

本文内容是对希尔排序的梳理和总结，本文内容包括：

## 希尔排序(Shell Sort)

算法步骤

举例说明

图解算法

代码实现

算法分析

## 排序算法十大经典方法

1. 冒泡排序：比较相邻元素并交换，每一轮将最大的元素移动到最后。
2. 选择排序：每一轮选出未排序部分中最小的元素，并将其放在已排序部分的末尾。
3. 插入排序：将未排序部分的第一个元素插入到已排序部分的合适位置
4. **希尔排序：改进的插入排序，将数据分组排序，最终合并排序**
5. 归并排序：将序列拆分成子序列，分别排序后合并，递归完成
6. 快速排序：选定一个基准值，将小于基准值的元素放在左边，大于基准值的元素放在右边，递归排序
7. 堆排序：将序列构建成一个堆，然后一次将堆顶元素取出并调整堆
8. 计数排序：统计每个元素出现的次数，再按照元素大小输出
9. 桶排序：将数据分到一个或多个桶中，对每个桶进行排序，最后输出
10. 基数排序：按照元素的位数从低到高进行排序，每次排序只考虑一位

# 1 希尔排序(Shell Sort)

希尔排序是按照其设计者希尔的名字命名的。希尔排序也是一种插入排序，它是插入排序经过改进之后的一个更高效的版本。

## 1.1 算法步骤

1. 先选定一个整数gap（gap是间隔、差距的意思）作为增量，一般情况下是将 $\text{length}/2$  作为增量，然后将所有距离为gap的元素分为一组，并对每一组进行插

入排序

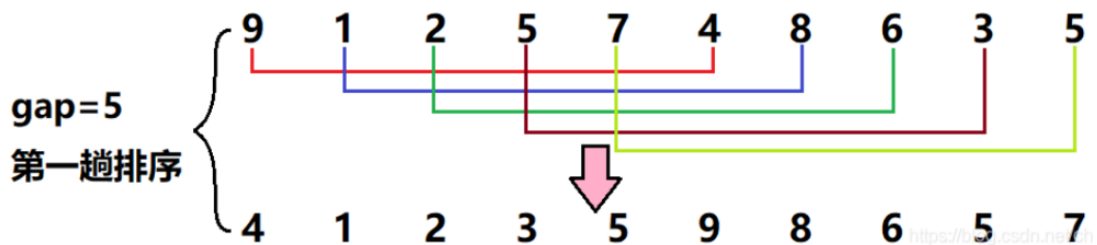
2. 缩小增量，让 $gap = gap/2$  然后重复第1步，直到 $gap$ 等于1停止

## 1.2 举例说明

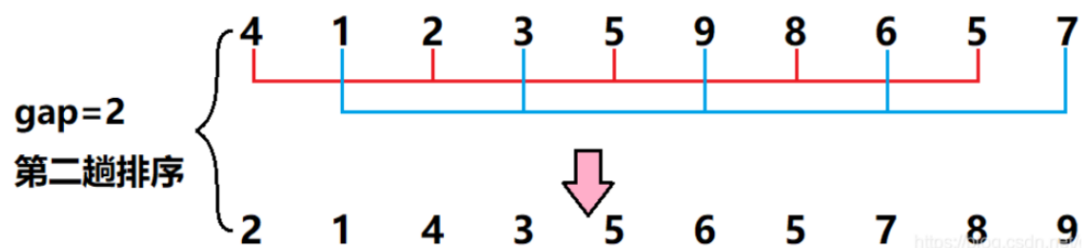
- 有数组如下，使用希尔排序该数组

9    1    2    5    7    4    8    6    3    5

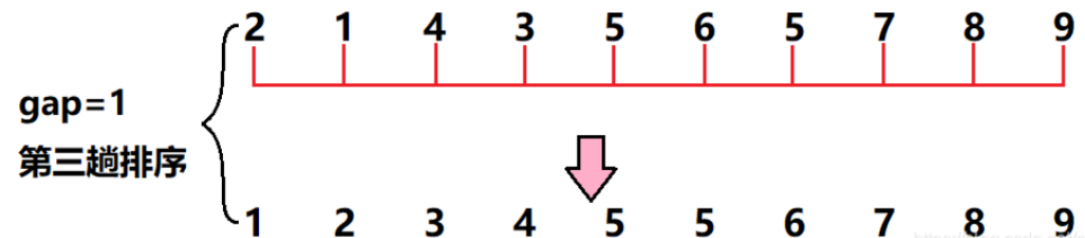
- 第一步：首先 $gap$ 取5，此时相隔距离为5的元素分到了一组（一共五组，每组两个元素），然后对每一组分别进行插入排序



- 第二步： $gap$ 折半为2，此时相隔距离为2的元素被分到了一组（一共两组，每组五个元素），然后对每一组分别进行插入排序

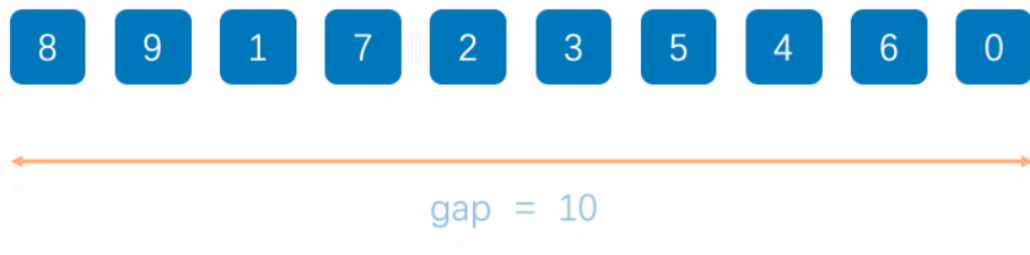


- 第三步： $gap$ 再次折半为1，此时所有元素被分到了一组，对它进行插入排序，至此插入排序完成



## 1.3 图解算法

如果下图不动，点击[这里](#)查看在线的图解



## 1.4 代码实现

```
1 public static int[] shellSort(int[] arr) {
2     for (int gap = arr.length / 2; gap > 0; gap /= 2) {
3         for (int i = gap; i < arr.length; i++) {
4             for (int j = i - gap; j >= 0; j -= gap) {
5                 if (arr[j] > arr[j + gap]) {
6                     int temp = arr[j];
7                     arr[j] = arr[j + gap];
8                     arr[j + gap] = temp;
9                 }
10            }
11        }
12    }
13    return arr;
14 }
```

## 1.5 算法分析

- **稳定性**：不稳定
- **时间复杂度**：最佳： $O(n\log^2 n)$ ，最差： $O(n\log^2 n)$ ，平均： $O(n\log n)$
- **空间复杂度**： $O(1)$