**数据结构实验报告**

姓名： 赵楠 学号：U201617007 班级：软工1601班

**实验八**

1. 问题描述

1、7.35，实现希尔排序。使用不同的增量序列，度量在不同序列的增量下，排序算法的时间性能。

2、如何找到10000个整数中第5大的数？假定内存只能一次容纳1000个数。编程实现你的算法，并分析时间复杂度。如果内存只能存放10个数呢？你

的算法是否仍然适用？

二、问题分析与算法设计

1、问题1

希尔排序算法简介：

根据增量对不同的序列进行插入排序。

如：一组数[ 13 14 94 33 82 25 59 94 65 23 45 27 73 25 39 10 ]，选择步长为5

排序之前：

13 14 94 33 82

25 59 94 65 23

45 27 73 25 39

10

排序之后：

10 14 73 25 23

13 27 94 33 39

25 59 94 65 82

45

可以看到每一列都已经排好序了。

而此时得到的数组即为 [ 10 14 73 25 23 13 27 94 33 39 25 59 94 65 82 45 ]，再选取步长为 3：

排序之前：

10 14 73

25 23 13

27 94 33

39 25 59

94 65 82

45

排序之后：

10 14 13

25 23 33

27 25 59

39 65 73

45 94 82

94

最后以1步长进行插入排序。

而不同增量问题，则只是改变步长改变的规则，其他不变。

1. 问题2

我们可以只存储当前所遇到的数的里面的前五个数字，这样可以节省空间。具体思想如下：

A. 首先用一个大小为5的数组来存储，需要计算的数组的前五个数并排序。

B. 之后便利需要计算的数组，若当前遇到的数字，比存储的数组里面的最小的数大，则插入存储的数组里面，但是这里注意，永远保持存储的数组的大小为5，插入的实际是替换和重排，但是由于数组本身一直是排好的，故插入重排的操作数会很低，同时随着排除的数的增加，需要插入的几率大大减少，几乎不需要插入，总的操作数也会相对减少。

1. 算法实现

代码见钉盘

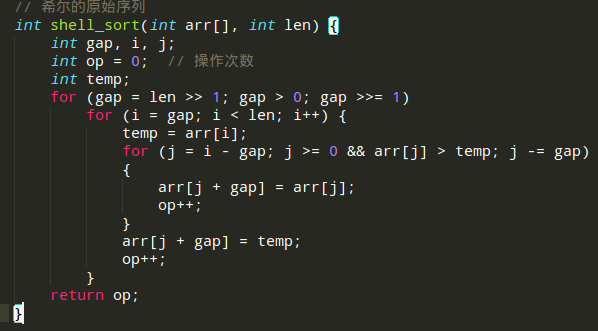
1、问题1

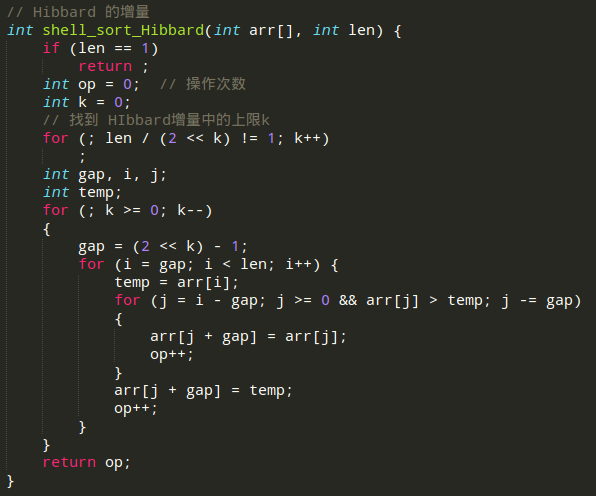
关键函数代码如下：

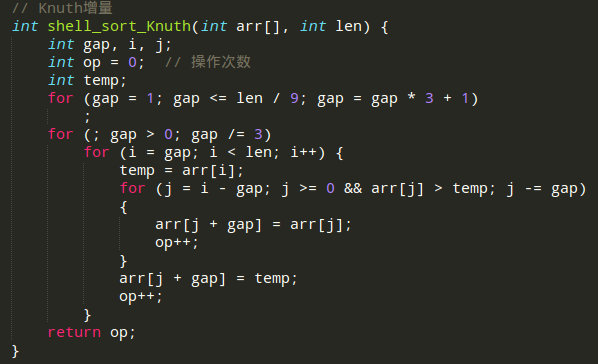
我只是计算了操作数，不计算时间，通过计算平均操作数来看算法复杂度。

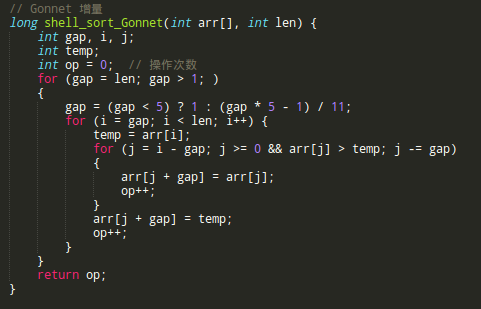
希尔的原始序列：

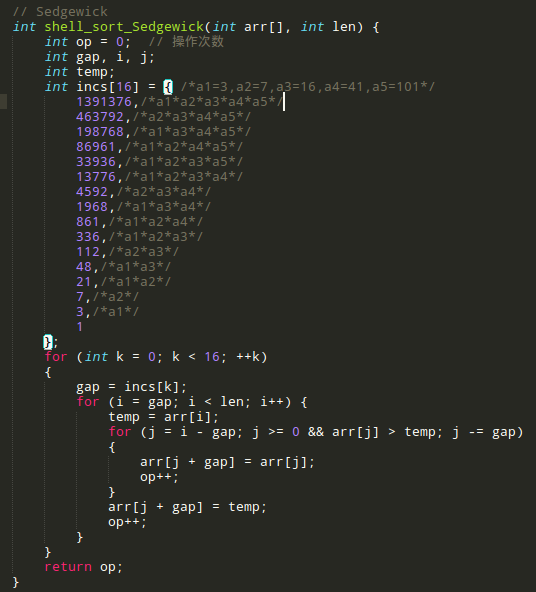
注：这里使用位运算加快速度







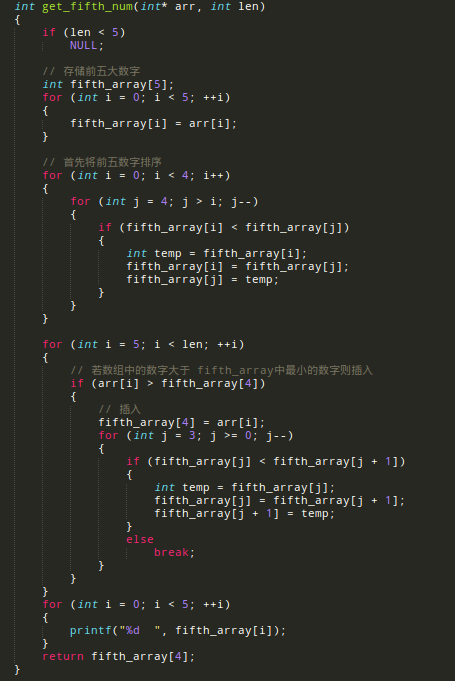




1. 问题2

关键函数代码如下：

这里只是存放了每次运算的前五大的数，以及循环需要的控制循环的类似i，j等数字，可以解决本问题所提出的要求。

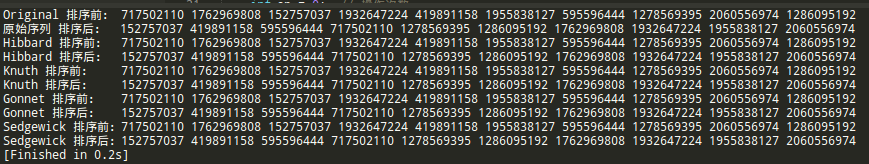


四、测试结果与分析

1. 问题1

测试数组生成：以当前时间做种（精确到毫秒），生成随机数，一共生成 SHELL\_LEN 次

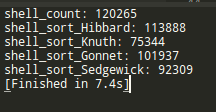
排序正确性测试：



ok

算法复杂度测试：

这里我对一个长达 10000的数组进行排序，并循环1000次，分别取得五种不同增量下的算法需要的操作数的平均值



这里可以看出 其中算法优劣性： Knuth > Sedgewick > Gonnet > Hibbard > 原始序列

1. 问题2

测试用例：1, 5, 3, 4, 6, 89, 67, 34, 2, 5, 4, 56, 24, 56, 67

结果：



ok