


# 算法与数据结构体系课程

liuyubobobo

# 堆和优先队列

# 实践：基于堆实现优先队列

# 优先队列

Interface Queue<E>       PriorityQueue<E>  
implement

- void enqueue(E)
- E dequeue()
- E getFront()
- int getSize()
- boolean isEmpty()

可以使用不同的底层实现

# 实践：基于堆的优先队列实现

# Leetcode 上优先队列相关的问题

# 优先队列的经典问题

top K 问题

在  $N$  个元素中选出最大的  $K$  个元素

在  $N$  个元素中选出最小的  $K$  个元素

select K 问题

# 在 N 个元素中选出最小的 K 个元素

剑指 Offer 40. 最小的k个数

难度 简单

123



输入整数数组 `arr`，找出其中最小的 `k` 个数。例如，输入4、5、1、6、2、7、3、8这8个数字，则最小的4个数字是1、2、3、4。

示例 1：

输入：arr = [3,2,1], k = 2

输出：[1,2] 或者 [2,1]

示例 2：

输入：arr = [0,1,2,1], k = 1

输出：[0]

限制：

- `0 <= k <= arr.length <= 10000`
- `0 <= arr[i] <= 10000`



# 优先队列的经典问题

top K 问题      在  $N$  个元素中选出最小的  $K$  个元素

使用优先队列，维护当前看到的最小的  $K$  个元素

对于每一个新的数据，如果比这  $K$  个最小元素中最大的还小，  
则替换

使用最大堆

# 实践：使用优先队列完成剑指 Offer 40

<https://leetcode-cn.com/problems/zui-xiao-de-kge-shu-lcof/>

作业：Top K, N 个元素选出最大的 K 个元素

# Leetcode 215

# 作业解析： Leetcode 215

# Java标准库中的优先队列

实践：使用 Java 标准库中的优先队列

# Java标准库中的优先队列

Comparator

Lambda



# 使用快排思想和优先队列的比较

# 使用快排思想和优先队列的比较

top K, select K 问题

既可以用快排思想解决，又可以用优先队列解决

快排：时间： $O(n)$ ；空间： $O(1)$

优先队列：时间： $O(n\log k)$ ；空间： $O(k)$

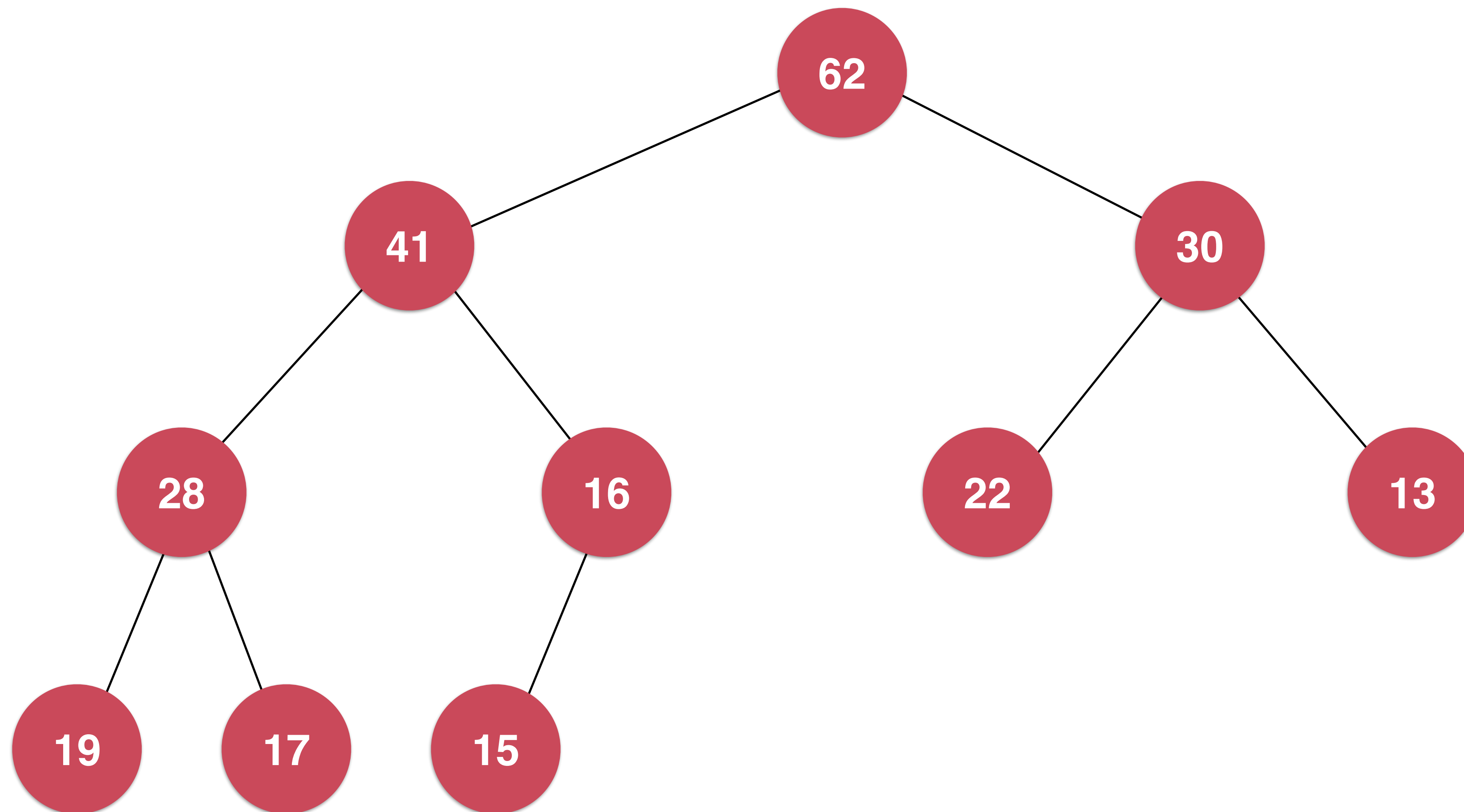
优先队列的优势：不需要一次性知道所有数据

数据流

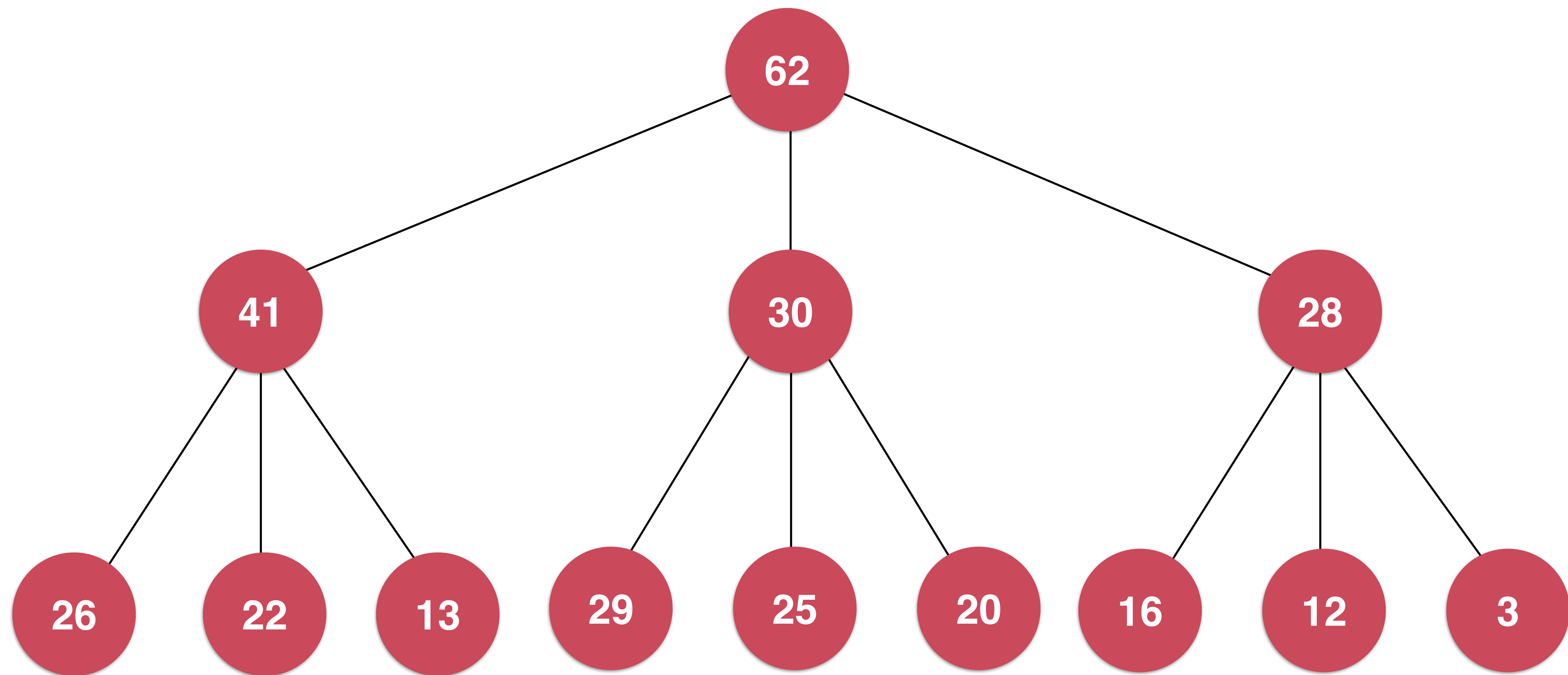
极大规模数据

更多和堆，队列相关的话题

# 二叉堆 Binary Heap



# d 叉堆 d-ary heap



# 索引|堆

二项堆

斐波那契堆

# 广义队列



# 广义队列

Interface Queue<E>

- void enqueue(E)
- E dequeue()
- E getFront()
- int getSize()
- boolean isEmpty()

# 广义队列

普通队列，优先队列

栈，也可以理解成是一个队列

随机队列

# 其他

欢迎大家关注我的个人公众号：是不是很酷



# 算法与数据结构体系课程

liuyubobobo