**南京师范大学中北学院**

**《数据结构》**

**实**

**验**

**报**

**告**

**姓 名: 钟玮**

**学 号: 85213241**

**班 级: 852132**

**日 期: 6.17**

**排序-2**

1. **实验目的**

**1、掌握待排记录的数据类型的定义；**

**2、掌握顺序表的几种排序算法，如冒泡排序、快速排序、简单选择排序等。**

1. **实验内容**

**1、构造一个待排序的顺序表；**

**2、对顺序表L作冒泡排序；**

**3、对顺序表L作快速排序；**

**4、对顺序表L作简单选择排序**

**5、遍历顺序表中的数据。**

1. **实验主要代码**

**可以截图并做简要解释，如：**

* 1. **构造一个待排序的顺序表**

**Status Create\_SqList(SqList &L){**

**printf("请输入待排序的表长: ");**

**scanf("%d",&L.length);**

**KeyType key;**

**printf("请输入待排序的数据(用空格隔开):\n");**

**for(int i=1;i<=L.length;i++){**

**scanf("%d",&key);**

**L.r[i].key=key;**

**}**

**return OK;**

**}**

* 1. **对顺序表L作冒泡排序**

**void BubbleSort(SqList &L) {**

**int i = L.length;**

**int lastExchangeIndex;**

**RcdType temp;**

**while (i > 1) {**

**lastExchangeIndex = 1;**

**for (int j = 1; j < i; j++)**

**if (L.r[j+1].key < L.r[j].key) {**

**temp = L.r[j];**

**L.r[j] = L.r[j+1];**

**L.r[j+1] = temp;**

**lastExchangeIndex = j;**

**}**

**i = lastExchangeIndex;**

**}**

**}**

* 1. **对顺序表L作快速排序**

**int Partition(SqList &L, int low, int high) {**

**KeyType pivotkey;**

**L.r[0] = L.r[low];**

**pivotkey = L.r[low].key;**

**while (low < high) {**

**while (low < high && L.r[high].key >= pivotkey) --high;**

**L.r[low] = L.r[high];**

**while (low < high && L.r[low].key <= pivotkey) ++low;**

**L.r[high] = L.r[low];**

**}**

**L.r[low] = L.r[0];**

**return low;**

**}**

**void QSort(SqList &L, int low, int high) {**

**if (low < high) {**

**int pivotloc = Partition(L, low, high);**

**QSort(L, low, pivotloc - 1);**

**QSort(L, pivotloc + 1, high);**

**}**

**}**

**void QuickSort(SqList &L) {**

**QSort(L, 1, L.length);**

**}**

* 1. **对顺序表L作简单选择排序**

**int SelectMinKey(SqList L, int i) {**

**int min = i;**

**for (int j = i + 1; j <= L.length; j++) {**

**if (L.r[j].key < L.r[min].key)**

**min = j;**

**}**

**return min;**

**}**

**void SelectSort(SqList &L) {**

**for (int i = 1; i < L.length; ++i) {**

**int j = SelectMinKey(L, i);**

**if (i != j) {**

**RcdType temp = L.r[i];**

**L.r[i] = L.r[j];**

**L.r[j] = temp;**

**}**

**}**

**}**

* 1. **遍历顺序表中的数据**

**void Traverse\_SqList(SqList L) {**

**for (int i = 1; i <= L.length; i++) {**

**printf("%d ", L.r[i].key);**

**}**

**printf("\n");**

**}**

**6、main函数**

**int main() {**

**SqList L1, L2, L3;**

**printf("冒泡排序测试:\n");**

**Create\_SqList(L1);**

**BubbleSort(L1);**

**printf("冒泡排序结果:\n");**

**Traverse\_SqList(L1);**

**printf("快速排序测试:\n");**

**Create\_SqList(L2);**

**QuickSort(L2);**

**printf("快速排序结果:\n");**

**Traverse\_SqList(L2);**

**printf("简单选择排序测试:\n");**

**Create\_SqList(L3);**

**SelectSort(L3);**

**printf("简单选择排序结果:\n");**

**Traverse\_SqList(L3);**

**return 0;**

**}**

1. **实验结果**

**实验运行结果的截图，如main函数运行结果截图：**

