Heaps

By: Dhruv Ahuja and Janani Sriram

Berkeley
UNIVERSITY OF CALIFORNIA

Section I
Heap Representations

Berkeley
UNIVERSITY OF CALIFORNIA

1. What is a heap?

- A Binary tree with added constraints
- When writing methods with heaps, we seek to always retain these heap properties

Property 1: Parents > Children

- Min heap: each parent is lesser than both of its children
    - The smallest element is stored at the root
    - Sibling Nodes have no relation
        - Unlike BSTs, the left node does not have to be less than the right node
- Max heap: similar properties, but with the largest element stored at the root

Property 2: Completeness

- During insertion, we must insert into the bottom of the heap
  - Ensures a more balanced heap

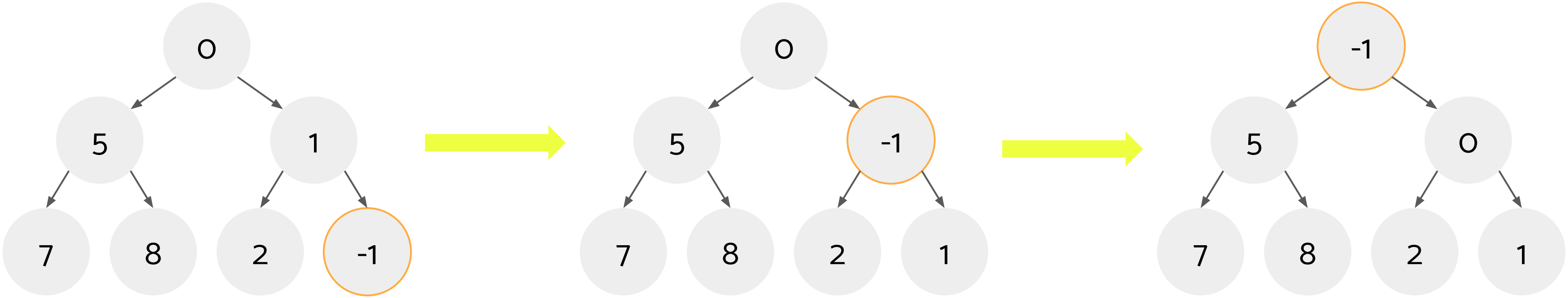
Representing Heaps

- Store the root at index 1
- If node is located at position n
    - Left Child: 2n
    - Right Child: 2n + 1

Section II
Heap Methods

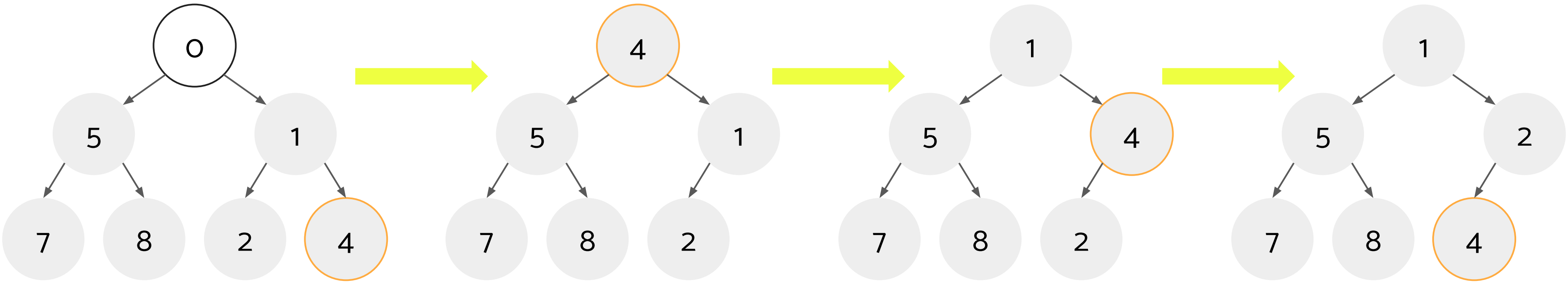
Insertion into (Min-)Heaps

We insert elements into the next available spot in the heap and **bubble up** as necessary: if a node is smaller than its parent, they will swap. (Check: what changes if this is a max heap?)



RemoveMin from (Min-)Heaps

We swap the last element with the root, remove the last element, and **bubble down** as necessary: if a node is greater than its children, it will swap with the lesser of its children. (Check: what changes if this is a max heap?)



Find

- Heaps aren’t BSTs, so searching generally takes O(N) time

Heap Asymptotics

Operation	Runtime
insert	Θ(logN)
findMin	Θ(1)
removeMin	Θ(logN)

Thank you!
Good luck :)

Berkeley
UNIVERSITY OF CALIFORNIA

Definition credit
Paul Hilfinger