

# 数据结构与算法专题

## 基于堆实现优先级队列

### *PriorityQueue*

作者：Plane老师

微信：PlaneZhong

官网：[www.qiqiker.com](http://www.qiqiker.com)

邮箱：[1785275942@qq.com](mailto:1785275942@qq.com)

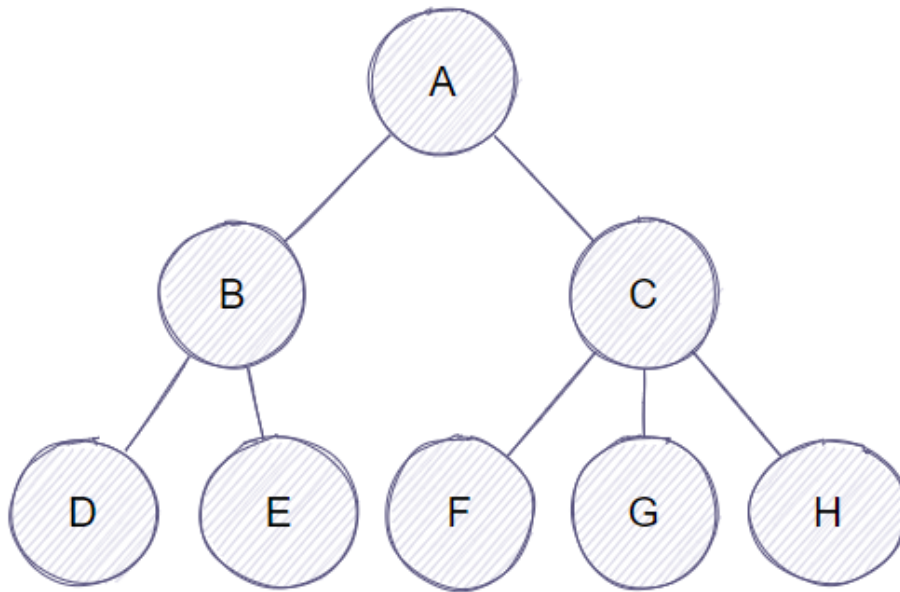
### 课程介绍

- 优先级队列与普通队列的差别
- 优先级队列的运用举例
  - 海量定时器
  - 实现搜索引擎的热搜目录
  - 寻路搜索算法
- 优先级队列实现思路
  - 基于二叉堆数据结构来实现
- 课程内容介绍
  - 用到的基础的数据结构概念
    - 树、二叉树
    - 完全二叉树
    - 完全二叉树的特性及存储方式
    - 堆、小顶堆、大顶堆
  - 实现堆的基础结构与操作
    - 建堆
    - 移除堆顶节点
    - 堆化
  - 开发案例测试
    - 数据项继承Comparable接口
    - 通过泛型让优先级队列便于使用

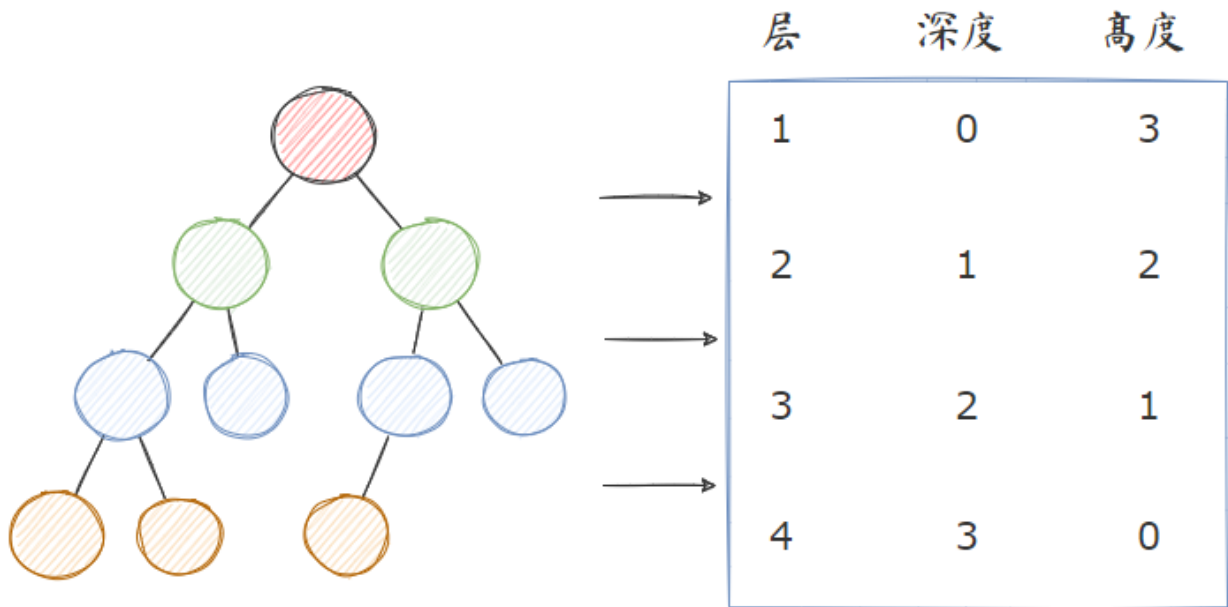
- 编写测试案例
  - 补充说明
    - java, c++等语言中都内置实现了优先级队列，.net6.0中也有了官方实现。
    - Unity对C#高版本支持有限，目前在Unity主流版本中必须自己开发。
    - 即使已经有了官方的内置实现，学习其实现原理能加深对基础数据结构的理解。
    - 便于对实现功能扩展，满足特殊需求。
- 

## 二叉树与堆的基础概念

### 1. 什么是树？

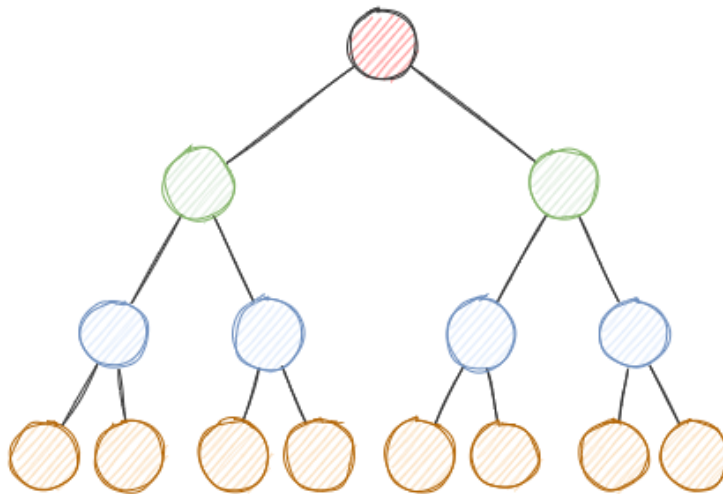


2. 树的几个重要描述属性：**层**、深度、高度。

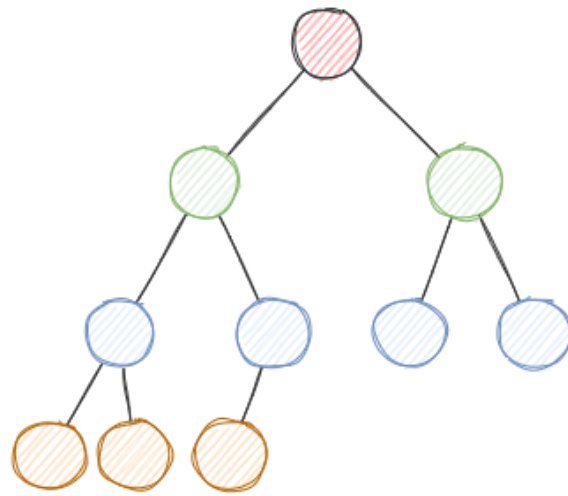


3. 二叉树：每个节点**最多**只能有2个子节点的树

- 满二叉树：叶子节点全在**底层**，且都有左右两个子节点

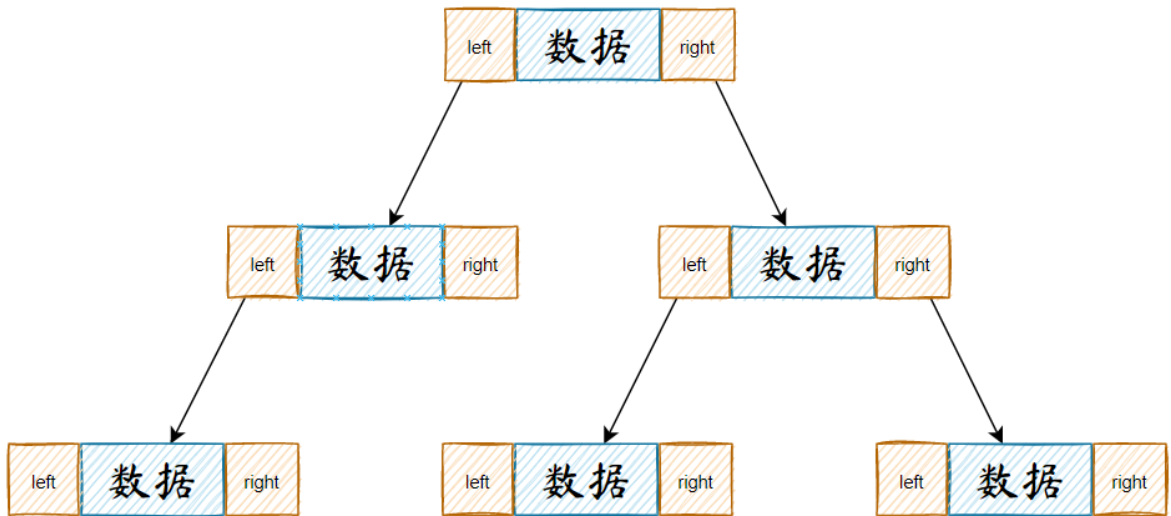


- 完全二叉树：叶子节点在**最下两层**，最后一层叶子节点**靠左**排列，除了最后一层，其它层节点个数必须为**最大**

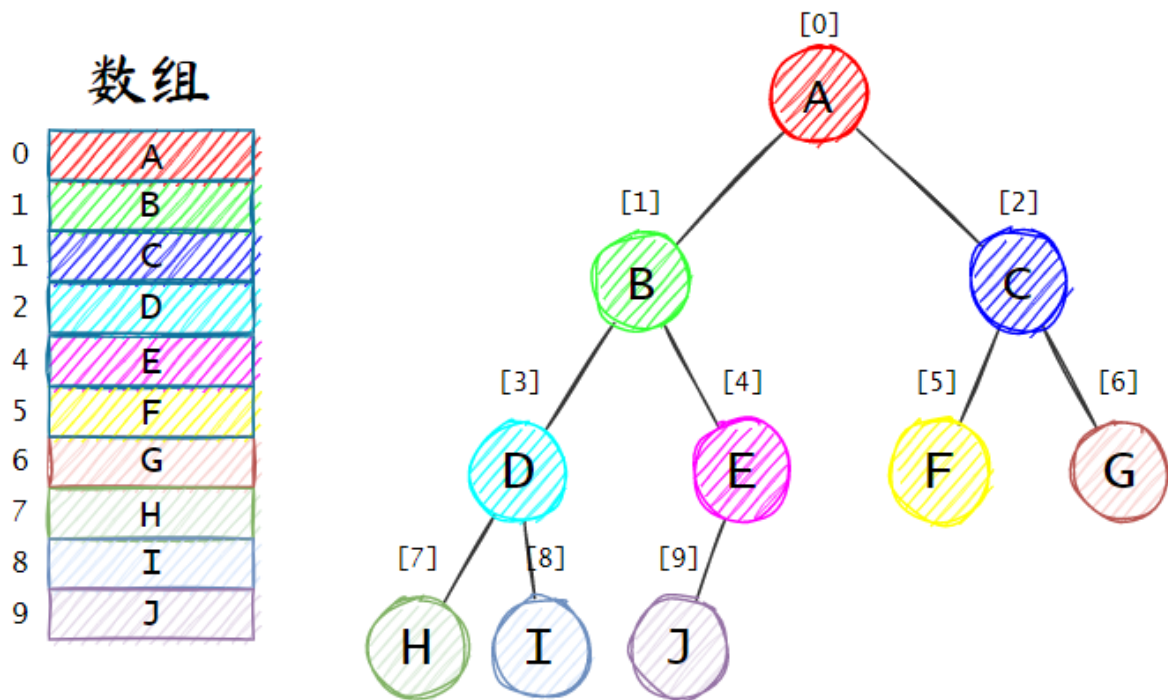


#### 4. 二叉树的存储方法

- **链式存储：**



- 数组存储：



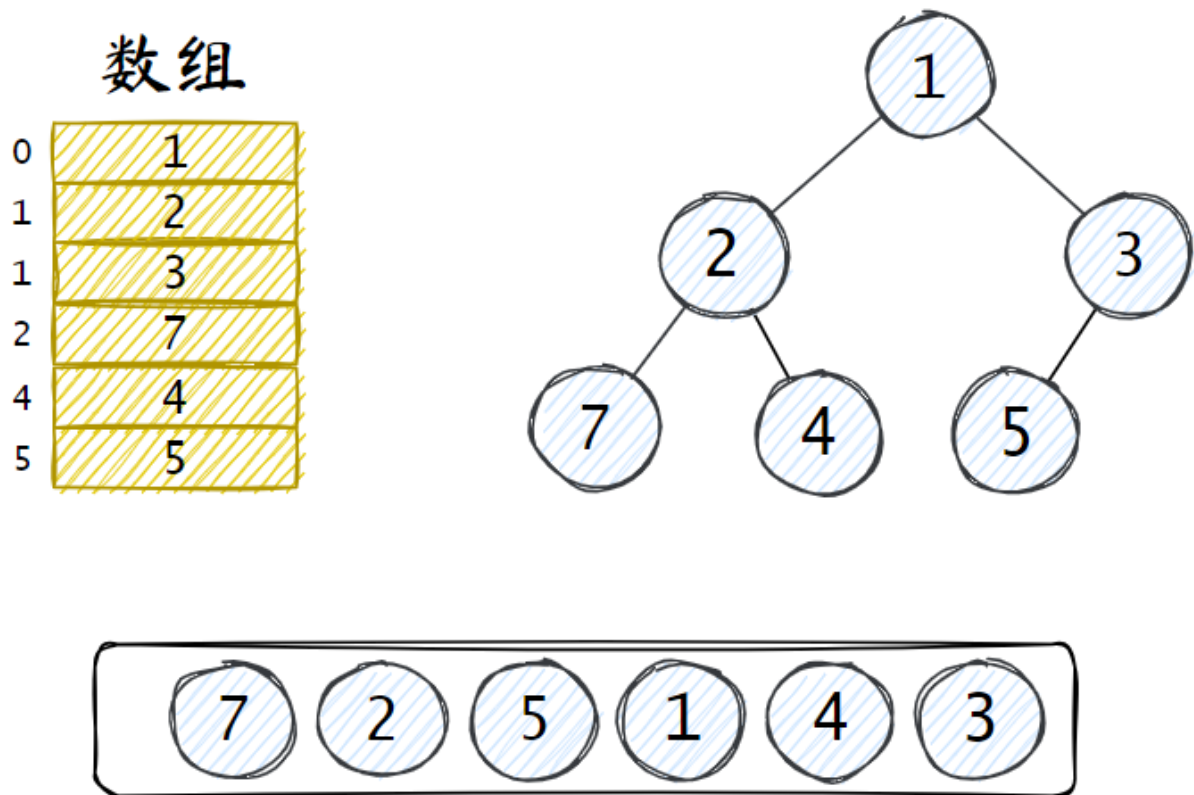
- 数组存储对比链式存储的优势：
  - 更节省内存空间
  - 索引更方便快捷（特别像完全二叉树这类结构）
- 完全二叉树使用数组来存储最为节省内存，不需要额外的空间来指向左右子节点，而且存储中不存在空洞。这也是为什么完全二叉树要求除了最后一层其它层节点必须为最大，且最后一层叶子节点**靠左排列**的原因。
- 完全二叉树用数组存储时位置索引的重要特性：
  - 左节点的位置索引： $\text{leftIndex} = 2 * \text{parentIndex} + 1$
  - 右节点的位置索引： $\text{rightIndex} = 2 * \text{parentIndex} + 2$
  - 父节点的位置索引： $\text{parentIndex} = (\text{childIndex} - 1) / 2$

## 5. 堆的相关定义

- 堆是一种特殊的树，树满足下列两点就是一个堆：
  - 是**完全二叉树**
  - 树中所有节点的值都**大于/小于**等于子节点中的值。
- **大顶堆**：所有节点的值都大于等于子节点值
- **小顶堆**：所有节点的值都小于等于子节点值

## 创建堆的基础结构

- 创建堆数据结构的实现思路：（以小顶堆来举例）



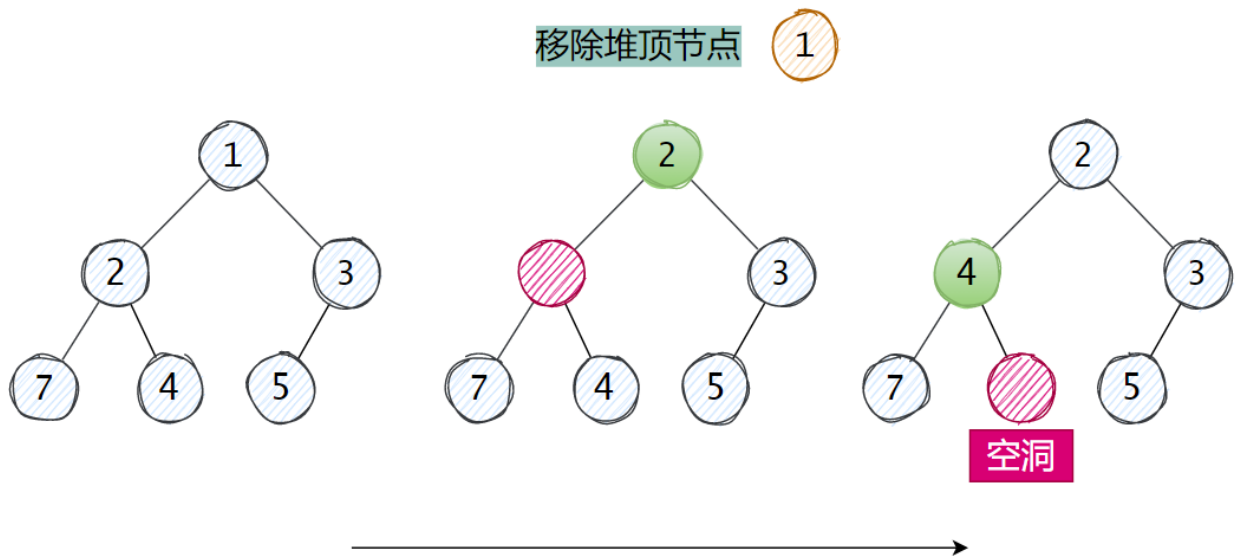
- 将需要放入堆中的数据逐个添加到一个空堆中
- 新插入的节点放到堆的**最末尾**，再让插入节点与父节点比较大小，父节点大于子节点时就**互换位置**，重复这个过程，直到满足堆的大小关系条件。
- 往堆中插入节点之后调整保持堆的特性，这个过程就叫「堆化」（heapify）
- 数据存储使用List来替代数组以便于自动扩容
  - List的底层也是使用**数组**来实现，默认容量为4
  - [List的源码实现](#)

---

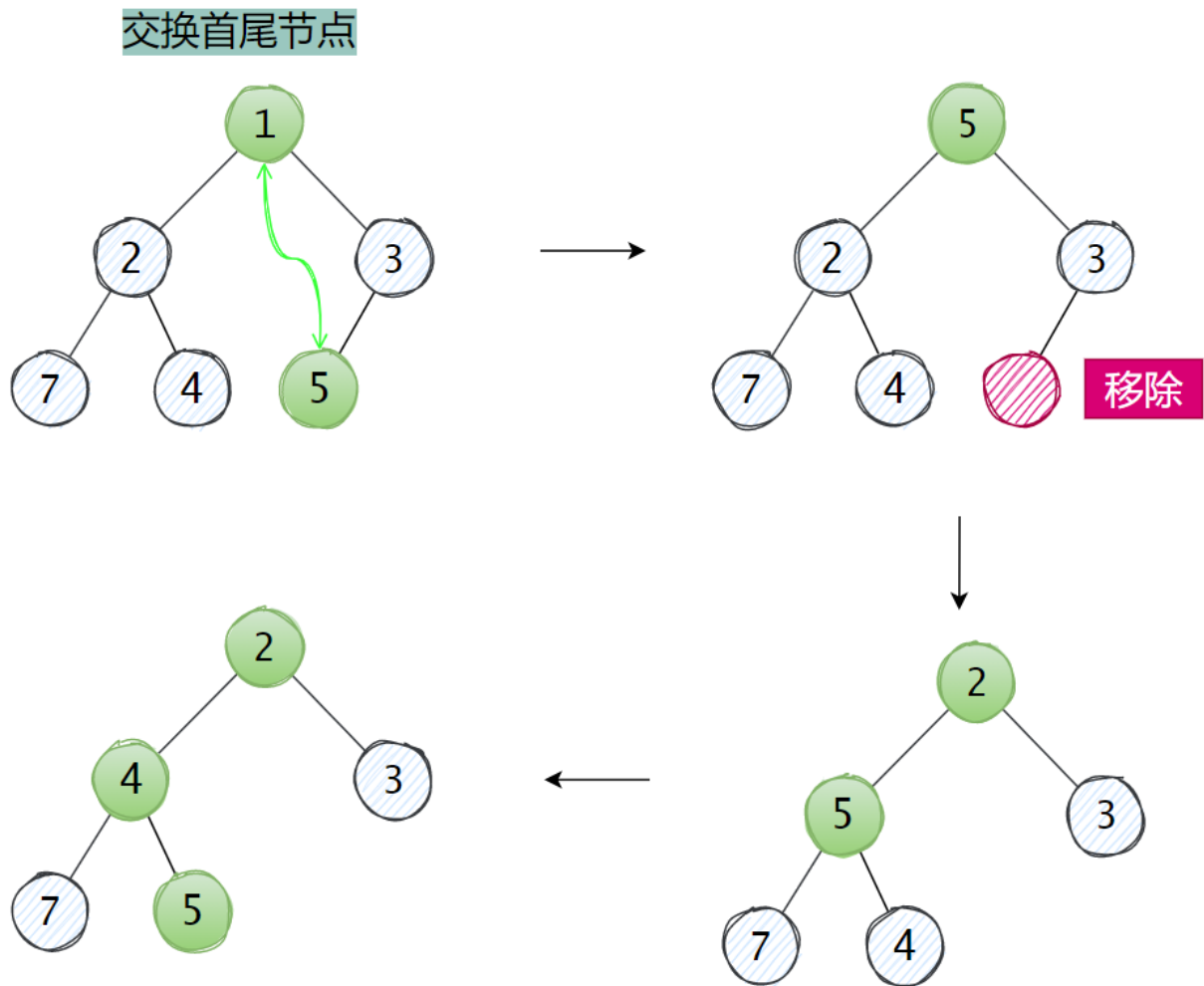
## 移除堆顶节点

- 移除堆顶节点  
堆的定义决定了堆顶的元素是最大值或最小值，移除堆顶的最小值或最大值时也需要保持堆的特性。同样需要进行「堆化」处理。

- 演示空洞问题



- 解决空洞问题思路：交换首尾节点，再移除尾节点，再从上到下「堆化」处理



- 数据项实现`Comparable<T>`接口，用于比较优先级
- [Comparable官方文档](#)
- 完善常用API
- 编写测试案例

## 扩展移除数据项

- 堆化方向判定

