内部排序算法比较课程设计

1. 问题描述

排序是计算机经常进行的一种操作。排序算法有直接插入排序，快速排序，选择排序，希尔排序，堆排序等等。排序算法的效率是影响计算机性能的重要因素。在教科书中，各种排序算法只给了执行时间，在这里将比较各个排序算法的比较和移动次数，对各种排序算法有更直观的感受。

1. 程序设计

（1）变量说明

全局变量cmp,mov记录排序算法的比较和移动次数，tmp数组在归并算法中临时储存关键字，is\_all用来在下文中判断是否对所有算法运行，并且记录在对所有算法运行时储存要测试的数据个数，mem二维数组用来记录在对所有算法运行时的比较和移动次数。

typedef struct

{

int length;

int data[200000];

}SqList;

SqList 同来储存要排序的表，其中Length表示长度，Data数组储存数据。

（2）函数说明

generate(SqList \*L)用来生成随机数并储存在L中

print(SqList \*L) 用来打印L

reverse(SqList \*L) 将列表转成逆序

以下十个函数用来实现这十个排序算法

BubbleSort(SqList \*L); 冒泡

QuickSort(SqList \*L); 快速排序

InsertSort(SqList \*L); 直接插入

SelectSort(SqList \*L); 选择排序

shellSort(SqList \*L); 希尔排序

HeapSort(SqList \*L); 堆排序

MergeSort(SqList \*L); 归并排序

BInsertSort(SqList \*L); 折半插入排序

Path2Insert(SqList \*L); 二路插入排序

radixsort(SqList \*L); 基数排序

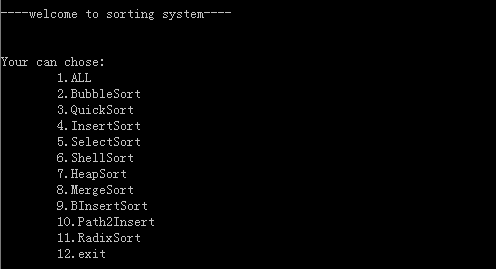
All（SqList \*L）

用来封装这十个函数并且统计正序和逆序的结果。

MAIN函数用一个while循环询问用户是否需要进行哪种服务，并调用相应的函数。如果为12则退出循环，并退出程序。

1. 使用方法

运行程序后界面如下



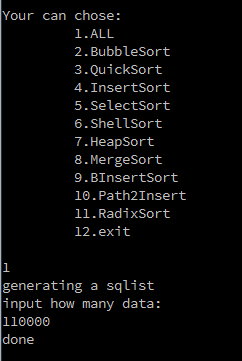
输入1用所有算法对表进行排序

输入2~11对单个算法进行排序

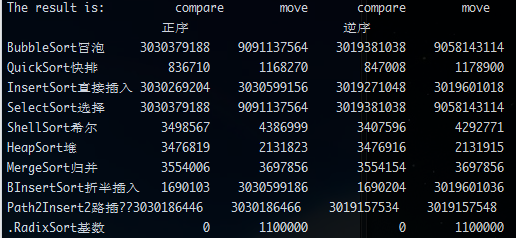
输入12退出

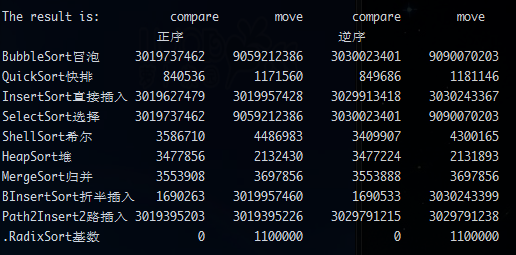
1. 同时对多个算法进行比较

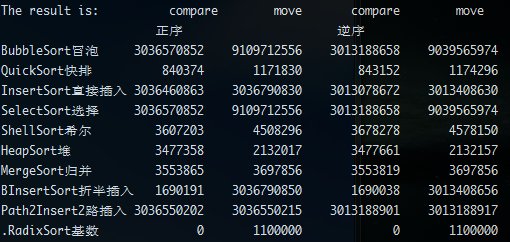
运行程序，输入1对所有算法进行测试，再输入110000对110000个数据进行测试。

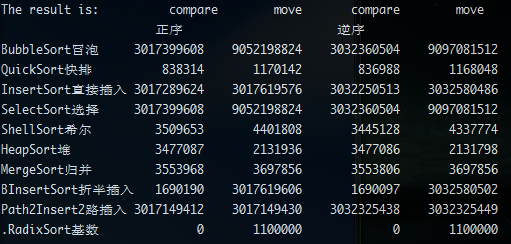


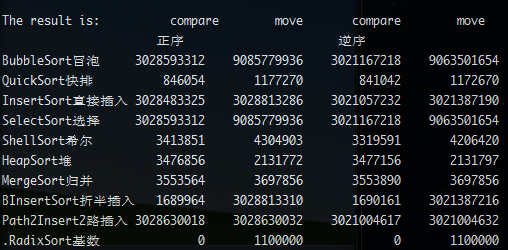
连续测试测试5次，结果如下











1. 数据处理

对五次正序和逆序的结果求和取平均如下：

|  |  |  |
| --- | --- | --- |
| 算法 | 比较次数 | 移动次数 |
| 冒泡 | 3024880124 | 9074640372 |
| 快排 | 841986 | 1173413 |
| 直接插入 | 3024770137 | 3025100096 |
| 选择 | 3024880124 | 9074640372 |
| 希尔 | 3487648 | 4380426 |
| 堆 | 3477201 | 2131953 |
| 归并 | 3553886 | 3697856 |
| 折半插入 | 1690174 | 3025100122 |
| 2路插入 | 3024737898 | 3024737915 |
| 基数 | 0 | 1100000 |
|  |  |  |

结论：

由上表可以看出，冒泡、直接插入、选择，二路插入的比较和移动次数最多，快排的比较和移动次数最少。基数排序比较次数为0，移动次数也较少。

除了基数排序与其无关，比较次数为0，移动次数是数据个数的十倍。

结论：

从以上图表可以看出，希尔排序的波动性最大，快速排序波动性较大，冒泡排序、选择排序、堆排序、基数排序、归并排序的波动性最小，二路插入排序、折半插入排序、直接插入排序算法波动性波动性较小

分析：

希尔排序，当排序序列为正序时，比较次数为n，移动次数为0，当为逆序时，移动次数增加，复杂度增加，因此波动性较大。

快速排序最好的情况是正序序列，只需一趟排序，有n-1次比较，不移动数据，最差需进行n-1趟排序，进行n(n-1)/2次比较，并移动n(n-1)/2次，因此快速排序的波动性很大。

冒泡排序，选择排序比较次数相同，均为n(n-1)/2，移动次数变化也趋于固定值，所以冒泡和选择排序波动性较小。

基数排序不进行比较，移动次数与关键字的个数有关，为d\*n,移动次数固定,故波动性较小。

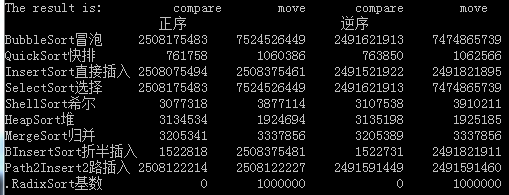
归并排序：MSort函数中递归将表平分，次数是固定的，一趟归并排序的操作是调用n/2h次算法merge,将两个有序段两两归并，并且在Merge算法中比较和移动的次数比较固定。

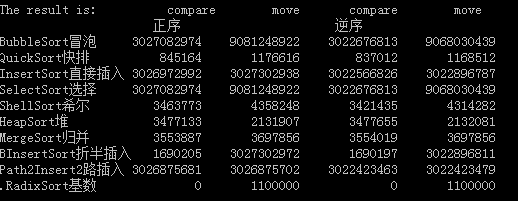
堆排序相对于快速排序在最坏的情况下，比较次数不超过2n(log2(n))，所以堆排序波动性较小。

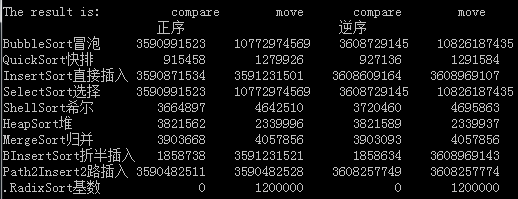
直接插入排序、2路插入排序、折半插入排序为正序时移动次数为0，比较次数小于等于n，逆序时在向后移动时复杂度增加。波动性较大。

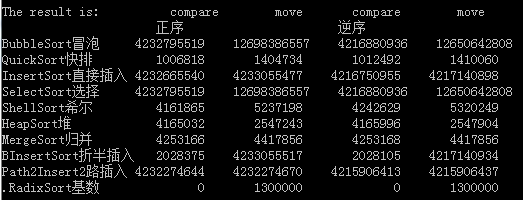
六． 不同表长对实验结果的影响

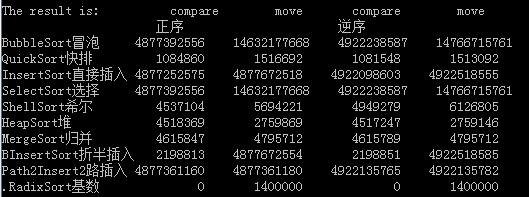
分别用表长为100000、110000、120000、130000、140000的数据进行测试，结果如下：











结论:从以上图表来看，除了希尔排序波动比较大，其他排序算法与要比较的数据量成大致成正比。