**排序**

1. **实验目的和要求**

设数组a[n]中存放了n个不同元素，元素的关键字值都介于0到n-1，且互不相等。试设计一个时间复杂度为O(N)，空间复杂度为O(1)的算法将其按递增次序排序。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 排序

所谓排序就是把集合中的数据元素按照它们的关键字的非递减或非递增序排成一个序列。

稳定与非稳定排序：假定在待排序的集合中存在多个关键字值相同的数据元素。如果经过排序后，这些数据元素的相对次序保持不变则称为稳定排序，否则称为不稳定排序。

内排序与外排序：

内排序是指被排序的数据元素全部存放在计算机的内存之中，并且在内存中调整数据元素的相对位置。

外排序是指在排序的过程中，数据元素主要存放在外存储器中，借助于内存储器逐步调整数据元素之间的相对位置。

1. 实验项目
   1. 分析
      1. 所给的数据并不是普通的数据，数据的元素的关键字值都介于0到n-1,且互不相等，所以我们要设计的排序的算法要根据所给数据的特点来设计。
      2. 由于数据元素的关键字值处于0到n-1,所以我们可以利用这个特性来将关键字值为k的元素放到数组a[k]的位置，放置好位置之后我们实际上就已经完成了排序。
      3. 先从a[0]元素开始扫描，若a[0]位置的元素的关键字值为0，则下标加一，之后同理，直到遇到一个元素a[i]的关键字值不为i(设关键字值为j),然后将a[i]中的元素取出，放到临时变量tmp中，然后找到tmp元素对应的关键字值对应的位置a[j]，将tmp与a[j]交换元素(设a[j]元素对应的关键字值为k)，然后找到a[k]…
   2. 方案
      1. 构造一个带有关键字值的简单结构
      2. 设计上述的排序算法
      3. 构造一个包含n个上述简单结构的数组a[n]
      4. 对数组a[n]进行排序
   3. 测试数据

int main()

{

element a[20];

for (int i = 0; i < 20; i++)

{

a[19 - i].key = i;

a[19 - i].data = i\*i;

}

sort(a);

for (int i = 0; i < 20; i++)

{

cout << a[i].data << endl;

}

return 0;

}

* 1. 出错情况

第一次的输出结果为：

0

324

289

256

225

196

169

144

121

100

81

64

49

36

25

16

9

4

1

361

这样的输出结果表明我们的算法只交换了首尾两个位置的元素，应该是原来的设计的算法存在一定的问题。

我们可以将原来的算法修改为直接交换a[i]和a[j]中的元素.

* 1. 最终结果

0

1

4

9

16

25

36

49

64

81

100

121

144

169

196

225

256

289

324

361

**三、实验小结**

1. 重点

针对特殊的数据结构的排序算法。

1. 难点

如何让设计的排序的算法的时间复杂度为O(N),空间复杂度为O(1)。

1. 收获、体会

排序算法多种多样，但是每种算法都有自己的特点，同时对于特定的数据的元素可以根据他们本身的特性来设计特殊的排序的算法。