

24算法暑期基础培训第二讲

l10k

2024 年 7 月 10 日

基础篇

冒泡排序

- ▶ 算法描述：重复地走访过要排序的数列，一次比较两个元素，如果它们的顺序错误就把它们交换过来。
- ▶ 时间复杂度： $O(n^2)$
- ▶ 空间复杂度： $O(1)$
- ▶ 稳定性：稳定

直接插入排序

- ▶ 算法描述：从未排序区间中取出第一个元素，插入到已排序区间中的适当位置，直到所有元素均已排序。
- ▶ 时间复杂度： $O(n^2)$
- ▶ 空间复杂度： $O(1)$
- ▶ 稳定性：稳定

基础篇

希尔排序

- ▶ 算法描述：先将整个待排序的记录序列分割成为若干子序列分别进行直接插入排序，待整个序列中记录基本有序时，再对全体记录进行一次直接插入排序。
- ▶ 时间复杂度：最坏情况 $O(n^2)$ ，最优情况 $O(n \log^2 n)$
- ▶ 空间复杂度： $O(1)$
- ▶ 稳定性：不稳定

选择排序

- ▶ 算法描述：每次从待排序的数据元素中选出最小（或最大）的一个元素，存放在序列的起始位置，直到全部待排序的数据元素排完。
- ▶ 时间复杂度： $O(n^2)$
- ▶ 空间复杂度： $O(1)$
- ▶ 稳定性：不稳定

进阶篇

快速排序

- ▶ 算法描述：通过一趟排序将待排记录分隔成独立的两部分，其中一部分的所有记录均比另一部分的所有记录小，然后分别对这两部分继续进行排序，直到整个序列有序。
- ▶ 时间复杂度：最坏情况 $O(n^2)$ ，平均 $O(n \log n)$
- ▶ 空间复杂度： $O(\log n)$
- ▶ 稳定性：不稳定

归并排序

- ▶ 算法描述：将序列分为若干子序列，每个子序列排好序后再将其合并成一个整体序列。
- ▶ 时间复杂度： $O(n \log n)$
- ▶ 空间复杂度： $O(n)$
- ▶ 稳定性：稳定

堆的基本操作

- ▶ 插入一个数
- ▶ 求集合当中的最小值或最大值
- ▶ 删除最小值或最大值
- ▶ 删除任意一个元素
- ▶ 修改任意一个元素

堆的实现

- ▶ 最小堆：每个节点都小于等于其子节点。
- ▶ 最大堆：每个节点都大于等于其子节点。

堆的应用

- ▶ 优先队列：实现优先队列常用堆结构。

基本操作

- ▶ top: 访问队头元素
- ▶ empty: 队列是否为空
- ▶ size: 返回队列内元素个数
- ▶ push: 插入元素到队尾（并排序）
- ▶ emplace: 原地构造一个元素并插入队列
- ▶ pop: 弹出队头元素
- ▶ swap: 交换内容

优先队列的实现

- ▶ 基于堆的实现：优先队列通常使用二叉堆来实现。

手写二分查找

- ▶ 算法描述：在一个有序数组中，通过不断地将查找范围减半来定位目标值。
- ▶ 时间复杂度： $O(\log n)$
- ▶ 空间复杂度： $O(1)$

二分答案

- ▶ 应用场景：通常用于解决在一个连续的数值区间内查找满足某个条件的最优值的问题。

STL中的二分查找

- ▶ `lower_bound()`: 返回大于或等于目标值的第一个位置
- ▶ `upper_bound()`: 返回大于目标值的第一个位置
- ▶ `binary_search()`: 若目标值存在则返回true, 否则返回false