EDU
CSDN学院 IT实战派

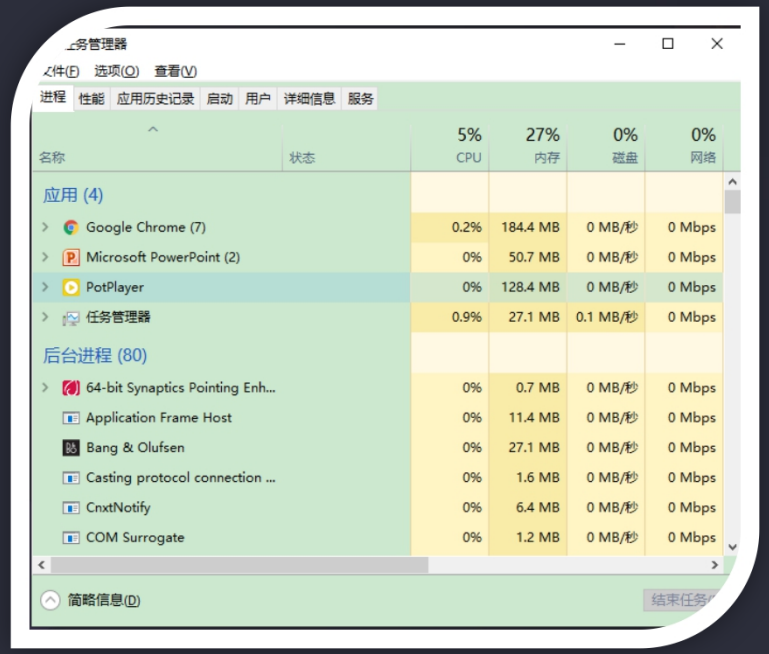
图解数据结构和算法

优先队列和堆

讲师：Samuel

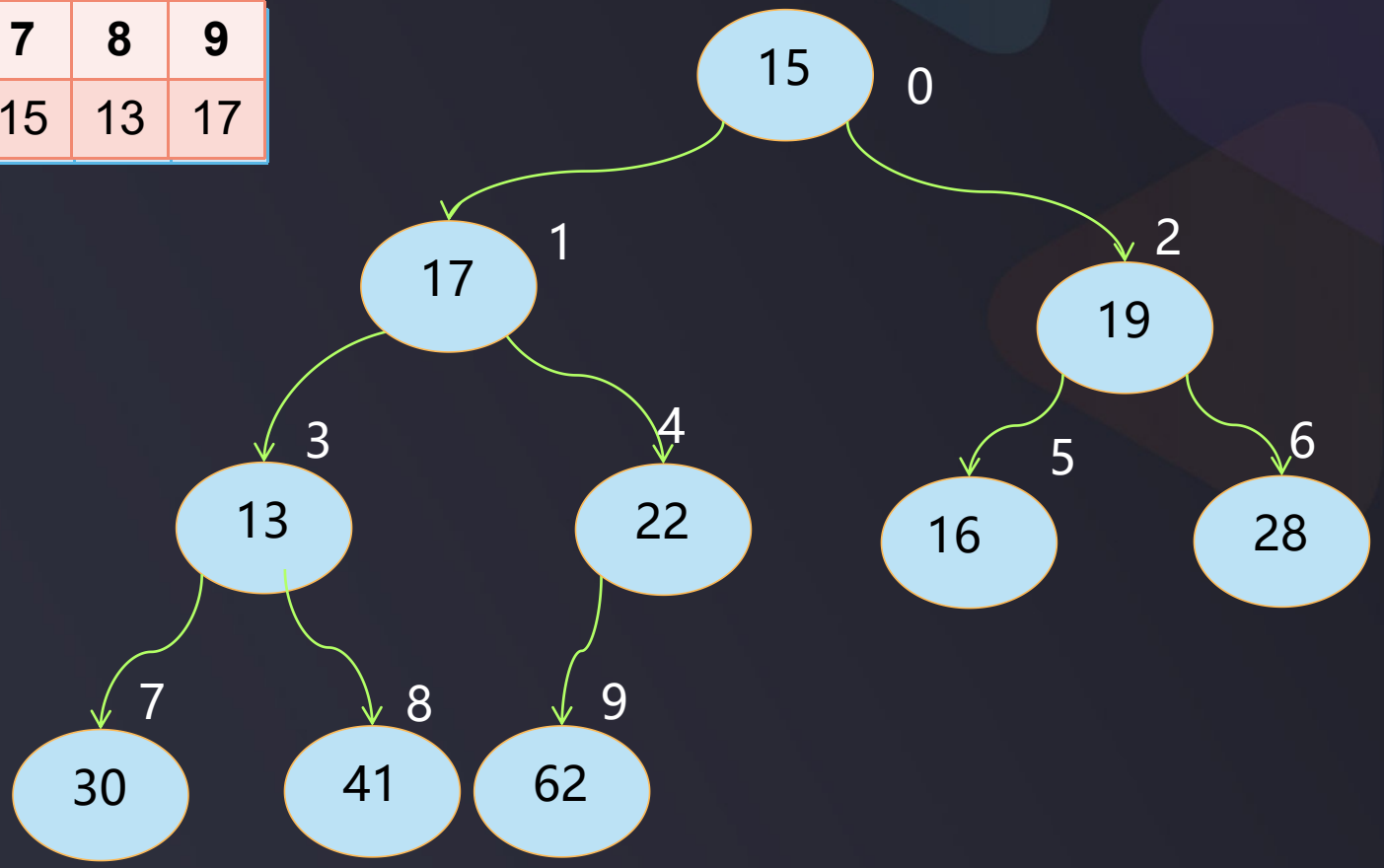
本章概述

- 优先队列无处不在
  - 医院看病
  - 银行VIP客户的插队
  - 操作系统的任务动态分配



本章概述

0	1	2	3	4	5	6	7	8	9
62	41	28	30	22	16	19	15	13	17



Content

优先队列

1 什么是优先队列

- 普通的队列是一种先进先出的数据结构，元素在队列尾追加，而从队列头删除
- 在优先队列中，元素被赋予优先级，当访问元素时具有最高优先级的元素最先出队
- 优先队列具有最高级先出 （first in, largest out）的行为特征

优先队列

2 优先队列的定义

```
public interface Queue<E> {
    void  enqueue(E e);
    E dequeue();
    E getFront();
    int getSize();
    boolean isEmpty();
}
```

3 使用不同的底层实现

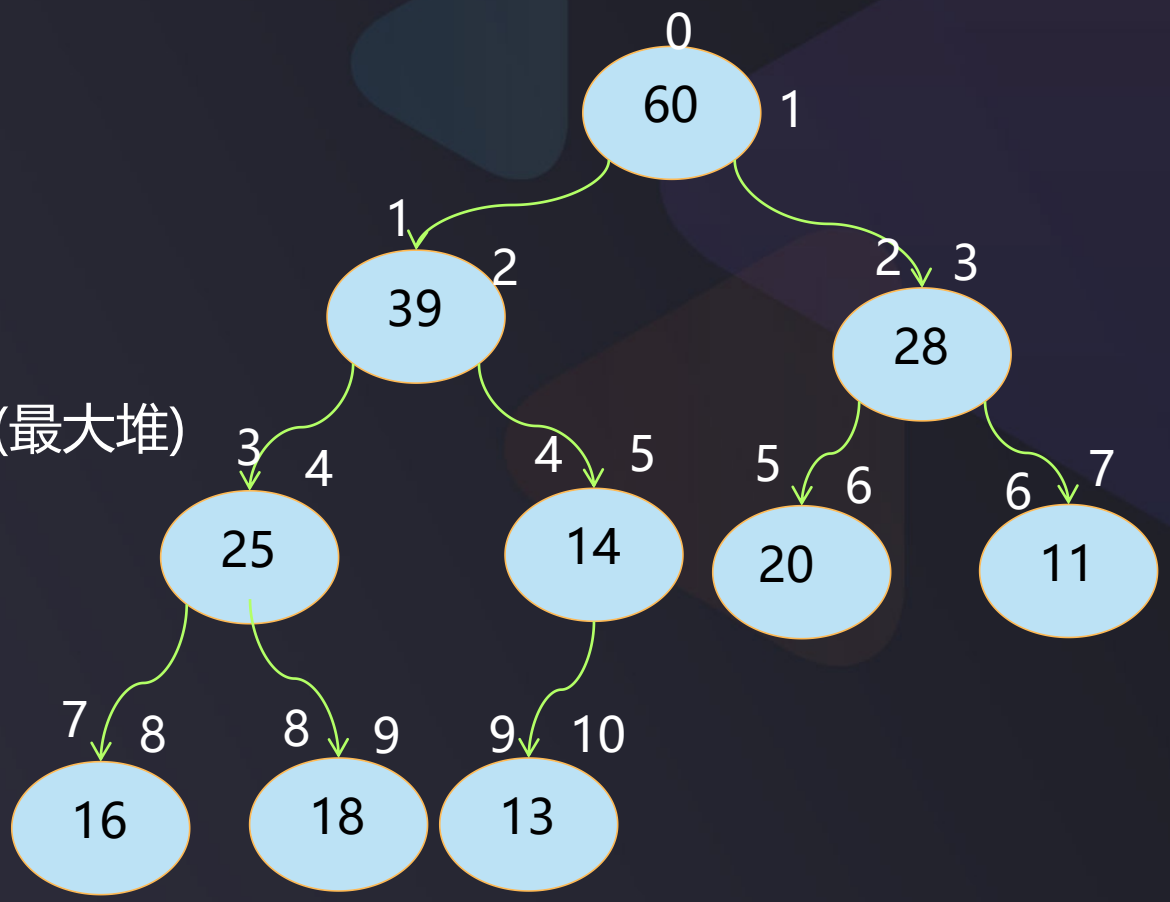
	入队	出队
普通线性结构	O(1)	O(n)
顺序线性结构	O(n)	O(1)
堆	O(logn)	O(logn)

堆

1 堆的基本结构

- 二叉堆是满足一些特殊性质的二叉树
  - 二叉堆是一棵完全二叉树
  - 堆中任一节点的值总是大于等于其孩子节点值(最大堆)
- 可以使用数组的形式来表示二叉堆
- left(i)=2*i　　right(i)=2*i+1　　parent(i)=i/2
- 如果数组的下标是从0开始的，那么公式变形为

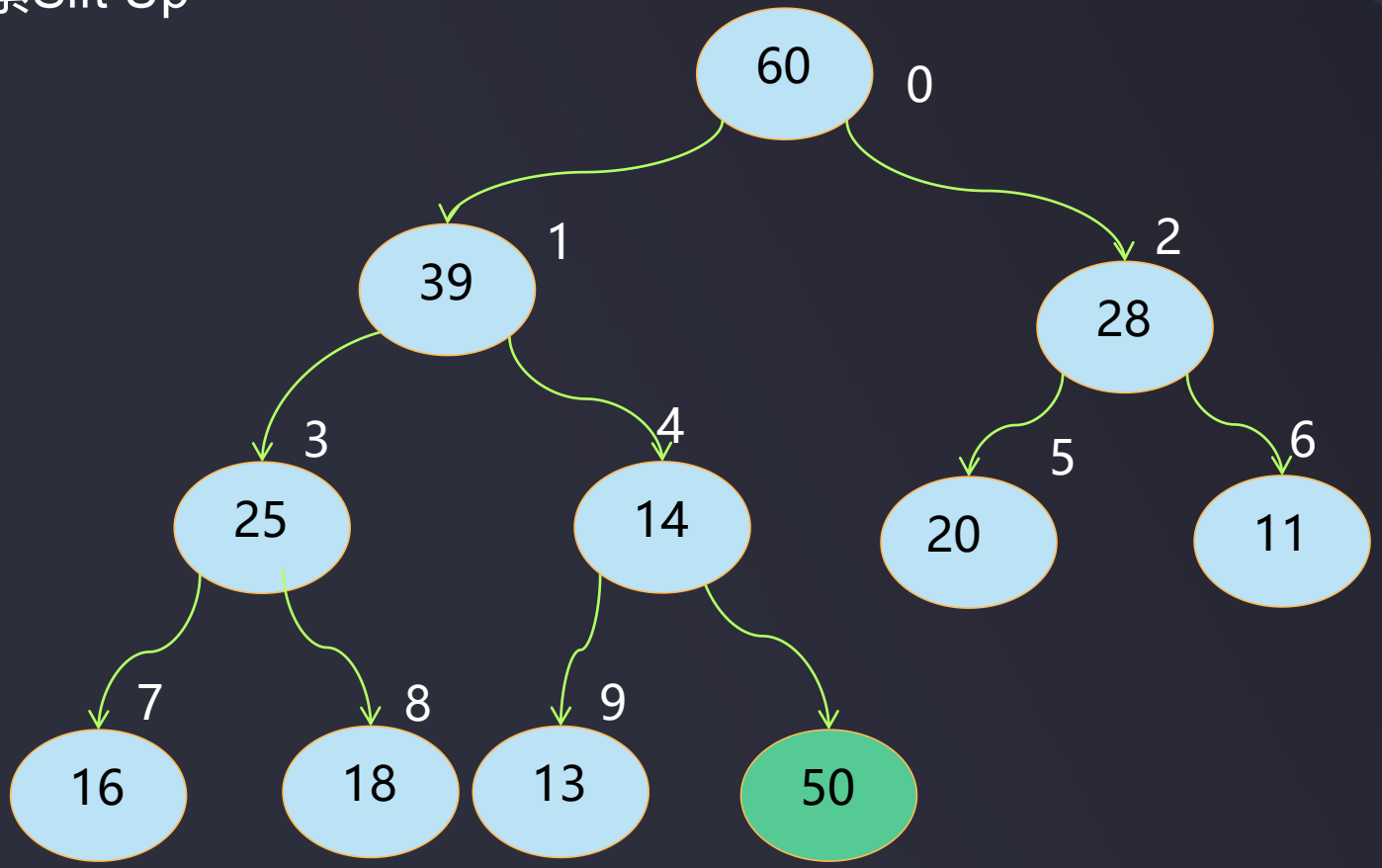
  left(i)=2*i+1 right(i)=2*i+2　parent(i)=(i-1)/2



0	1	2	3	4	5	6	7	8	9	10
60	39	28	25	14	20	11	16	18	13	13

堆

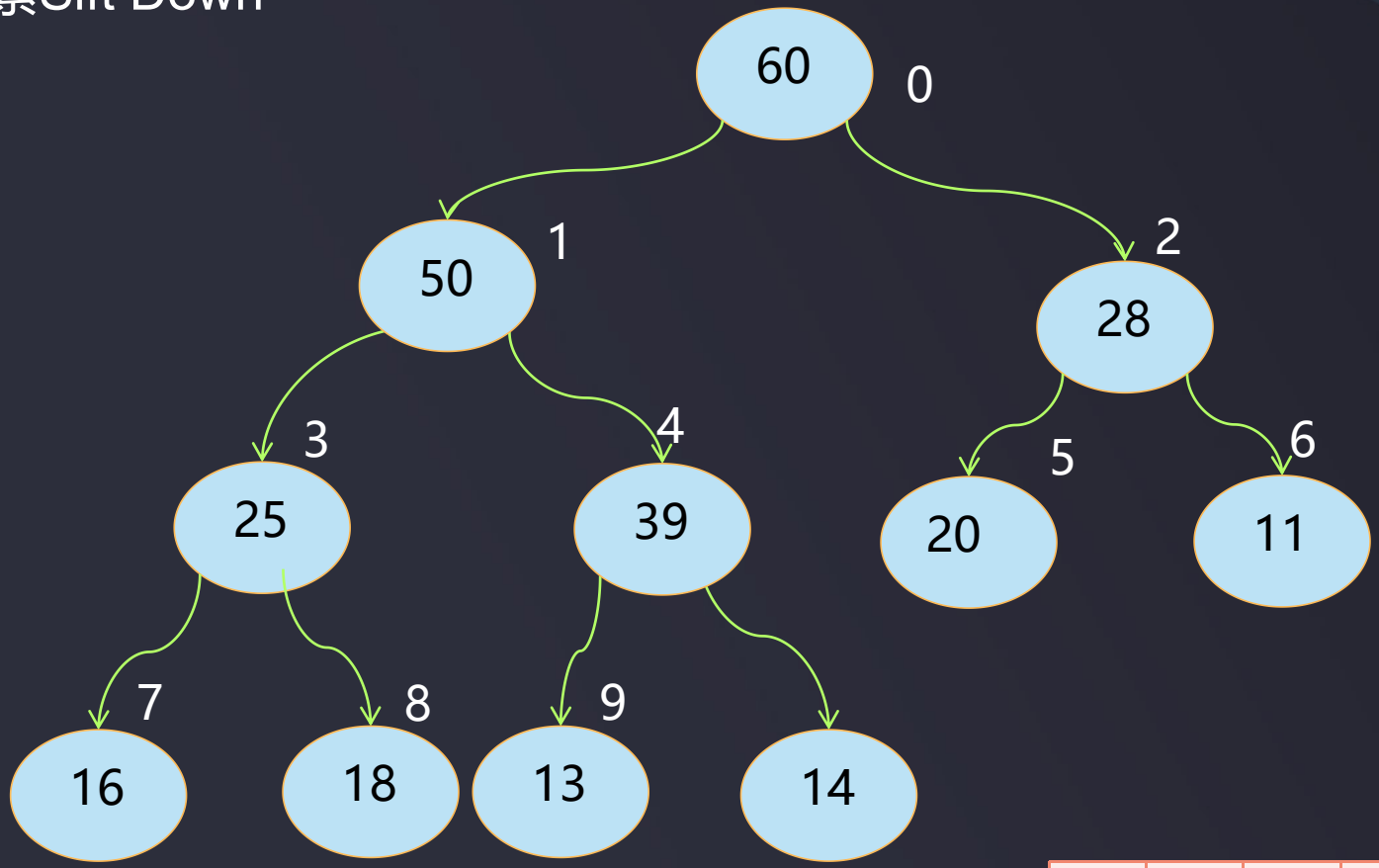
2 添加元素Sift Up



0	1	2	3	4	5	6	7	8	9	10
60	50	28	25	39	20	11	16	18	13	50

堆

3 取出元素Sift Down



0	1	2	3	4	5	6	7	8	9	10
50	39	28	25	14	20	11	16	18	13	

堆

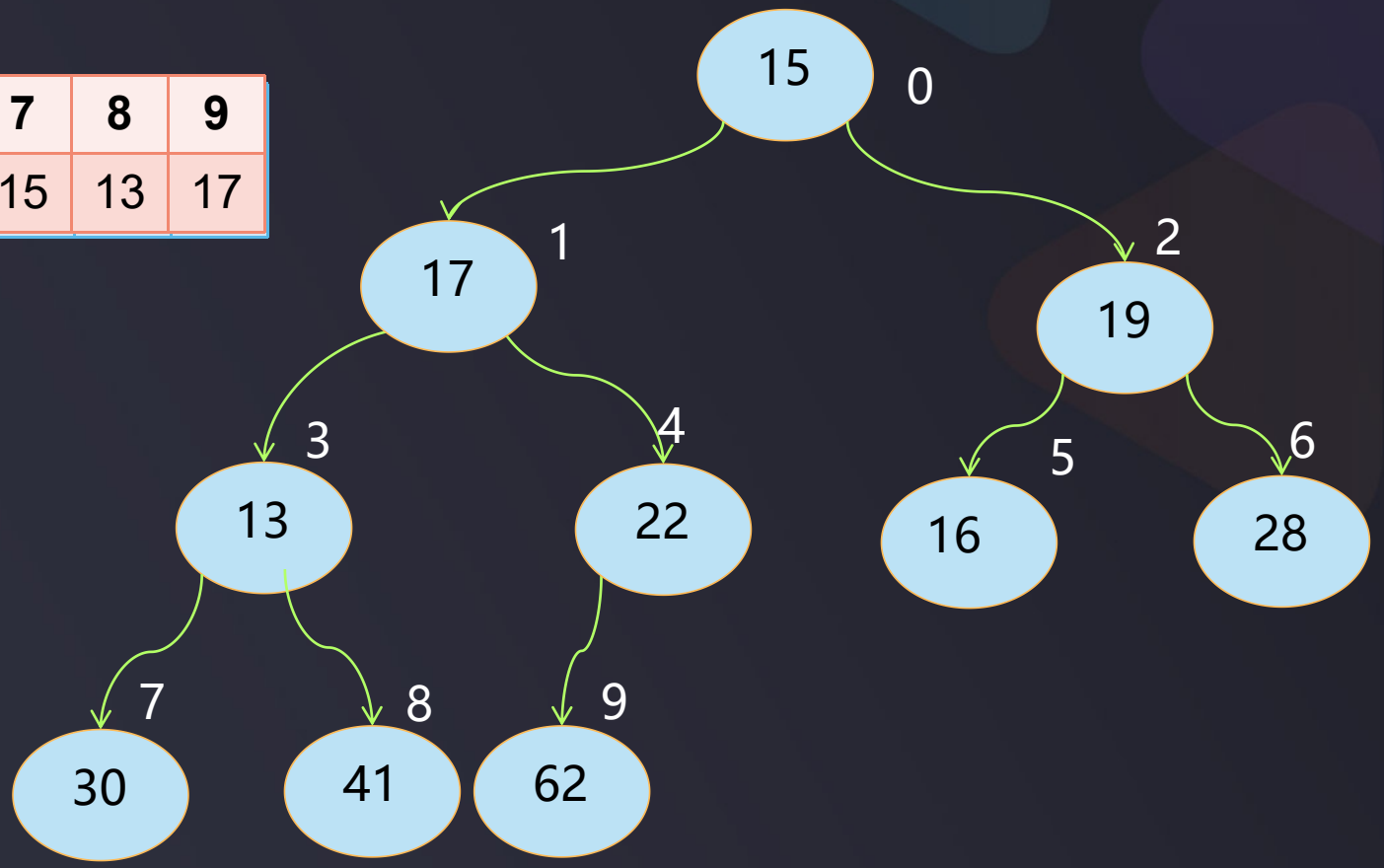
4 Replace和Heapify

- Replace:取出最大元素后，放入一个新元素
  - 实现一：先取出最大元素再添加元素，即Sift Down和Sift Up，就经历过了两次O(logn)的操作
  - 实现二：Replace操作
- Heapify:将任意数组整理成堆
  - 实现一：扫描一遍数组，将扫描出来的数组元素添加到堆中O(nlogn)
  - 实现二：Heapify操作O(n)

堆

5 演示Heapify

0	1	2	3	4	5	6	7	8	9
62	41	28	30	22	16	19	15	13	17



堆

6 优先队列和堆的应用

- 在n个元素中选出前m名
  - 如果m=1,那么直接遍历一遍即可，时间复杂度就是O(n)
  - n个元素排序，再选出前m个元素也可以完成，时间复杂度是O(nlogn)
  - 如果使用优先队里，在O(nlogm)的时间复杂度中即可以完成该操作
- 给定一个非空的整数数组，返回其中出现频率前 k 高的元素
  - 输入: nums = [1,1,1,2,2,3], k = 2
  - 输出: [1,2]
- Java中已经存在了PriorityQueue

EDU

CSDN学院 IT实战派

下节课再见，记得关注公众号