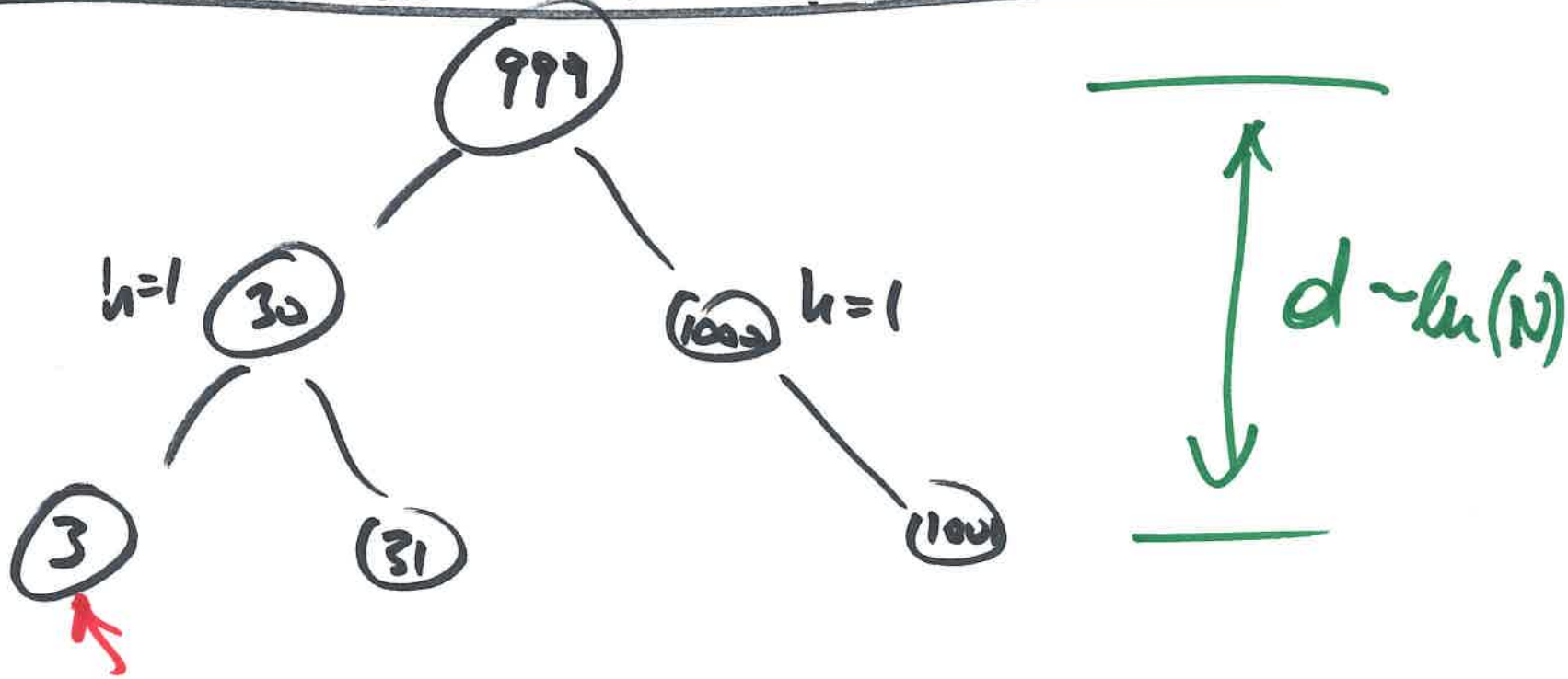
CS 2341

Chapter 6

Heaps

Balanced Binary Search Tree

Find Minimum



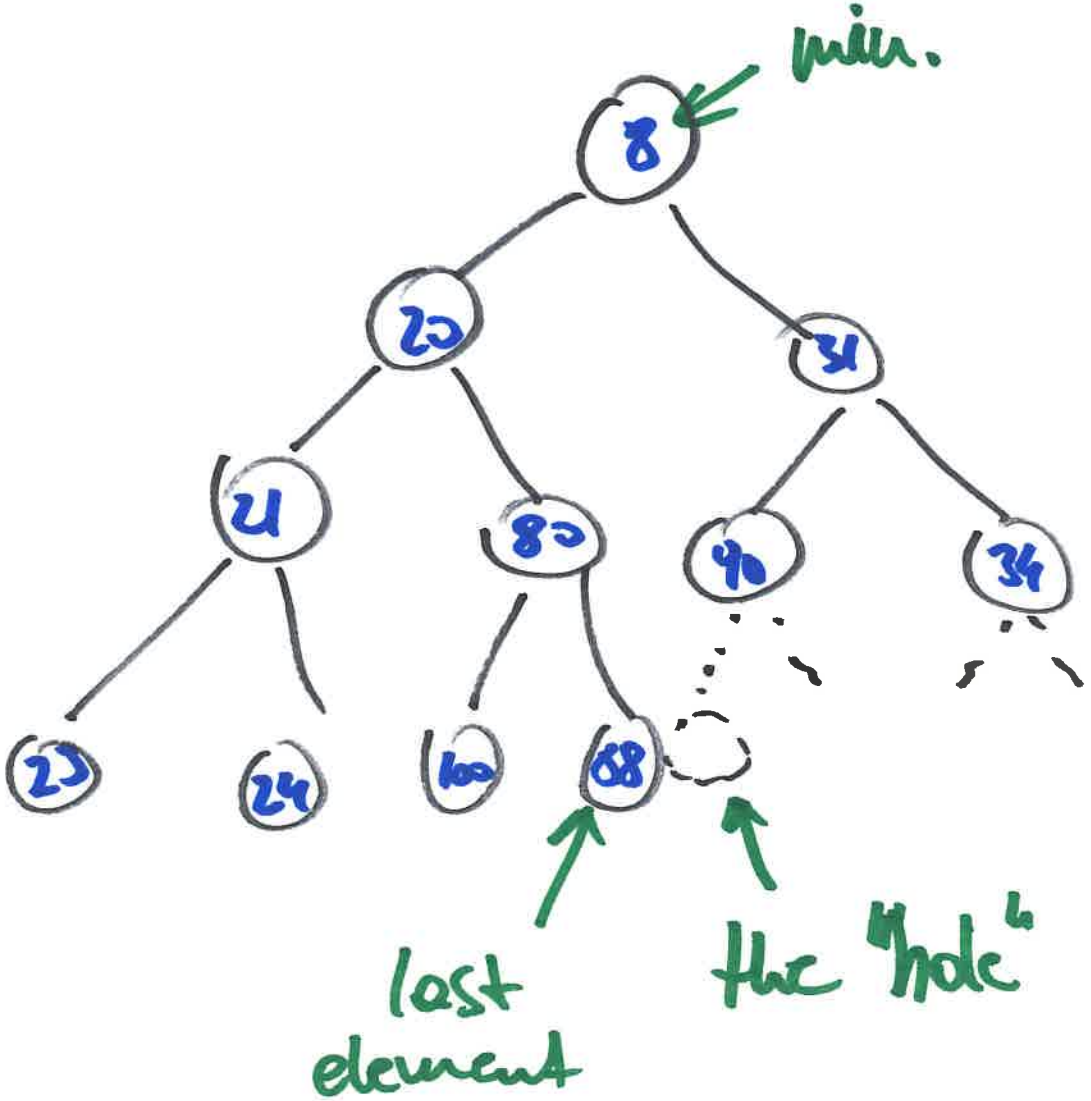
min

find min:

- $O(d) = O(\ln N)$

needs balancing operations which are expensive.

Heap



1. structure property

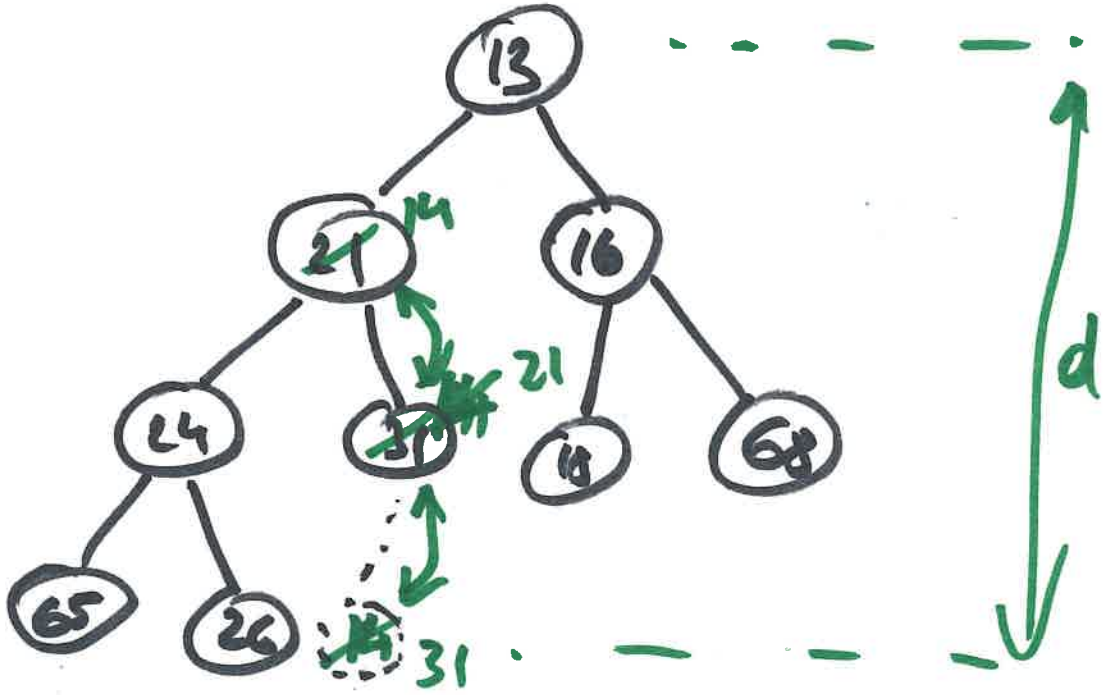
⇒ balanced binary tree

fill level-wise

2. heap-order property

parent key is always smaller than all its children's keys

Heap: Insertion

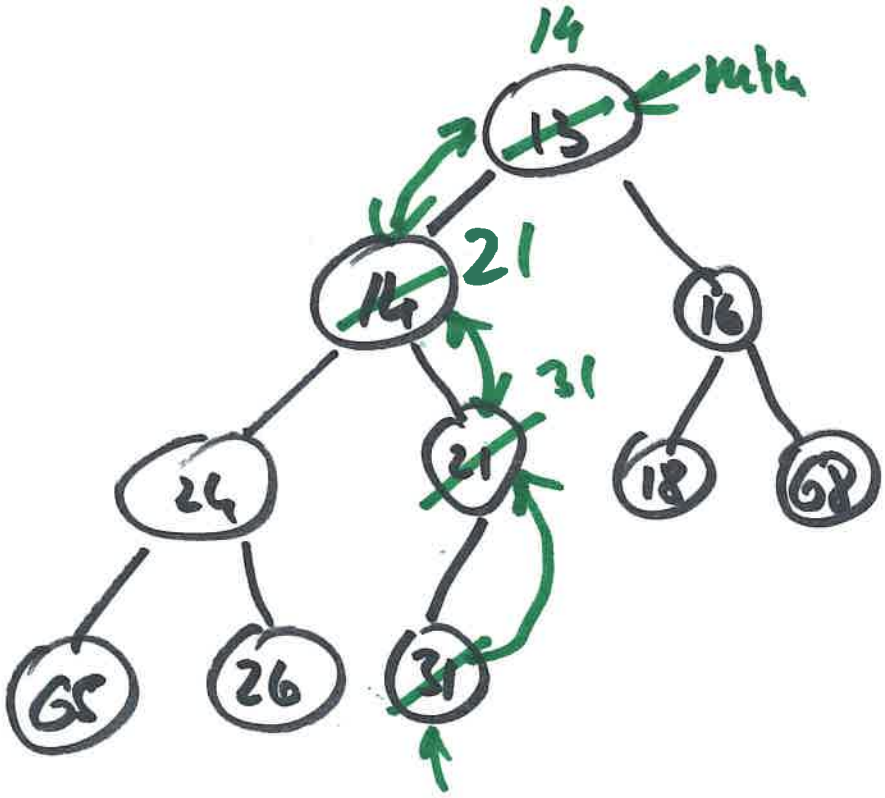


insert (14)

1. insert into "hole"
2. "percolate up" till it fits

$$O(d) = O(\log N)$$

Heap : Delete Min



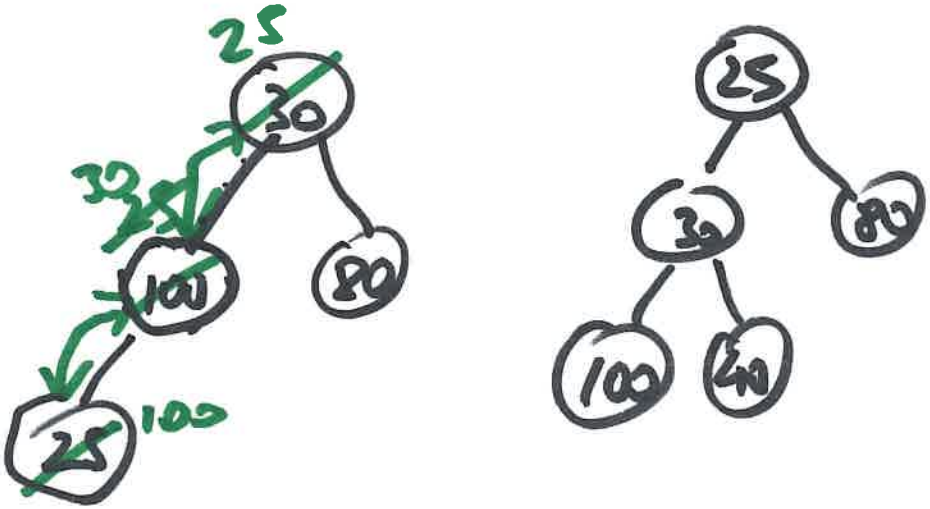
$O(\log N)$

delete Min() → 13

1. root becomes new "hole"
2. percolate down the hole. to the <u>smallest</u> child till the last element fits in the hole.

Heap Example

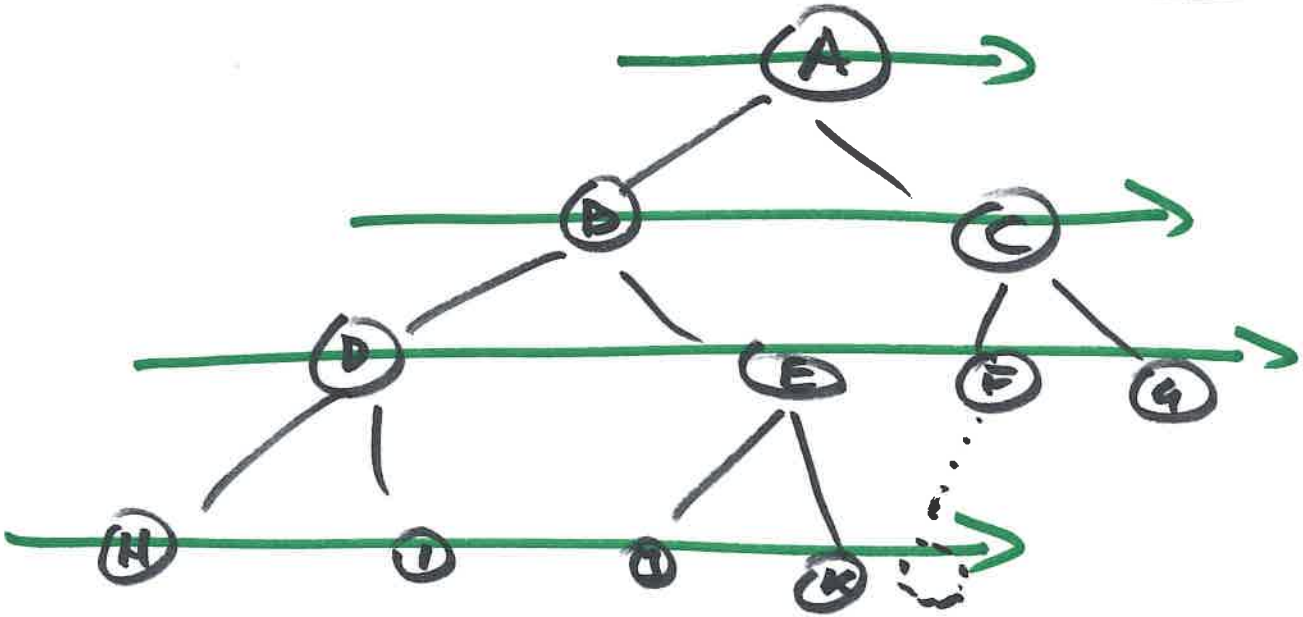
insert : 30, 100, 80, 25, 40



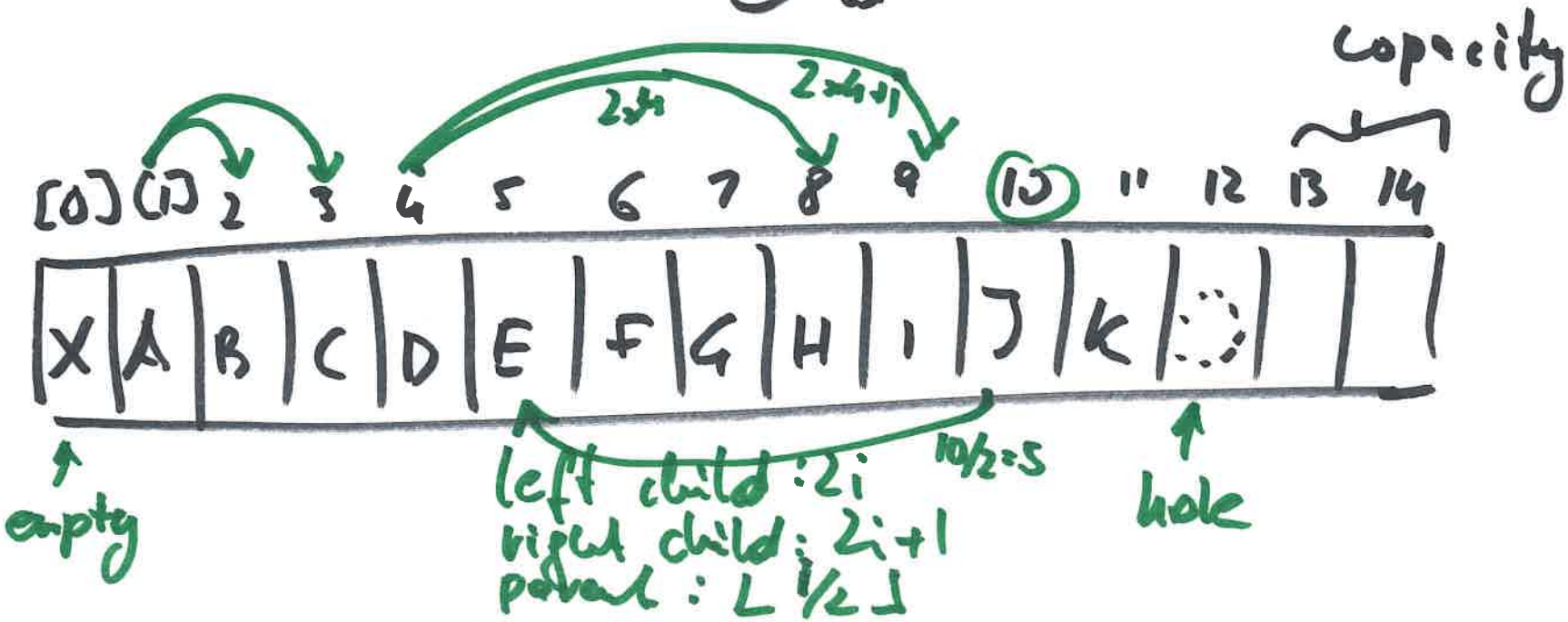
deleteMin() till heap is empty. → 25, 30, 40, 80, 100

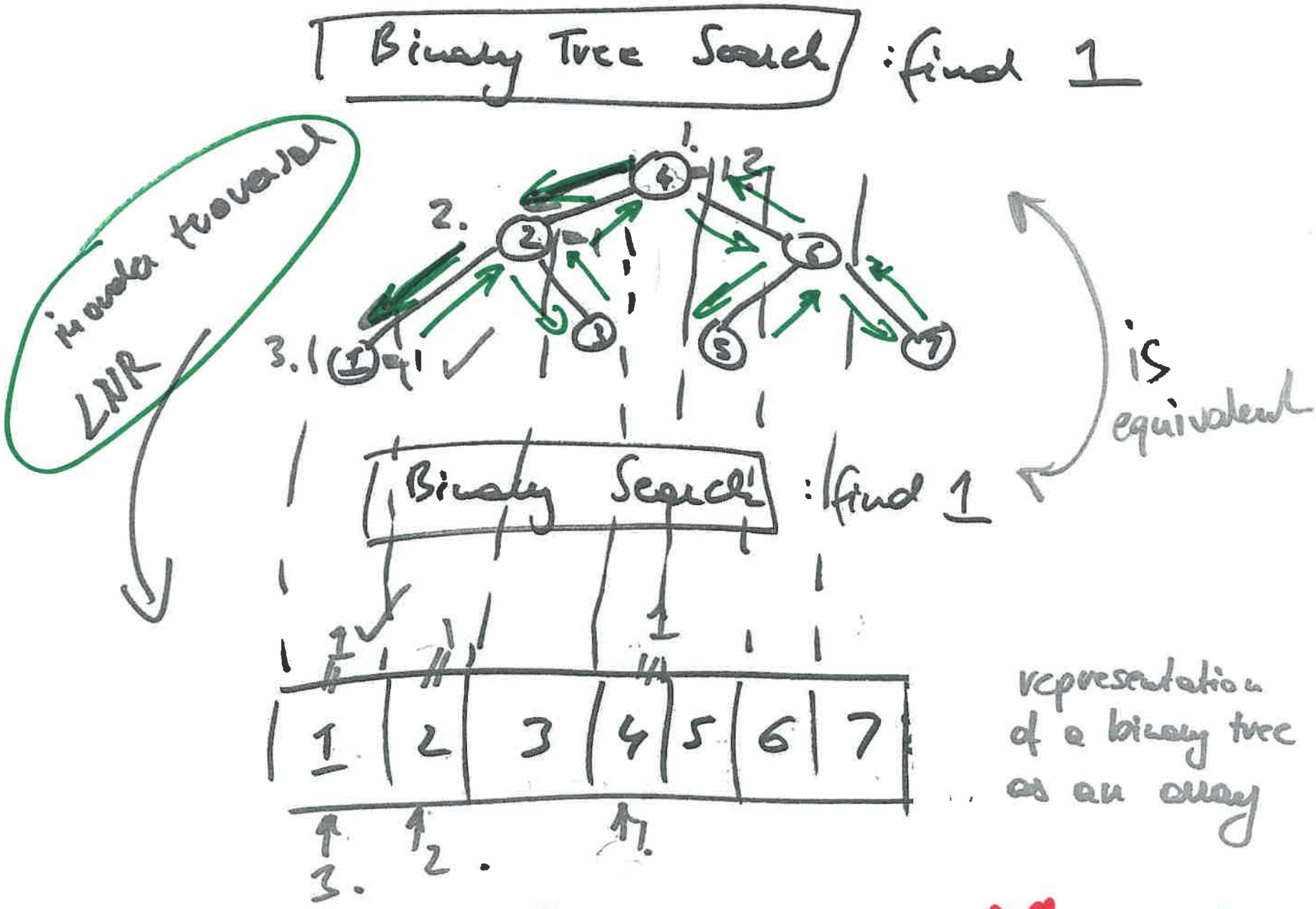
Heap represented as an Array

Note: heaps are never stored as a tree!



fill array level-wise





The array representation is very different from the one for heaps!