

# 希尔排序 (Shell Sort)

- 1959年由唐纳德·希尔 (Donald Shell) 提出
- 希尔排序把序列看作是一个矩阵，分成  $m$  列，逐列进行排序
  - $m$  从某个整数逐渐减为1
  - 当  $m$  为1时，整个序列将完全有序
- 因此，希尔排序也被称为递减增量排序 (Diminishing Increment Sort)
- 矩阵的列数取决于步长序列 (step sequence)
  - ✓ 比如，如果步长序列为{1,5,19,41,109,...}，就代表依次分成109列、41列、19列、5列、1列进行排序
  - ✓ 不同的步长序列，执行效率也不同

# 希尔排序 - 实例

- 希尔本人给出的步长序列是  $n/2^k$ , 比如  $n$  为16时, 步长序列是{1, 2, 4, 8}

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---

- 分成8列进行排序

16	15	14	13	12	11	10	9
8	7	6	5	4	3	2	1



8	7	6	5	4	3	2	1
16	15	14	13	12	11	10	9



8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	---

# 希尔排序 - 实例

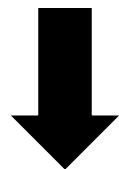
## ■ 分成4列进行排序

8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	---

8	7	6	5	4	3	2	1
4	3	2	1	16	15	14	13
16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5



4	3	2	1	8	7	6	5
8	7	6	5	12	11	10	9
12	11	10	9	16	15	14	13
16	15	14	13	4	3	2	1



4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13
---	---	---	---	---	---	---	---	----	----	----	---	----	----	----	----

# 希尔排序 - 实例

■ 分成2列进行排序

4	3	2	1	8	7	6	5	12	11	10	9	16	15	14	13
---	---	---	---	---	---	---	---	----	----	----	---	----	----	----	----

4	3
2	1
8	7
6	5
12	11
10	9
16	15
14	13



2	1
4	3
6	5
8	7
10	9
12	11
14	13
16	15

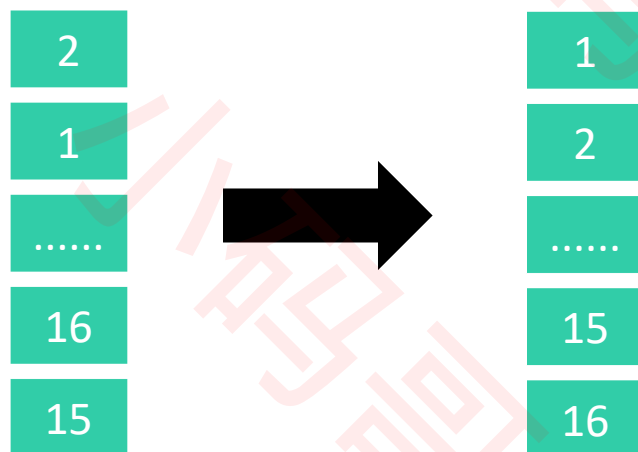


2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
---	---	---	---	---	---	---	---	----	---	----	----	----	----	----	----

# 希尔排序 - 实例

## ■ 分成1列进行排序

2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
---	---	---	---	---	---	---	---	----	---	----	----	----	----	----	----



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

■ 不难看出，从8列 变为 1列的过程中，逆序对的数量在逐渐减少

□ 因此希尔排序底层一般使用插入排序对每一列进行排序，也很多资料认为希尔排序是插入排序的改进版

# 希尔排序 - 实例

- 假设有11个元素，步长序列是{1, 2, 5}

11	10	9	8	7	6	5	4	3	2	1
----	----	---	---	---	---	---	---	---	---	---

11	10	9	8	7
----	----	---	---	---

6	5	4	3	2
---	---	---	---	---

1
---



1	5	4	3	2
---	---	---	---	---

6	10	9	8	7
---	----	---	---	---

11
----

- 假设元素在第 col 列、第 row 行，步长（总列数）是 step

- 那么这个元素在数组中的索引是  $col + row * step$

- 比如 9 在排序前是第 2 列、第 0 行，那么它排序前的索引是  $2 + 0 * 5 = 2$

- 比如 4 在排序前是第 2 列、第 1 行，那么它排序前的索引是  $2 + 1 * 5 = 7$

# 希尔排序 - 实现

```
List<Integer> stepSequence = sedgewickStepSequence();  
for (Integer step : stepSequence) {  
    sort(step);  
}
```

```
private void sort(int step) {  
    for (int col = 0; col < step; col++) {  
        for (int begin = col + step; begin < array.length; begin += step) {  
            int cur = begin;  
            while (cur > col && cmp(cur, cur - step) < 0) {  
                swap(cur, cur - step);  
                cur -= step;  
            }  
        }  
    }  
}
```

- 最好情况是步长序列只有1，且序列几乎有序，时间复杂度为  $O(n)$
- 空间复杂度为  $O(1)$ ，属于不稳定排序

# 希尔排序 - 步长序列

- 希尔本人给出的步长序列，最坏情况时间复杂度是  $O(n^2)$

```
private List<Integer> shellStepSequence() {  
    List<Integer> stepSequence = new ArrayList<>();  
    int step = array.length;  
    while ((step >>= 1) > 0) {  
        stepSequence.add(step);  
    }  
    return stepSequence;  
}
```



# 希尔排序 - 步长序列

■ 目前已知的最好的步长序列，最坏情况时间复杂度是  $O(n^{4/3})$ ，1986年由Robert Sedgewick提出

$$\begin{cases} 9 \left( 2^k - 2^{\frac{k}{2}} \right) + 1 & k \text{ even,} \\ 8 \cdot 2^k - 6 \cdot 2^{(k+1)/2} + 1 & k \text{ odd} \end{cases}$$

1, 5, 19, 41, 109, ...

```
private List<Integer> sedgewickStepSequence() {  
    List<Integer> stepSequence = new LinkedList<>();  
    int k = 0, step = 0;  
    while (true) {  
        if (k % 2 == 0) {  
            int pow = (int) Math.pow(2, k >> 1);  
            step = 1 + 9 * (pow * pow - pow);  
        } else {  
            int pow1 = (int) Math.pow(2, (k - 1) >> 1);  
            int pow2 = (int) Math.pow(2, (k + 1) >> 1);  
            step = 1 + 8 * pow1 * pow2 - 6 * pow2;  
        }  
        if (step >= array.length) break;  
        stepSequence.add(0, step);  
        k++;  
    }  
    return stepSequence;  
}
```