

希尔排序

希尔排序可以看做优化的插入排序， 缩小增量排序

1.排序原理

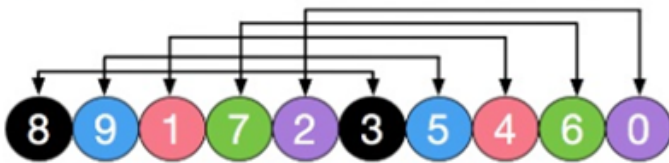
希尔排序是把记录按下标的一定增量分组，对每组使用直接插入排序算法排序；随着增量逐渐减少，每组包含的关键词越来越多，当增量减至1时，整个文件恰被分成一组，算法便终止

图解

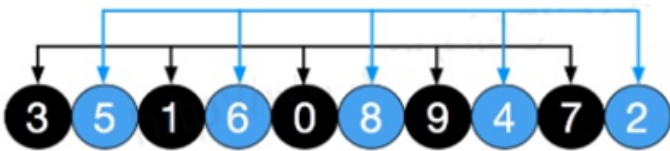
原始数组 以下数据元素颜色相同为一组



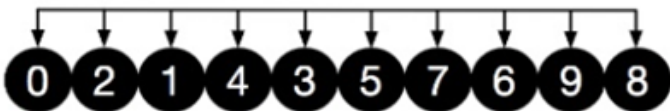
初始增量 $gap=length/2=5$ ，意味着整个数组被分为5组，[8,3] [9,5] [1,4] [7,6] [2,0]



对这5组分别进行直接插入排序，结果如下，可以看到，像3，5，6这些小元素都被调到前面了，然后缩小增量 $gap=5/2=2$ ，数组被分为2组 [3,1,0,9,7] [5,6,8,4,2]



对以上2组再分别进行直接插入排序，结果如下，可以看到，此时整个数组的有序程度更进一步啦。再缩小增量 $gap=2/2=1$ ，此时，整个数组为1组[0,2,1,4,3,5,7,6,9,8]，如下



经过上面的“宏观调控”，整个数组的有序化程度成果喜人。

此时，仅仅需要对以上数列简单微调，无需大量移动操作即可完成整个数组的排序。



2.代码实现

```
package Sort排序;
import java.util.Date;
/**
 * @author jndeng
 * @create 2019-12-17 17:15
 */
public class ShellSort {
    public static void main(String[] args) {

        // 创建要给80000个的随机的数组
        int[] arr = new int[800000];
        for (int i = 0; i < 800000; i++) {
            arr[i] = (int) (Math.random() * 800000);
        }
        Date date1 = new Date();
        long time1 = date1.getTime();

        //交换式
        //shellSort(arr);
        //移位方式
        shellSort2(arr);

        Date date2 = new Date();
        long time2 = date2.getTime();
        System.out.println(time2-time1);

        //System.out.println(Arrays.toString(arr));
    }

    // 使用逐步推导的方式来编写希尔排序
    // 希尔排序时， 对有序序列在插入时采用交换法，
    // 思路(算法) ==> 代码
    public static void shellSort(int[] arr) {

        int temp = 0;
        // 根据前面的逐步分析， 使用循环处理
        for (int gap = arr.length / 2; gap > 0; gap /= 2) {
            for (int i = gap; i < arr.length; i++) {
                // 遍历各组中所有的元素(共gap组， 每组有个元素), 步长gap
                for (int j = i - gap; j >= 0; j -= gap) {
                    // 如果当前元素大于加上步长后的那个元素， 说明交换
                    if (arr[j] > arr[j + gap]) {
                        temp = arr[j];
                        arr[j] = arr[j + gap];
                        arr[j + gap] = temp;
                    }
                }
            }
        }
    }
}
```

```

    }
    }
}
//对交换式的希尔排序进行优化->移位法
public static void shellSort2(int[] arr) {
    // 增量gap, 并逐步的缩小增量
    for (int gap = arr.length / 2; gap > 0; gap /= 2) {
        // 从第gap个元素, 逐个对其所在的组进行直接插入排序
        for (int i = gap; i < arr.length; i++) {
            int j = i;
            int temp = arr[j];
            if (arr[j] < arr[j - gap]) {
                while (j - gap >= 0 && temp < arr[j - gap]) {
                    //移动
                    arr[j] = arr[j - gap];
                    j -= gap;
                }
                //当退出while后, 就给temp找到插入的位置
                arr[j] = temp;
            }
        }
    }
}

```

优化的希尔排序, 排序8w个数的数组, 排序时间是**16s**