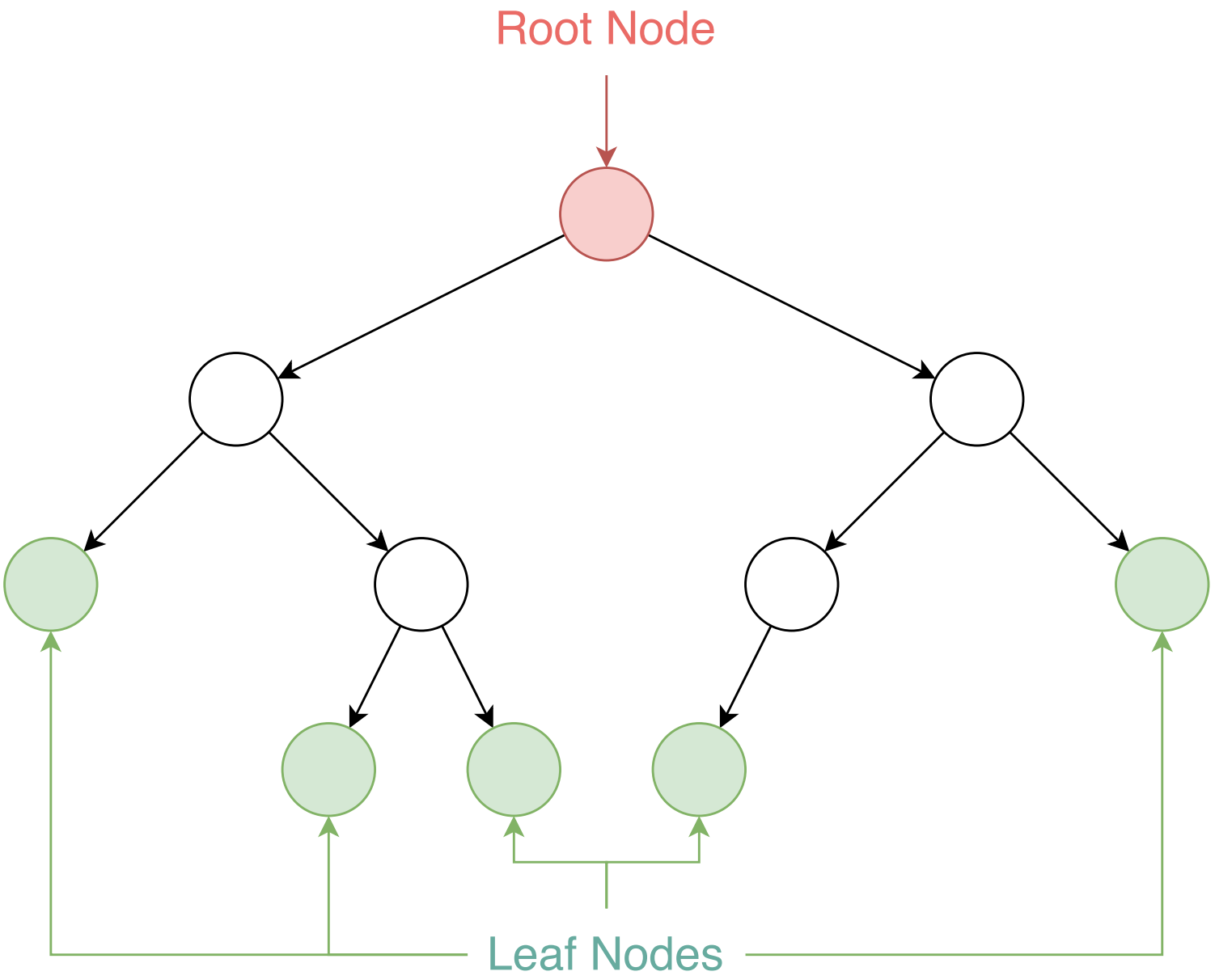
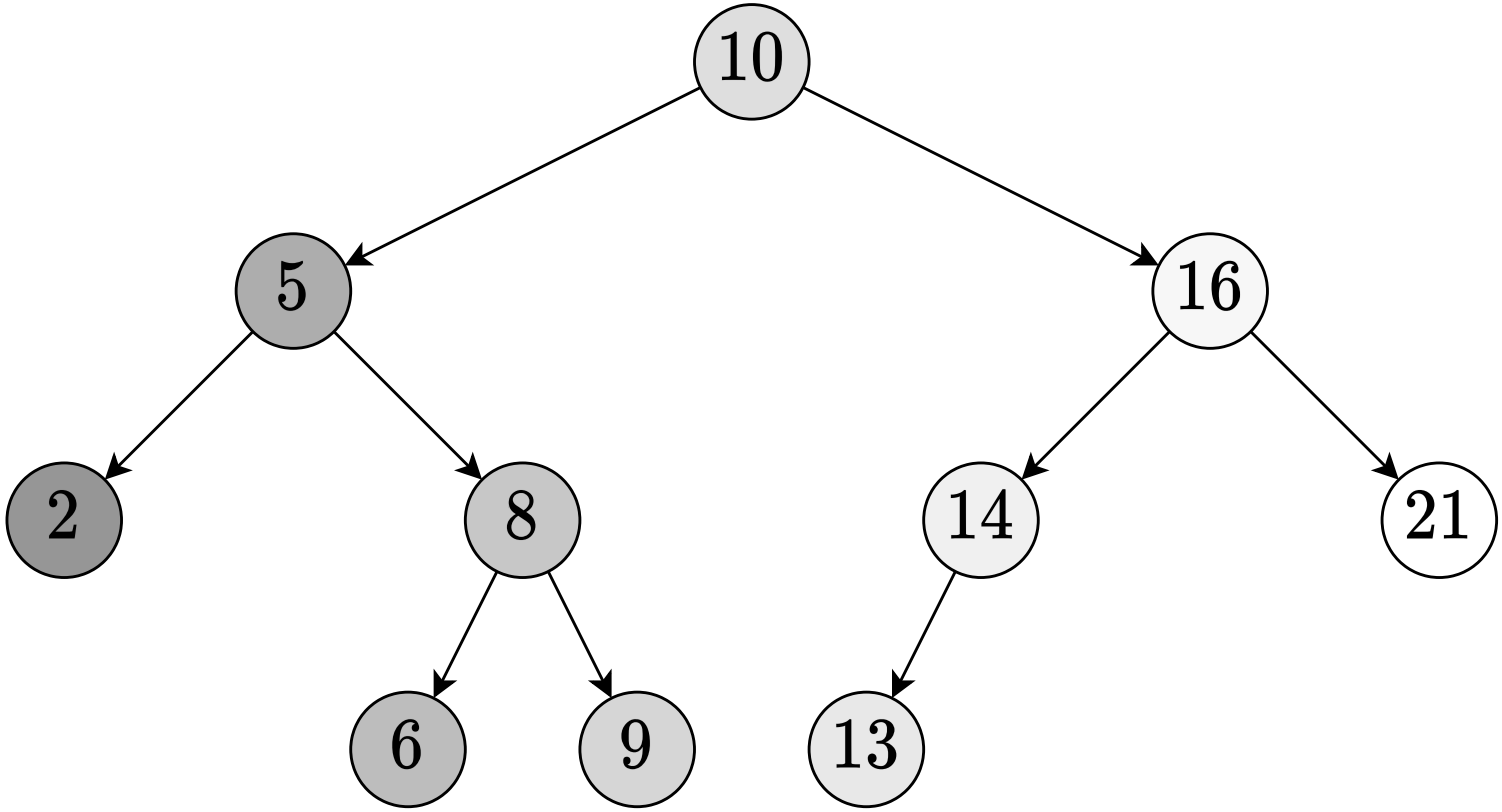
Tutorial 10 - Binary Trees

- A **binary tree** is a linked data structure where each element has a link to at most two **child nodes**
- There is a **root node** that is not the child of any other node
- There are **leaf nodes** which have no children


Root Node
Leaf Nodes

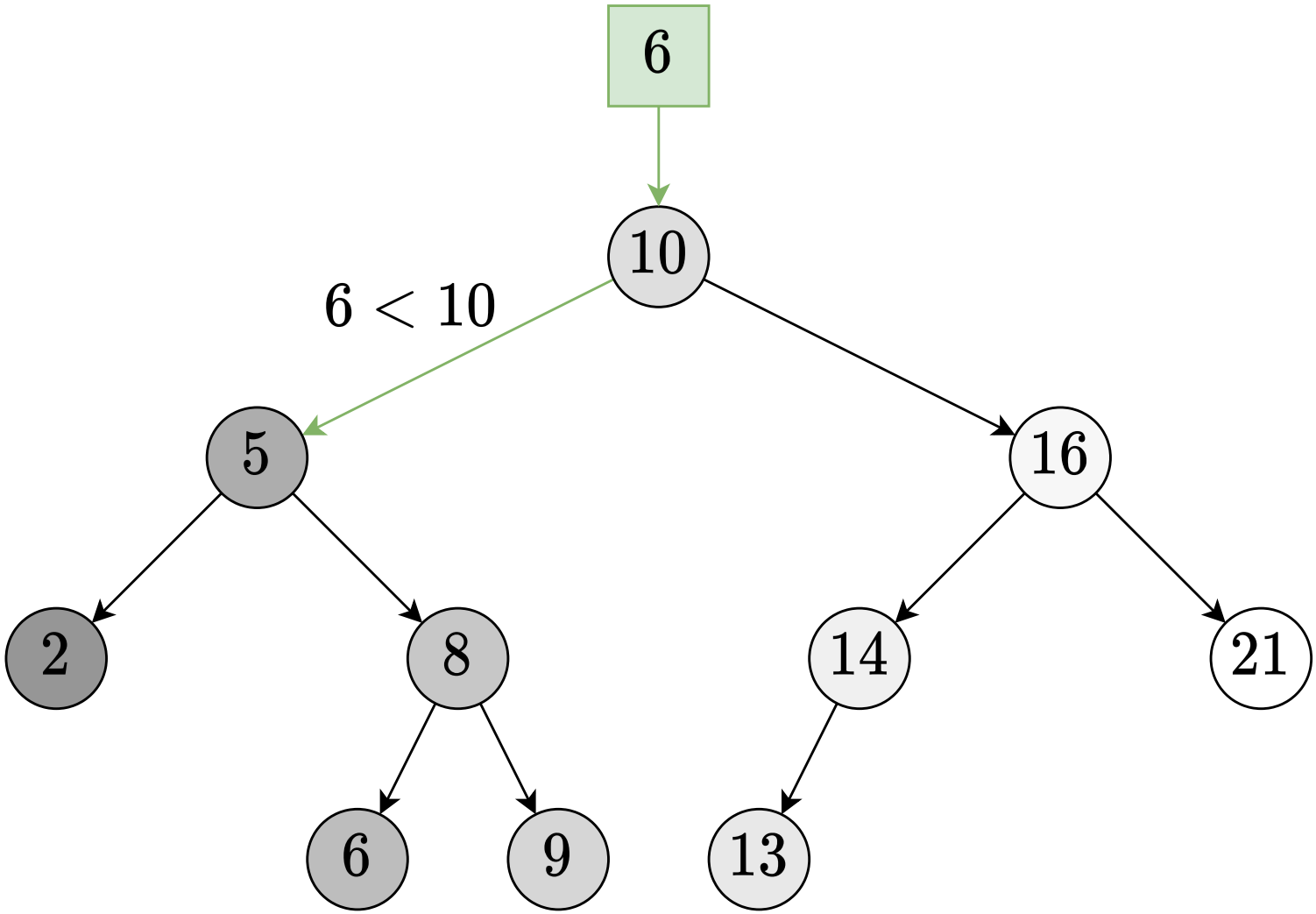
Binary Search Trees

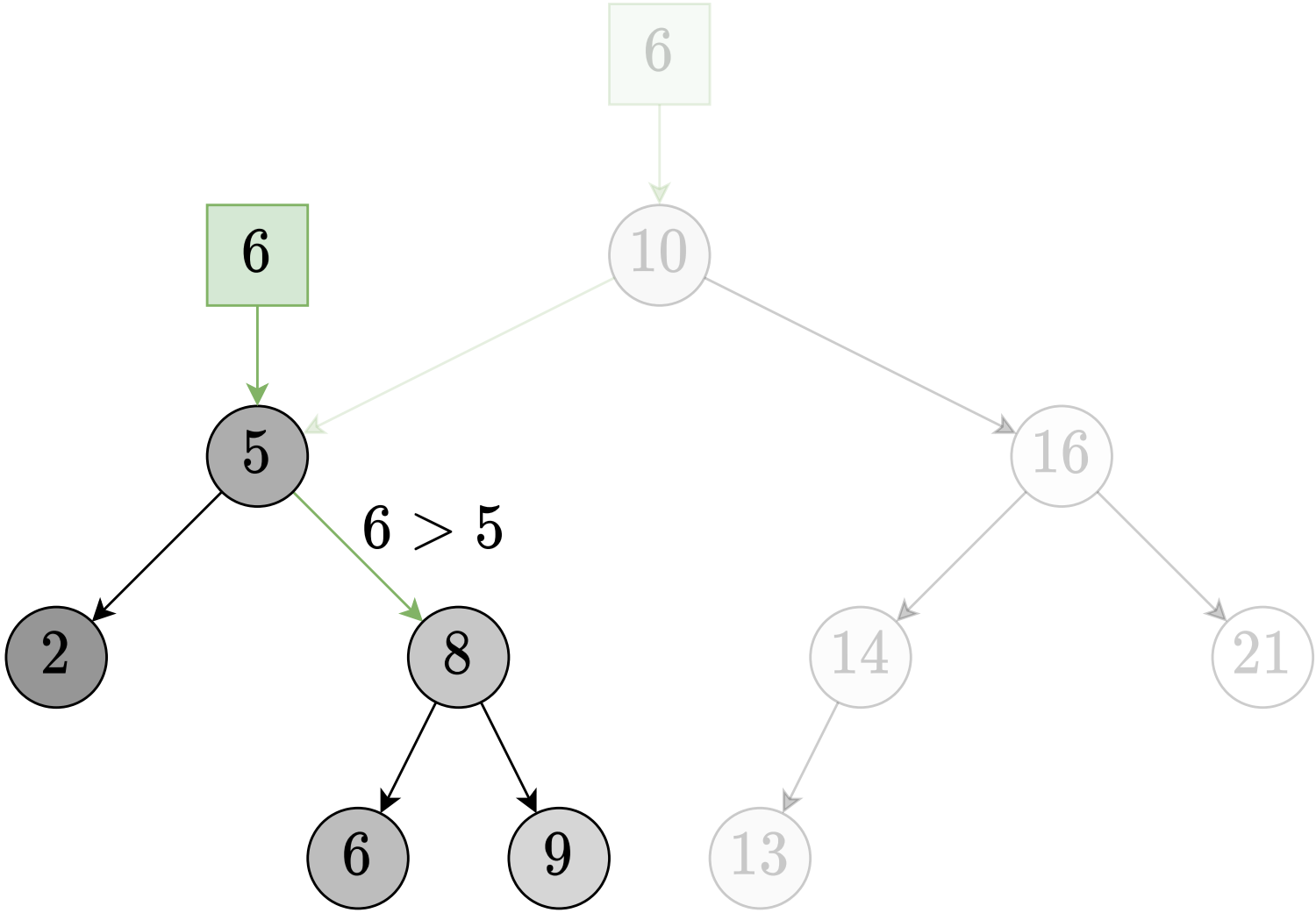
- A **binary search tree** is a binary tree where each element is associated with a **value**
- The left child and all of its **descendants** have values that are **lower** than it's parent's value
- The right child and all of its descendants have values that are **higher** than it's parent's value

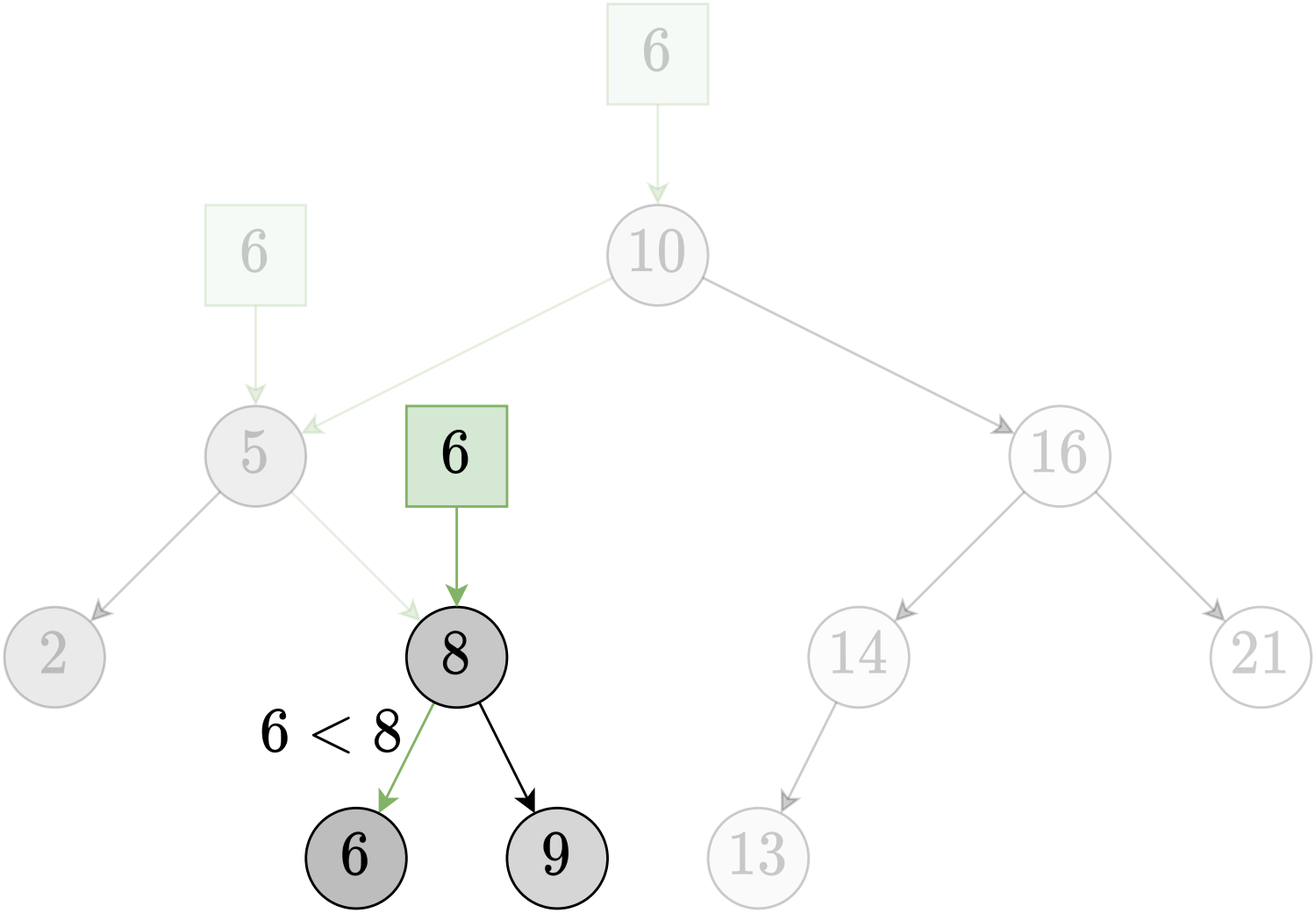

10
5
16
2
8
14
21
6
9
13

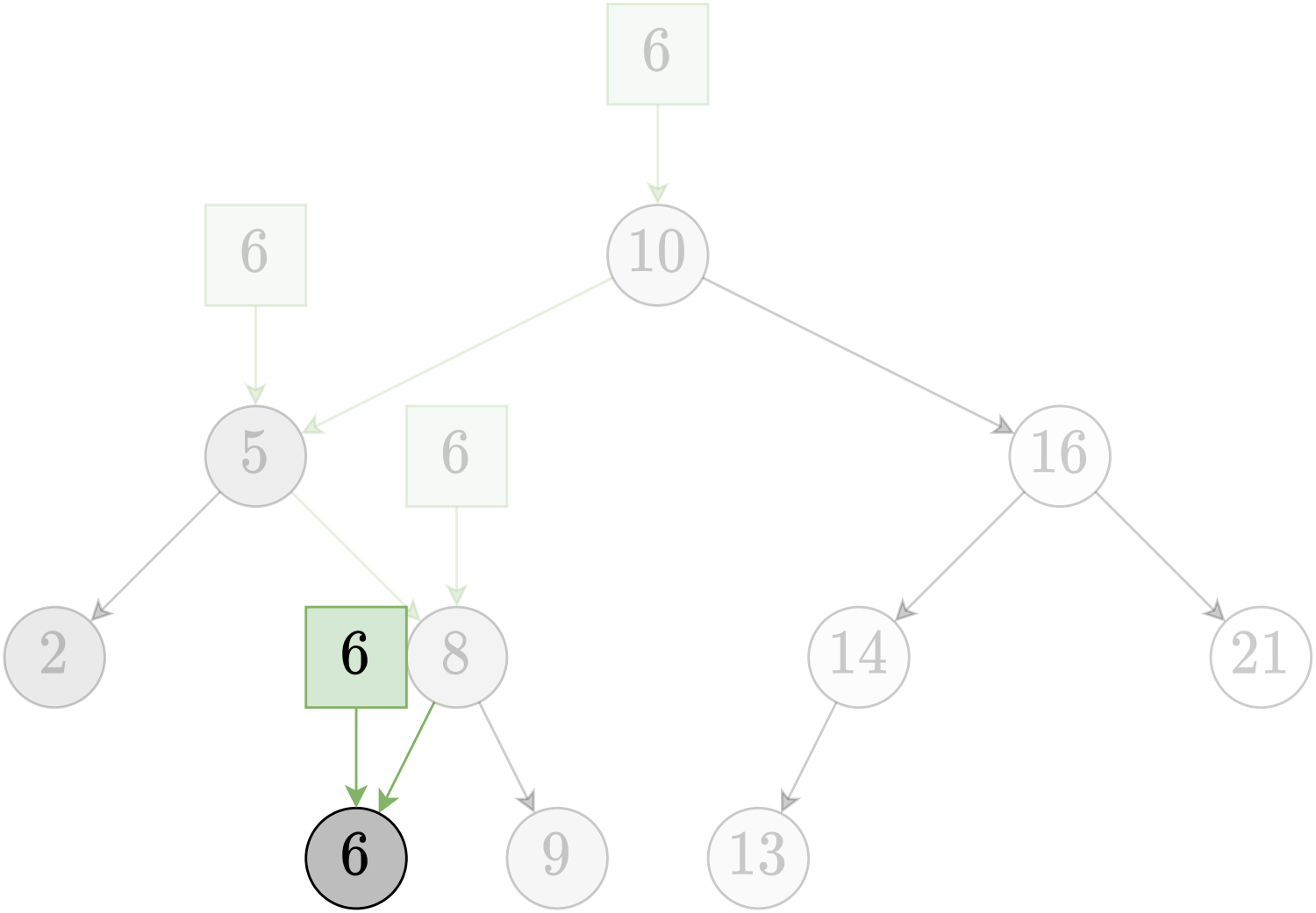
Binary Search Trees

- Binary search trees represent collections of data that can be ordered
- We can check whether or not a value is stored in the binary search tree using **binary search**


6
10
$6 < 10$
5
16
2
8
14
21
6
9
13


6
6
10
5
16
$6 > 5$
2
8
14
21
6
9
13


6
6
10
5
6
16
2
8
14
21
$6 < 8$
6
9
13


6
6
10
5
6
16
2
6
8
14
21
6
9
13

Exercise: Implement a Binary Search Tree

1. Implement a binary search tree in C.
2. Provide a method called `push` for adding new nodes to a tree.
3. Provide a method called `find` that checks whether some `value` is contained in the binary tree. Use binary search.
4. Provide a method called `print` that prints a binary search tree.