | | | 0 | | × | 1 |
| 起泡排序 | n^2 | n | n^2 | 0 | n^2 | $\sqrt{}$ | 1 |
| 快速排序 | $nlog_2n$ | $nlog_2n$ | n^2 | $n \log_2 n$ | n^2 | × | log_2n |
| 简单选择 | n^2 | n^2 | | 0 | n | V | 1 |
| 堆排序 | <u>nlog</u> 2n | <u>n log</u> 2n | | <u>n log_n</u> | | <u>×</u> | <u> </u> |

4. 比较分析所选择三种算法的优缺点和适用范围。

```
2) 实际运行时间统计

start = clock();

InsertSort(R, n);

finish = clock();

Total_time = (double)(finish - start) / CLOCKS_PER_SEC;

printf("插入排序后(耗时=%f秒):\n", Total_time);
```

- 1. 深入理解各种排序的算法思想、方法、稳定性及时间和空间复杂度
- 2. 建议采用C、<u>C++等高</u>级语言;
- 3. 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;
- 4. 课前编写实现实验内容的程序;
- 5. 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;
- 6. 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。
- 7. 数据点个数要大于1000, 每个数据点的维数至少为2.
- 8. 提交实验报告。

1. 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;
  //读数据_
  如: signal=ReadNDSignalFromFile(SourceFileName,nNum, nFlag
  //输出读入的数据:
                          R CloneArr(signal,n,0);printf("排序前: "); DispList(R, n,0);
  //计时: start = clock();
  //调用排序算法: InsertSort(R, n);
                 finish = clock();Total_time = (double)(finish - start) / CLOCKS_PER_SEC;printf("插入
  //计时结束:
  排序后(耗时=%f秒):\n", Total_time);
  //显示并输出结果: DispList(R, n,0);WriteNDSignalToFile(ResultFileName, R, n,1); free(R);R=NULL;
  //调用其他排序类似上述步骤
  //清理现场
```

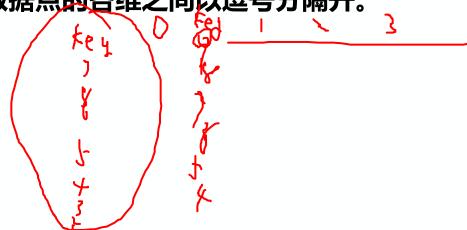
- 2. 测试用例:简单到复杂、边界条件。
- 1) 简单输入输出测试用例: 很容易知道真实答案的用例

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。



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;

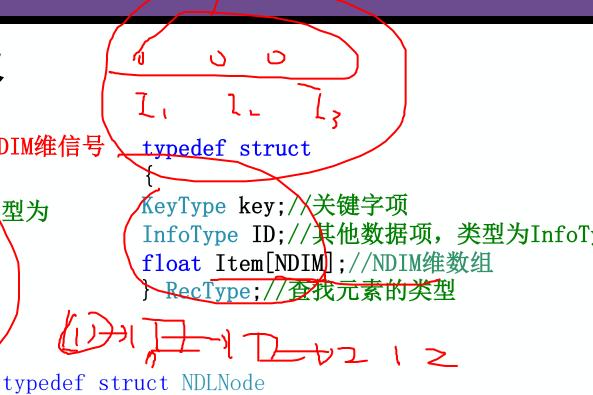
the of

- (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。

▶▶▶补充

0.1多维信号数据结构定义

由于数据文件中数据个数 未知,先建立链表临时存 储



KeyType key;//关键字项
InfoType ID; //其他数据项, 类型为
InfoType //
float Item1;
float Item2;
float Item3;
struct NDLNode* next;//指向后继结点
} NDLinkNode;

0.2多维信号读取

```
while (!feof(fps))
fscanf_s(fps, "%f,", &ID, sizeof(float));
f(nFlag)
<u>fscanf s(fps, "%f,", &key, sizeof(KeyType</u>))
                 "%f, ", &e[0], sizeof(KeyType))
fscanf_s(fps,
fscanf_s(fps, "%f,", &e[1], sizeof(KeyType))
fscanf_s(fps, "%f\n", &e[2], sizeof(KeyType))
NDListAppend(h, e, ID, nFlag, key);
 int(aPointNum) > NDListLength(h);
 signal = (RecType*) malloc(sizeof(RecType) * nPointNum)
 while (tmpp != NULL)
 signal[k].ID = tmpp->ID;
 signal[k].Item1 = tmpp->Item1;
 signal[k].Item2 = tmpp->Item2;
 signal[k].Item3 = tmpp->Item3;
 k++:
 tmpp = tmpp->next;
```

文件(F) 編辑(E) 格式(O) 查看(V) 帮助(H)

24,1.000000,41,000000,40.000000,31.000000

23,2.000000,34.000000,36.000000,44.800000

22,3.000000,34.000000,44.000000,59.000000

21,3.000000,47.000000,40.000000,47.000000

20,3.000000,55.000000,39.000000,70.000000

19,6.000000,16.000000,11.000000,18.000000

18,7.000000,85.000000,86.000000,79.000000

17,8.000000,88.000000,125.000000,161.000000

16,9.000000,137.000000,164.000000,181.000000

生成Signal数组

▶▶▶ 0.3写文件参考

```
void WriteNDSignalToFile const char* FileName, RecType s[], int n, int
nFlag)//nFlag=0,便于观看; nFlag=1格式同输入文件,便于后续查找)
FILE* fpr;
fopen_s(&fpr, _FileName, "w");
fopen s (&fpr, "d:\\result.txt", "w");
if (hFlag)
for (int i = 0; i < n; i++)
fprintf(fpr, "%d, %f, %f, %f, %f\n", s[i]. ID, \[ \] i]. key, \[ \] s[i]. Item1, s[i]. Item2,
s[i]. Item3);
else
for (int i = 0; i < n; i++)
fprintf(fpr, "Key=%f, ID=%d, Item1=%f, Item2=%f, Item3=%f \n", s[i].key,
\{s[i]. ID, s[i]. Item1, s[i]. Item2, s[i]. Item3\}\}
fclose(fpr);}
```

▶▶▶ 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

- 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) 实验程序包命名方式: PDS实验序号(之位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**1000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。