在希尔排序（Shell Sort）中，**增量序列（Gap Sequence）**的选择直接影响算法的效率。不同的增量序列会显著改变排序的时间复杂度。以下是**希尔增量序列**和**Hibbard增量序列**的详细解释及对比：

---

1. 希尔增量序列（Shell's Original Sequence）

定义：
希尔排序的原始增量序列由Donald Shell于1959年提出，按以下规则生成：

$$\text{Gap} = \left\lfloor \frac{n}{2^k} \right\rfloor \quad (k = 1, 2, 3, \ldots)$$

即：初始间隔为 $\left\lfloor \frac{n}{2} \right\rfloor$，后续每次折半，直到间隔为1。

示例（$n = 16$）：

$$8 \to 4 \to 2 \to 1$$

特点：

- **简单易实现**，但**时间复杂度较差**，最坏情况下仍为 $O(n^2)$。
- 增量之间可能存在**公因子**（如8、4、2、1），导致某些间隔的排序重复无效操作。

---

2. Hibbard增量序列（Hibbard's Sequence）

定义：
Hibbard增量序列由Thomas Hibbard提出，其形式为：

$$\text{Gap} = 2^k - 1 \quad (k = \lfloor \log_2 n \rfloor, \ldots, 1)$$

即：增量取小于 n 的形如 $1, 3, 7, 15, 31, \ldots$ 的数。

示例（$n = 16$）：

$$15 \to 7 \to 3 \to 1$$

特点：

- **显著提升效率**，最坏时间复杂度降至 $O(n^{3/2})$，平均接近 $O(n^{5/4})$。
- **避免重复无效比较**：增量之间互质（无公因子），减少了元素在不同间隔中被重复比较的情况。
- **理论支持**：数学证明该序列能有效减少逆序对。

---

3. 关键对比

增量序列	生成公式	示例（n=16）	时间复杂度	优势	劣势
希尔增量	$\left\lfloor \frac{n}{2^k} \right\rfloor$	8, 4, 2, 1	$O(n^2)$	简单	效率低，存在冗余揖
Hibbard增量	$2^k - 1$	15, 7, 3, 1	$O(n^{3/2})$	效率高，数学优化	计算增量稍复杂

---

4. 为什么增量序列影响效率？

希尔增量的问题：
若增量序列有公因子（如8、4、2、1），可能导致某些位置的元素在多个间隔中被重复比较，但未有效减少逆序对。

Hibbard增量的优化：

增量之间互质，确保元素在不同间隔中能"跳跃式"移动，更快接近最终位置。

增量按指数递减，平衡了远距离移动和局部微调的效率。

---

5. 其他常见增量序列

Sedgewick序列：结合数学优化，最坏 $O(n^{4/3})$，如 $1, 5, 19, 41, 109, \ldots$（混合 $9 \times 4^k - 9 \times 2^k + 1$ 和 $4^k - 3 \times 2^k + 1$）。

Knuth序列：$\frac{3^k - 1}{2}$（如1, 4, 13, 40, ...），时间复杂度 $O(n^{3/2})$。

---

6. 如何选择增量序列？

小规模数据：希尔增量（简单够用）。

追求性能：Hibbard或Sedgewick序列（需额外计算增量）。

理论兴趣：研究增量序列与时间复杂度的数学关系（如Pratt提出的质数序列）。

总结

希尔增量是基础版本，效率有限但易于实现。

Hibbard增量通过数学优化显著提升性能，适合对效率要求较高的场景。

核心思想：增量序列的设计决定了元素“跳跃”的智能性，是希尔排序优化的关键。