

排序

T E A C H I N G C O U R S E W A R E P O W P O I N T

授课时间：2025.08.05



目录

● PART-01 排序概念 TEACHING COURSEWARE

● PART-02 冒泡排序 TEACHING COURSEWARE

● PART-03 复杂度分析 TEACHING COURSEWARE

● PART-04 总结 TEACHING COURSEWARE

01

排序概念

TEACHING
COURSEWARE

TEACH

排序概念

排序指的是将一组无序的数据元素（如数字、字符、结构体等）按照某种预定的规则（通常是大小关系、字典顺序等）重新排列成有序序列的过程。有序序列可以是升序（从小到大）、降序（从大到小），也可以是根据自定义的比较规则排序（如按字符串长度、对象的某个属性等）。

排序的核心要素

1. 数据元素：需要排序的对象，可以是基本数据类型（如整数、浮点数），也可以是复杂数据类型（如数组、对象）。
2. 比较规则：定义元素之间的“顺序”，例如：
 - 数值型：按大小比较（如 $3 < 5$ ）；
 - 字符型：按 ASCII 码或字典顺序比较（如 $'a' < 'b'$ ）；
 - 自定义类型：按对象的某个属性比较（如按学生的成绩排序）。
3. 结果：经过排序后，数据元素需满足“对于任意相邻的两个元素，都符合比较规则”（如升序中前一个元素 \leq 后一个元素）

稳定排序

定义：如果排序算法能够保证相等元素的相对顺序在排序前后保持不变，则称该算法是稳定的。

02

冒泡排序

TEACHING
COURSEWARE

TEACH



冒泡排序

冒泡排序详解

基本概念

- 比较：通过相邻元素比较决定顺序
- 命名来源：较大元素像气泡一样"浮"到顶端
- 核心思想：重复遍历数组，每次将最大元素"冒泡"到末尾

算法步骤

1. 从数组第一个元素开始遍历
2. 比较相邻元素 $arr[j]$ 和 $arr[j+1]$
3. 如果逆序 ($arr[j] > arr[j+1]$)，则交换它们
4. 每轮遍历将当前最大值"冒泡"到末尾
5. 重复直到没有元素需要交换



冒泡排序

```
// 冒泡排序（下标从1开始）  
for (int i = 1; i < n; i++) {  
    for (int j = 1; j <= n - i; j++) {  
        if (a[j] > a[j + 1]) {  
            // 交换元素  
            swap(a[j], a[j+1]);  
        }  
    }  
}
```



冒泡排序

步骤	操作	数组状态
初始		[5, 1, 4, 2, 8]
第1轮第1次	比较5和1 → 交换	[1, 5, 4, 2, 8]
第1轮第2次	比较5和4 → 交换	[1, 4, 5, 2, 8]
第1轮第3次	比较5和2 → 交换	[1, 4, 2, 5, 8]
第1轮第4次	比较5和8 → 保持	[1, 4, 2, 5, 8]
第2轮第1次	比较1和4 → 保持	[1, 4, 2, 5, 8]
第2轮第2次	比较4和2 → 交换	[1, 2, 4, 5, 8]
第2轮第3次	比较4和5 → 保持	[1, 2, 4, 5, 8]
第3轮第1次	比较1和2 → 保持	[1, 2, 4, 5, 8]
第3轮第2次	比较2和4 → 保持	[1, 2, 4, 5, 8]
结束	检测无交换 → 终止	[1, 2, 4, 5, 8]

02

复杂度分析

TEACHING
COURSEWARE

TEACH



冒泡排序

外层循环：执行 $n-1$ 次（ i 从 1 到 $n-1$ ）

内层循环：第 i 轮时，需要比较 $n-i$ 次（ j 从 1 到 $n-i$ ）

总操作次数为：

$$(n-1) + (n-2) + (n-3) + \dots + 1 = n*(n-1)/2$$

04

总结

TEACHING
COURSEWARE

TEACH



总结

冒泡排序是不是稳定排序，如何保证 稳定。

[5, 1, 4, 2, 8]

第一轮结束后是什么样的，描述一下冒泡过程。