

# 极客大学算法训练营

## 堆 Heap 和 二叉堆 Binary Heap

覃超

Sophon Tech 创始人，前 Facebook 工程师

# 定义

# 堆 Heap

Heap: 可以迅速找到一堆数中的最大或者最小值的数据结构。

将根节点最大的堆叫做大顶堆或大根堆，根节点最小的堆叫做小顶堆或小根堆。  
常见的堆有二叉堆、斐波那契堆等。

假设是大顶堆，则常见操作（API）：

find-max:  $O(1)$   
delete-max:  $O(\log N)$   
insert (create):  $O(\log N)$  or  $O(1)$

不同实现的比较：[https://en.wikipedia.org/wiki/Heap\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Heap_(data_structure))

# 二叉堆性质

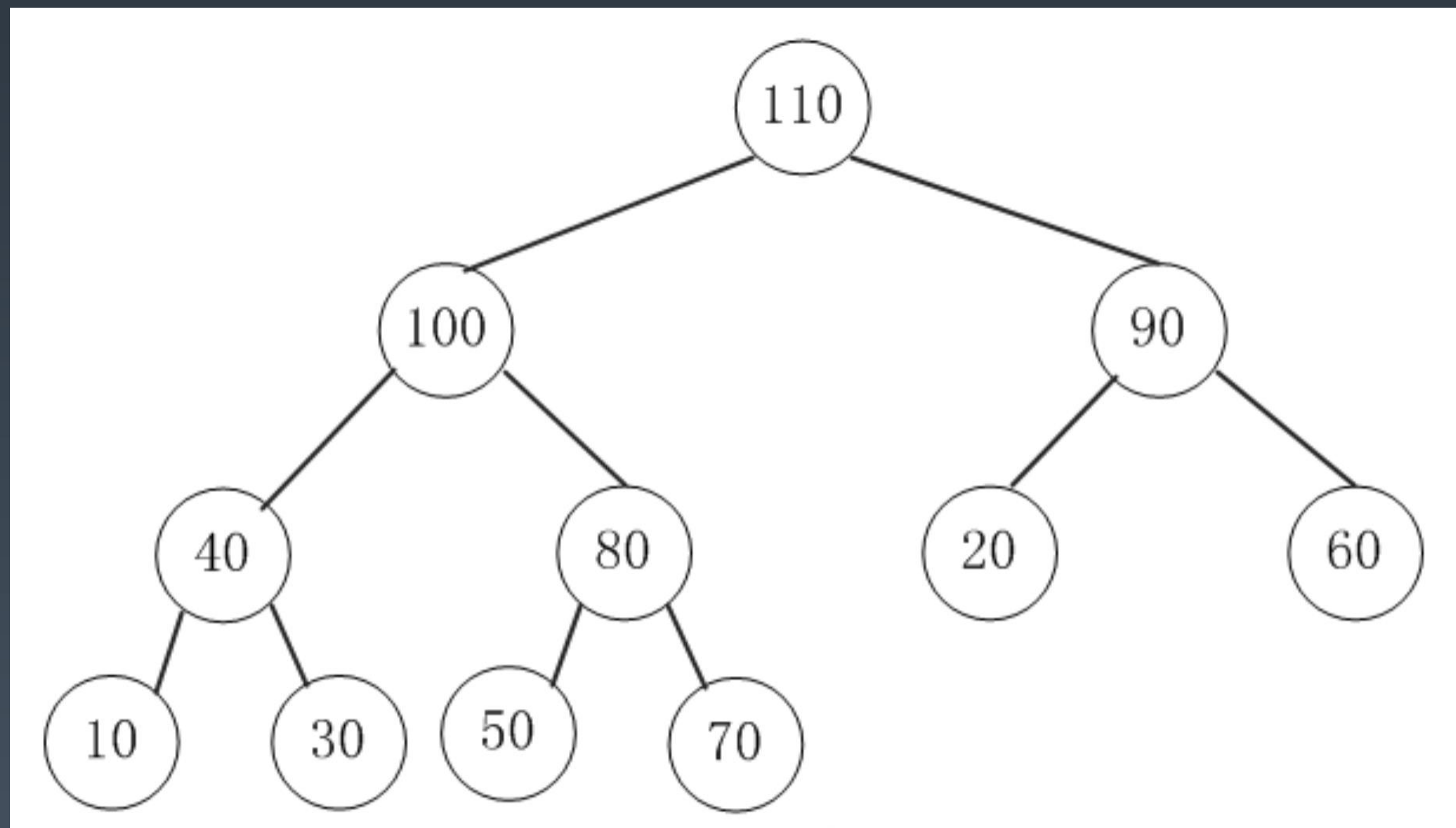
通过完全二叉树来实现（注意：不是二叉搜索树）；

二叉堆（大顶）它满足下列性质：

[性质一] 是一棵完全树。

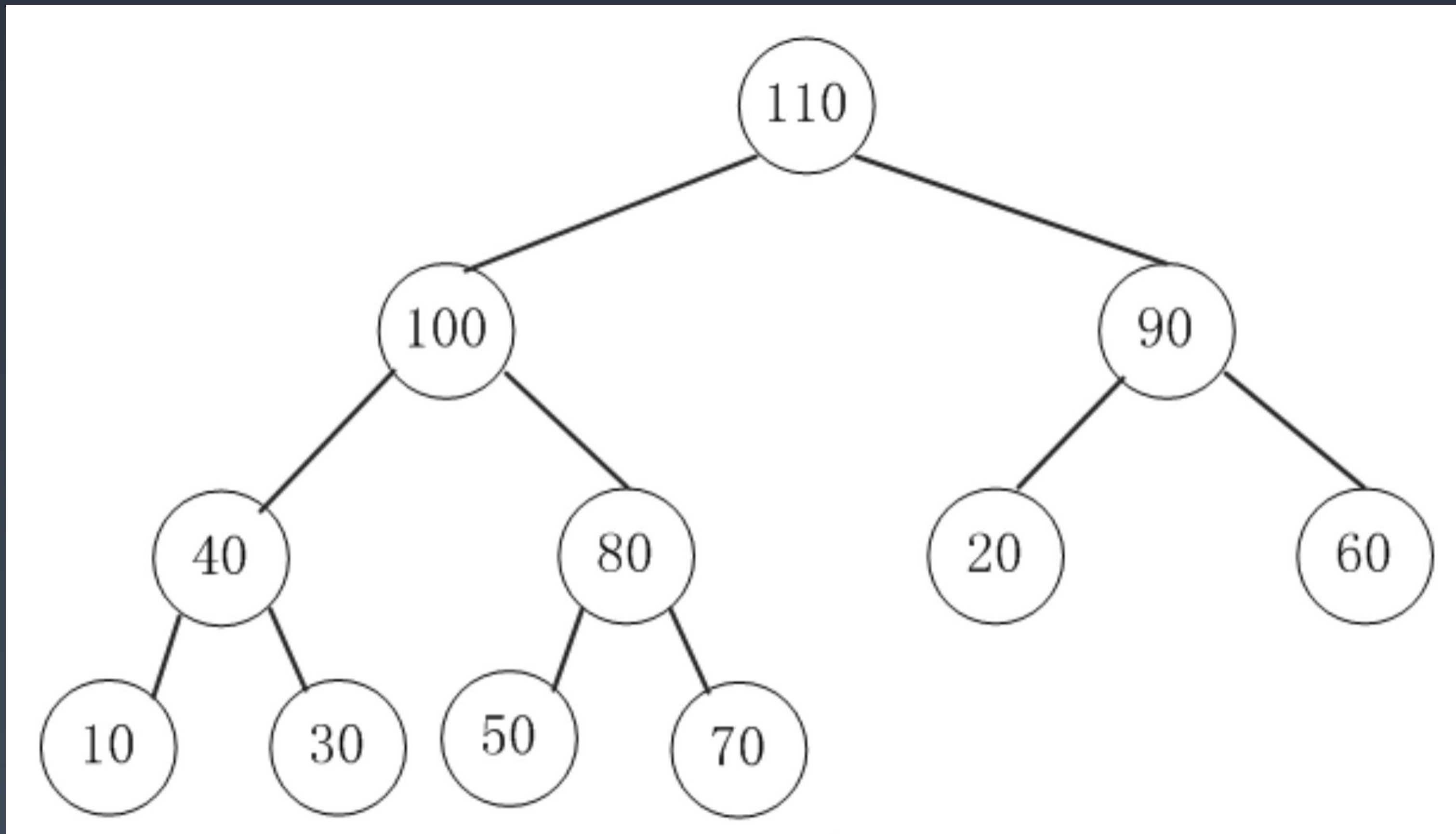
[性质二] 树中任意节点的值总是  $\geq$  其子节点的值；

# 二叉堆



# 二叉堆实现细节

1. 二叉堆一般都通过 “数组” 来实现
2. 假设“第一个元素”在数组中的索引为 0 的话, 则父节点和子节点的位置关系如下:
  - (01) 索引为 $i$ 的左孩子的索引是  $(2*i+1)$ ;
  - (02) 索引为 $i$ 的右孩子的索引是  $(2*i+2)$ ;
  - (03) 索引为 $i$ 的父结点的索引是  $\text{floor}((i-1)/2)$ ;



一维数组: [110, 100, 90, 40, 80, 20, 60, 10, 30, 50, 70]

# 二叉堆

- 0. 根节点（顶堆元素）是：  $a[0]$
- 1. 索引为  $i$  的左孩子的索引是  $(2*i+1)$ ;
- 2. 索引为  $i$  的右孩子的索引是  $(2*i+2)$ ;
- 3. 索引为  $i$  的父结点的索引是  $\text{floor}((i-1)/2)$ ;

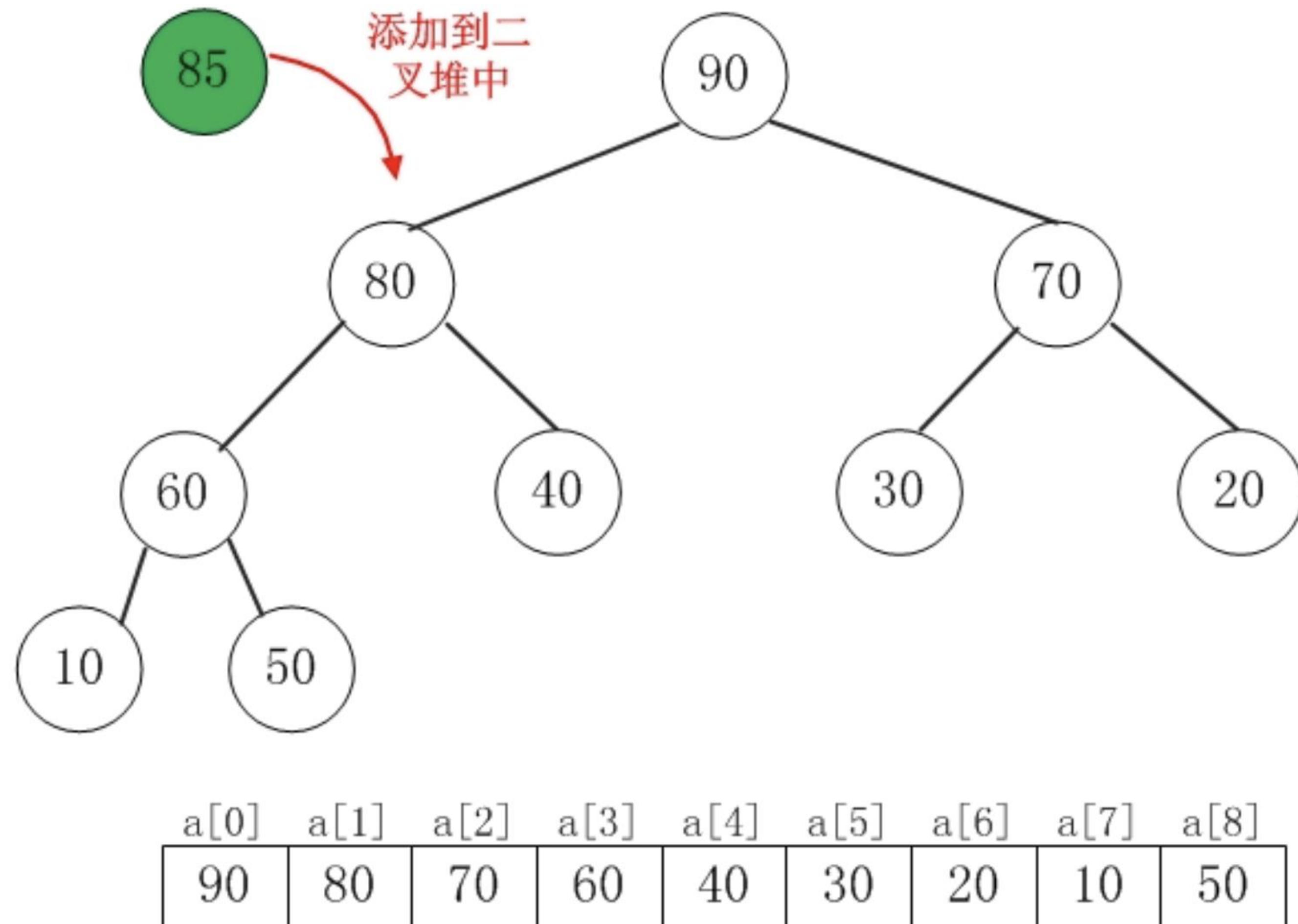


# Insert 插入操作

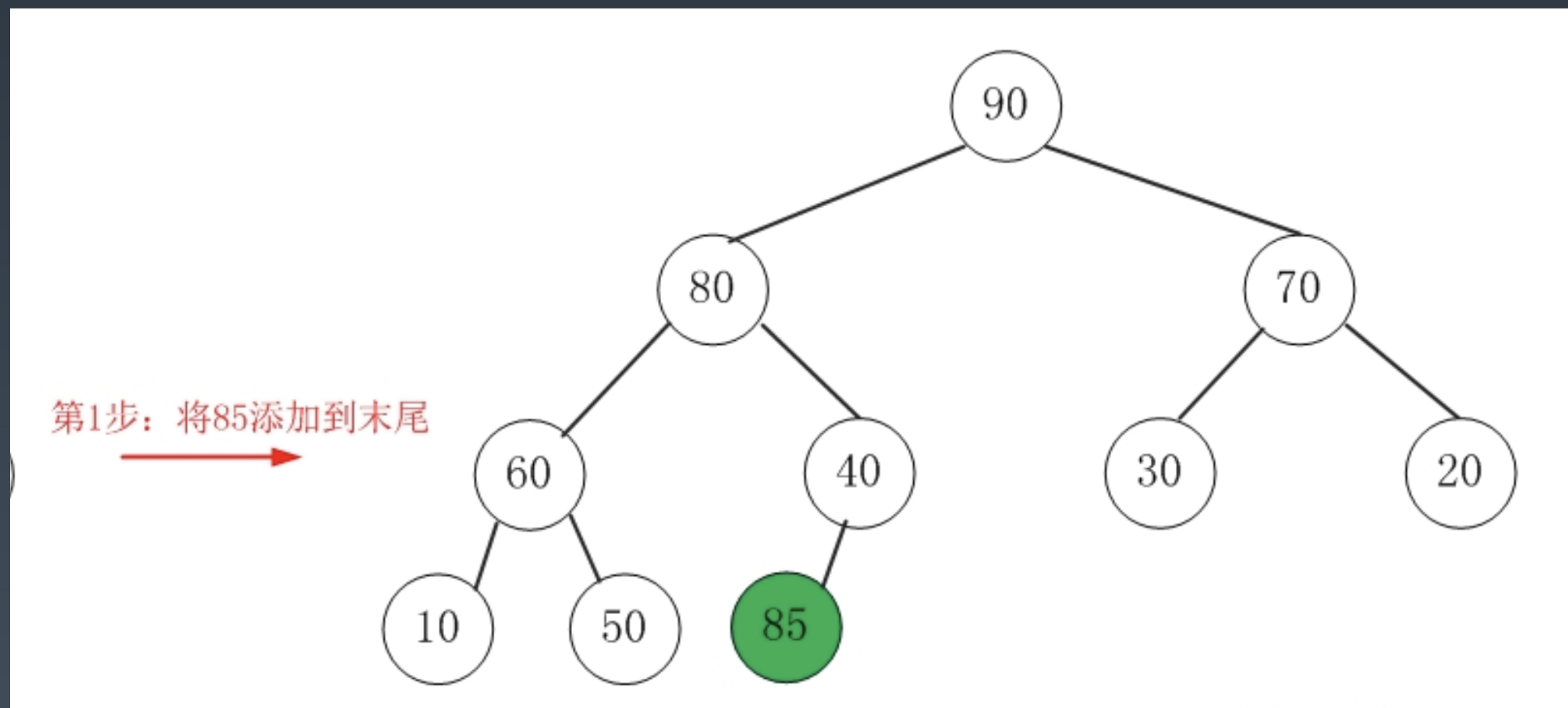
1. 新元素一律先插入到堆的尾部
2. 依次向上调整整个堆的结构（一直到根即可）

HeapifyUp

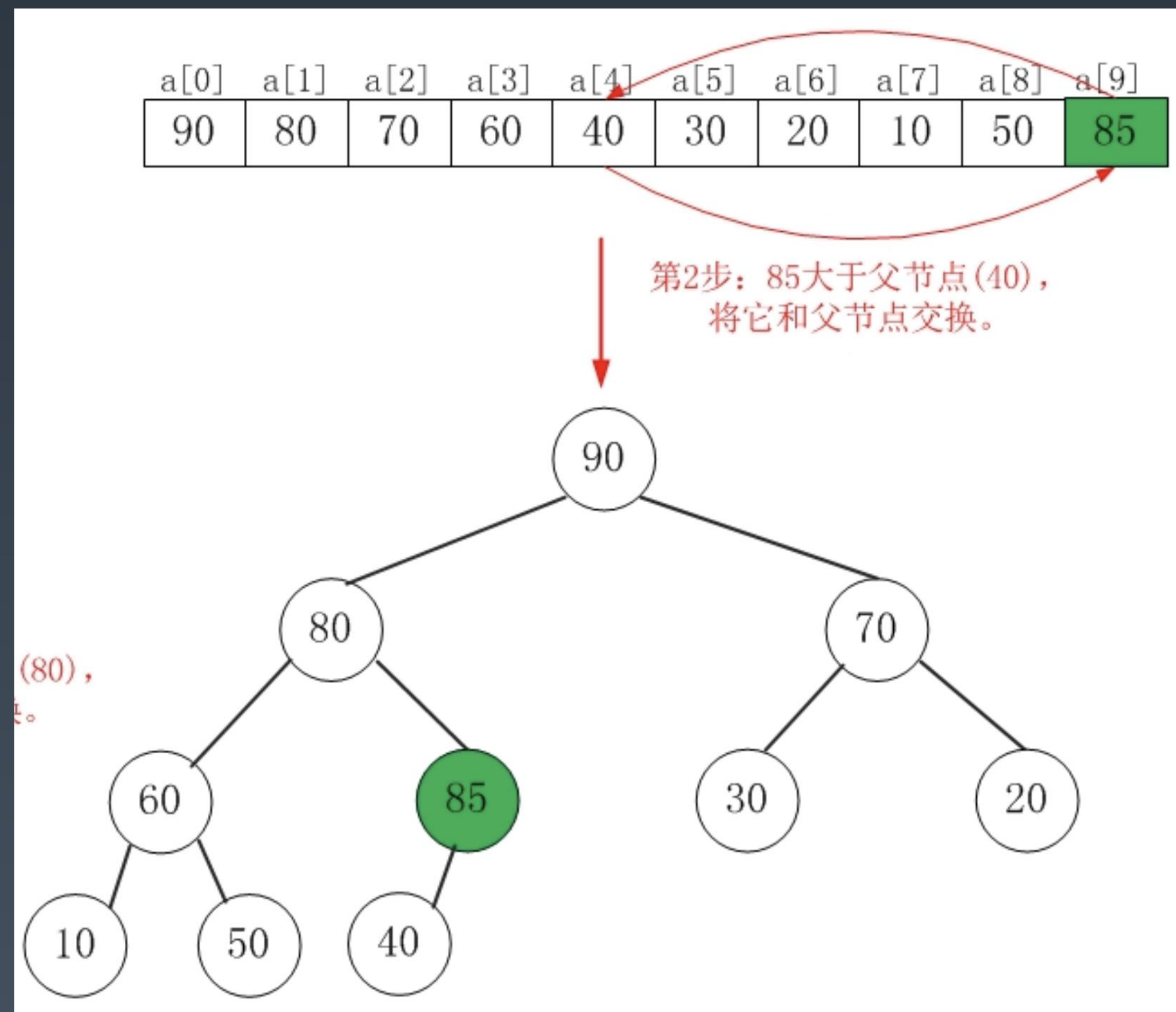
# Insert - $O(\log N)$



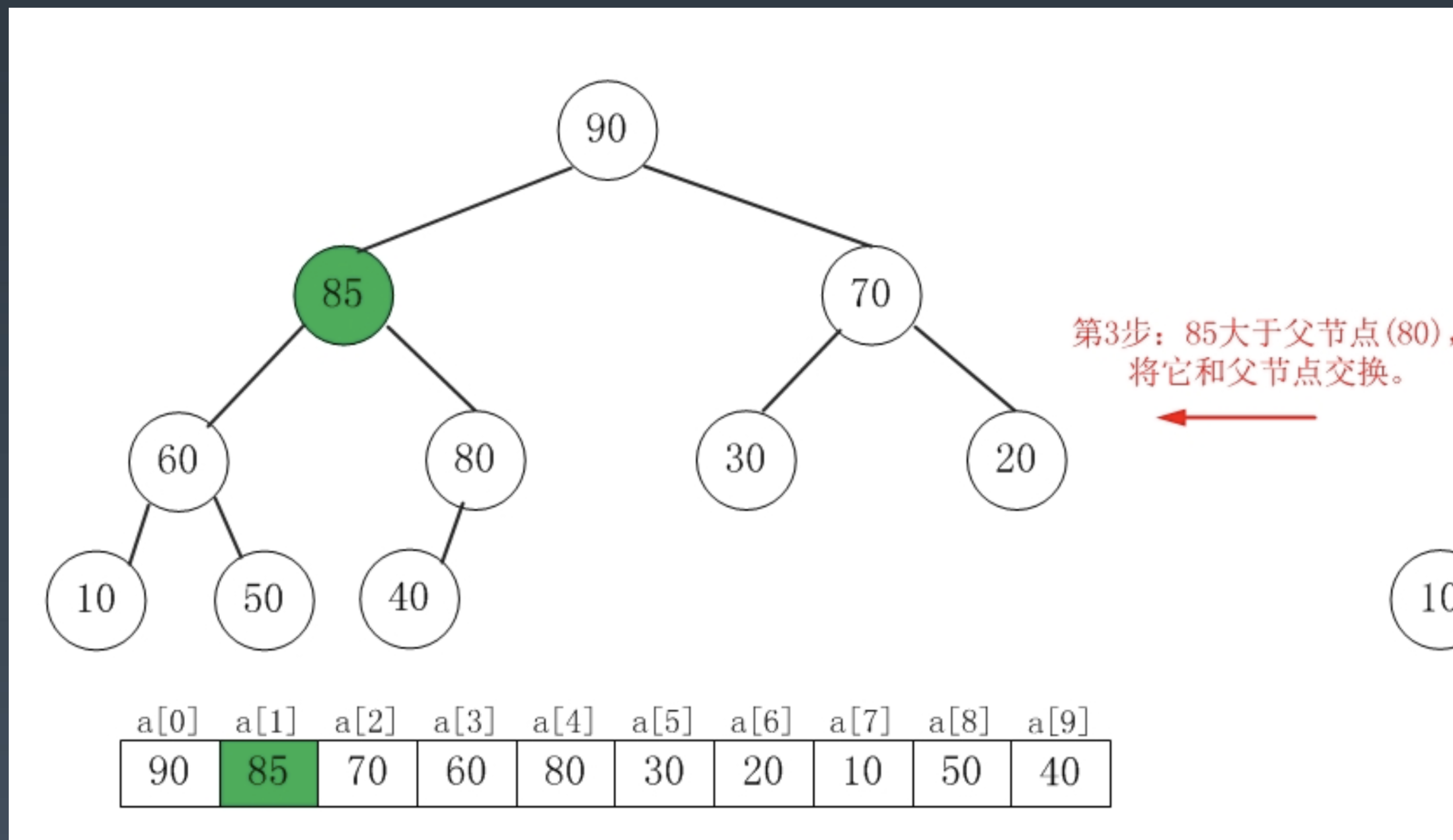
# Insert



# Insert



# Insert - $O(\log N)$



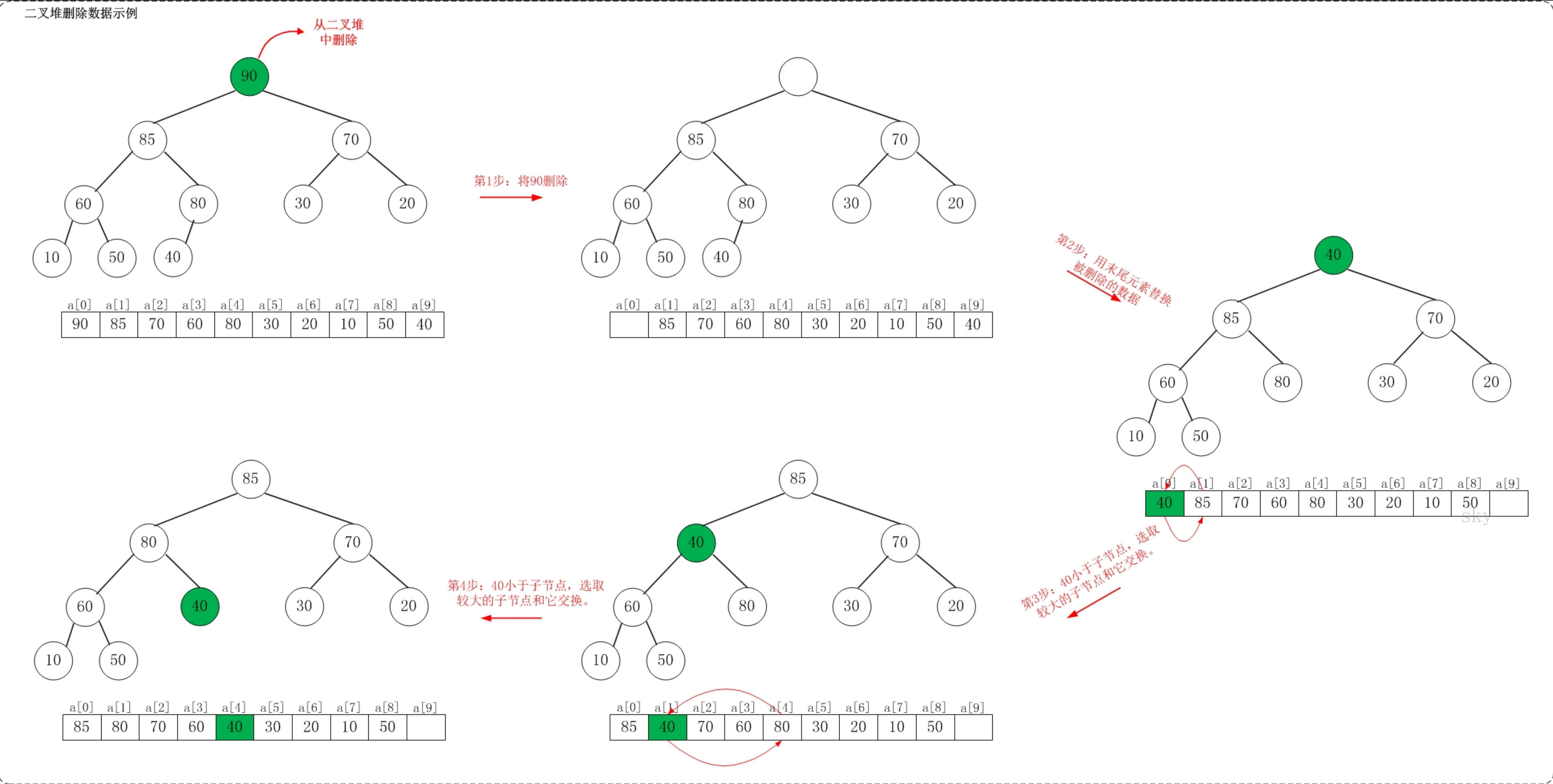
# Delete Max 删除堆顶操作

1. 将堆尾元素替换到顶部（即对顶被替代删除掉）
2. 依次从根部向下调整整个堆的结构（一直到堆尾即可）

HeapifyDown



# Delete Max - $O(\log N)$



注意：二叉堆是堆（优先队列 `priority_queue`）的一种常见且简单的实现；但是并不是最优的实现。

[https://en.wikipedia.org/wiki/Heap\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Heap_(data_structure))



# 实战例题

# 例题

1. <https://leetcode-cn.com/problems/zui-xiao-de-kge-shu-lcof/>
2. <https://leetcode-cn.com/problems/sliding-window-maximum/>
3. <https://leetcode-cn.com/problems/top-k-frequent-elements/>

# Homework

1. HeapSort : 自学 <https://www.geeksforgeeks.org/heap-sort/>
2. <https://leetcode-cn.com/problems/chou-shu-lcof/>
3. <https://leetcode-cn.com/problems/top-k-frequent-elements/>