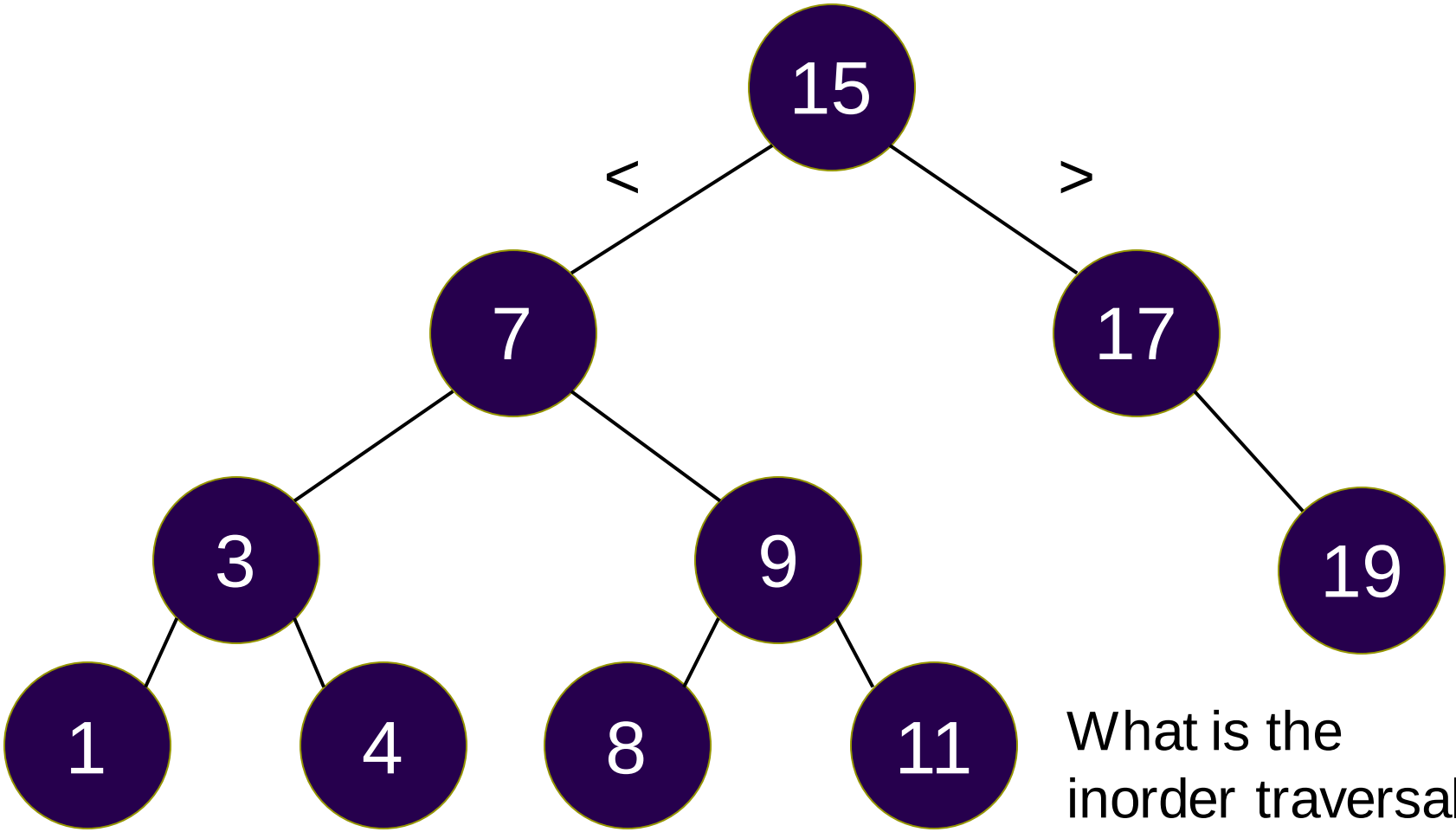
Binary Search Tree (BST)



What is the inorder traversal of a BST?

BST Definition

- BST is a binary tree
- Each node stores a value from an ordered domain, e.g., numbers or strings
- The value of any node is $\geq$ all values in the left subtree, and $\leq$ all values in the right subtree

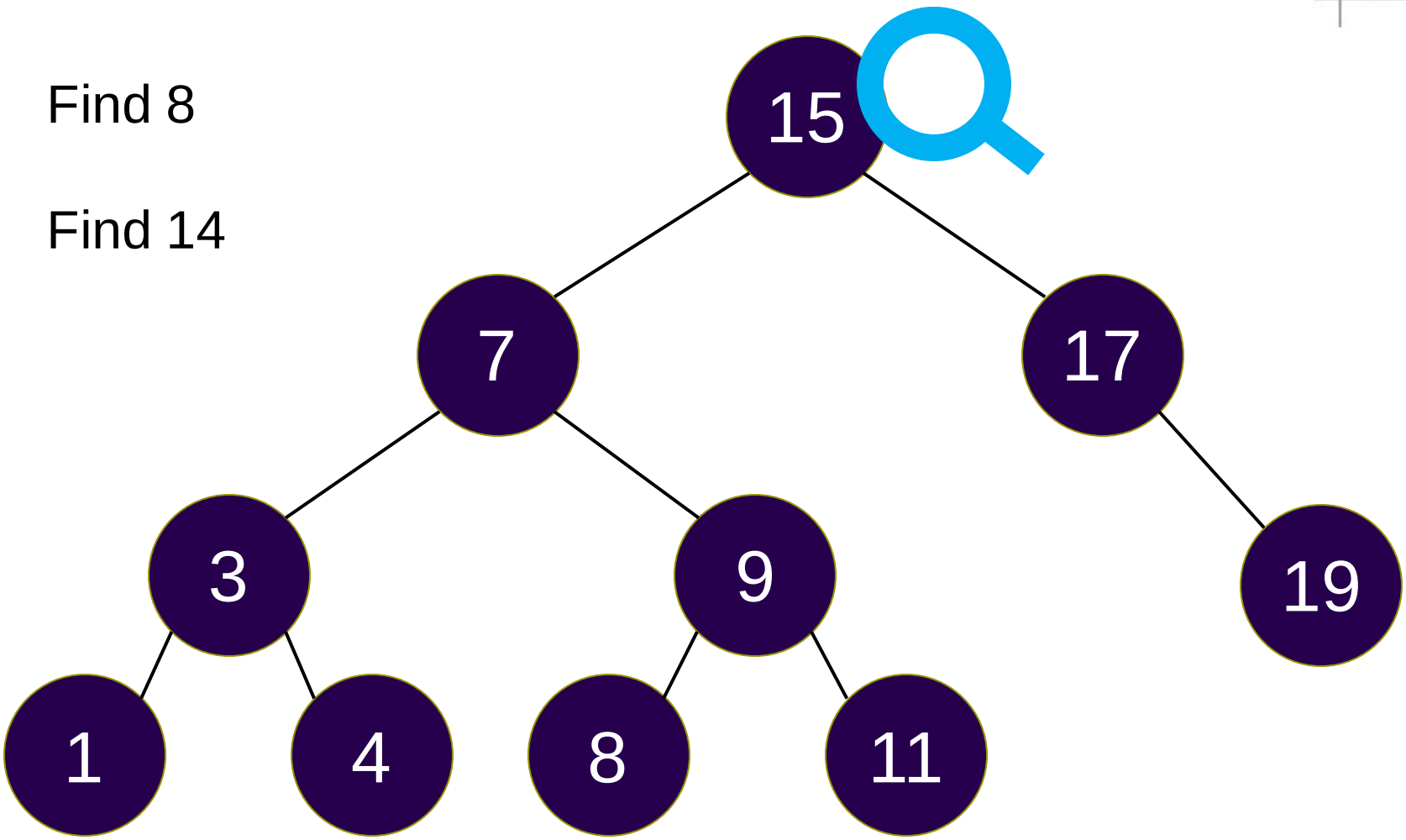
BST ADT

- Initialize() – Initializes an empty BST
- Find(x) – Searches for x in the tree and returns true or false
- Insert(x) – Inserts the value x into the BST
- Delete(x) – Deletes the value x from the tree if it is in the tree
- FindMin() – Returns the minimum value in the tree without deleting it
- FindMax() – Returns the maximum value in the tree without deleting it

BST.Find

Find 8

Find 14



BST.Find

```
bool find(Node* node, T x) {
  if (node == null)
     return false;
  if (x == node.value)
    return true;
  if (x < node.value)
    find(node._____, x);
  if (x > node.value)
    find(node._____, x);
}
```

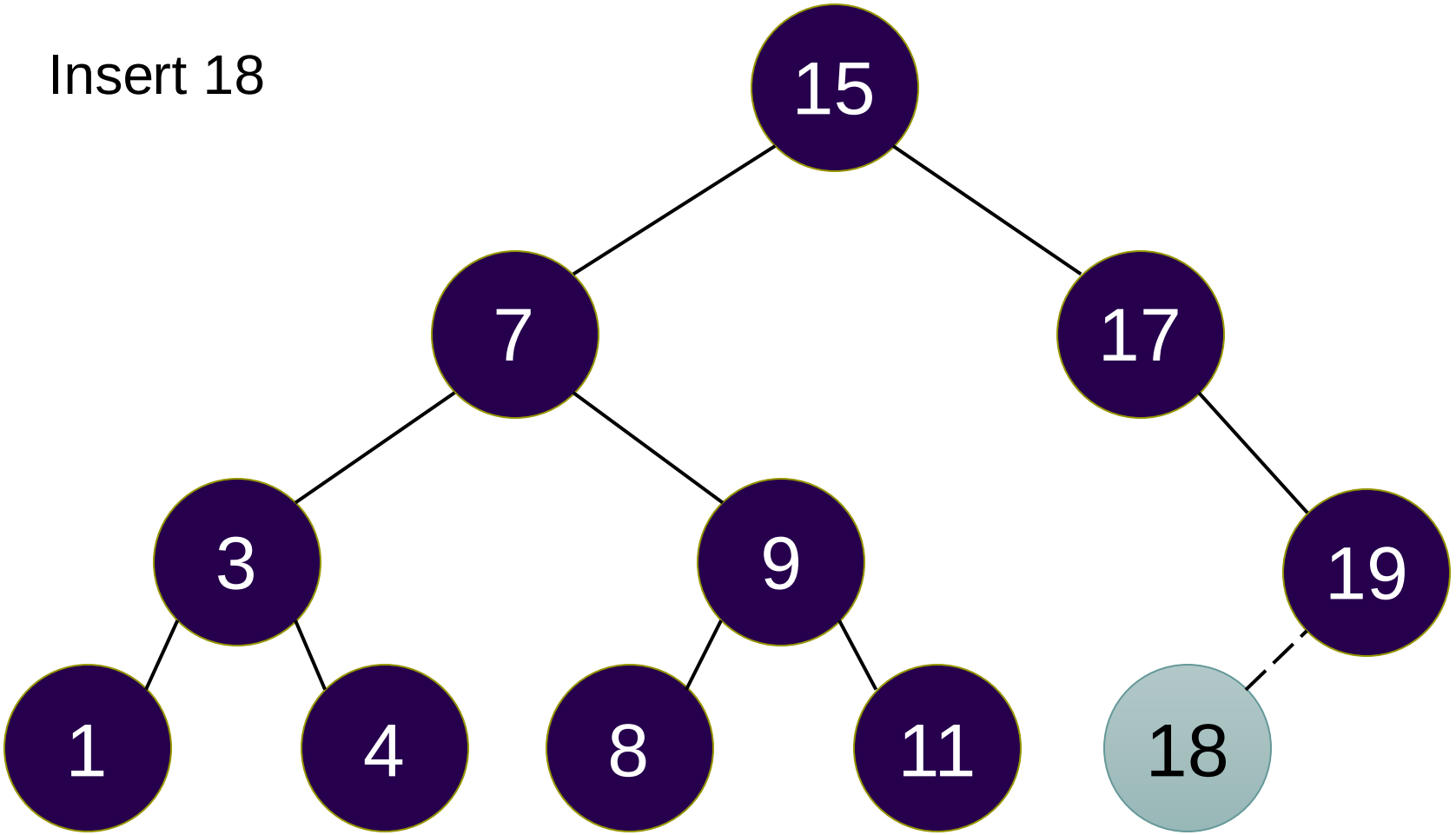
BST.Find

```
bool find(Node* node, T x) {
  if (node == null)
     return false;
  if (x == node.value)
    return true;
  if (x < node.value)
    find(node.left, x);
  if (x > node.value)
    find(node.right, x);
}
```

FindMin()
FindMax()

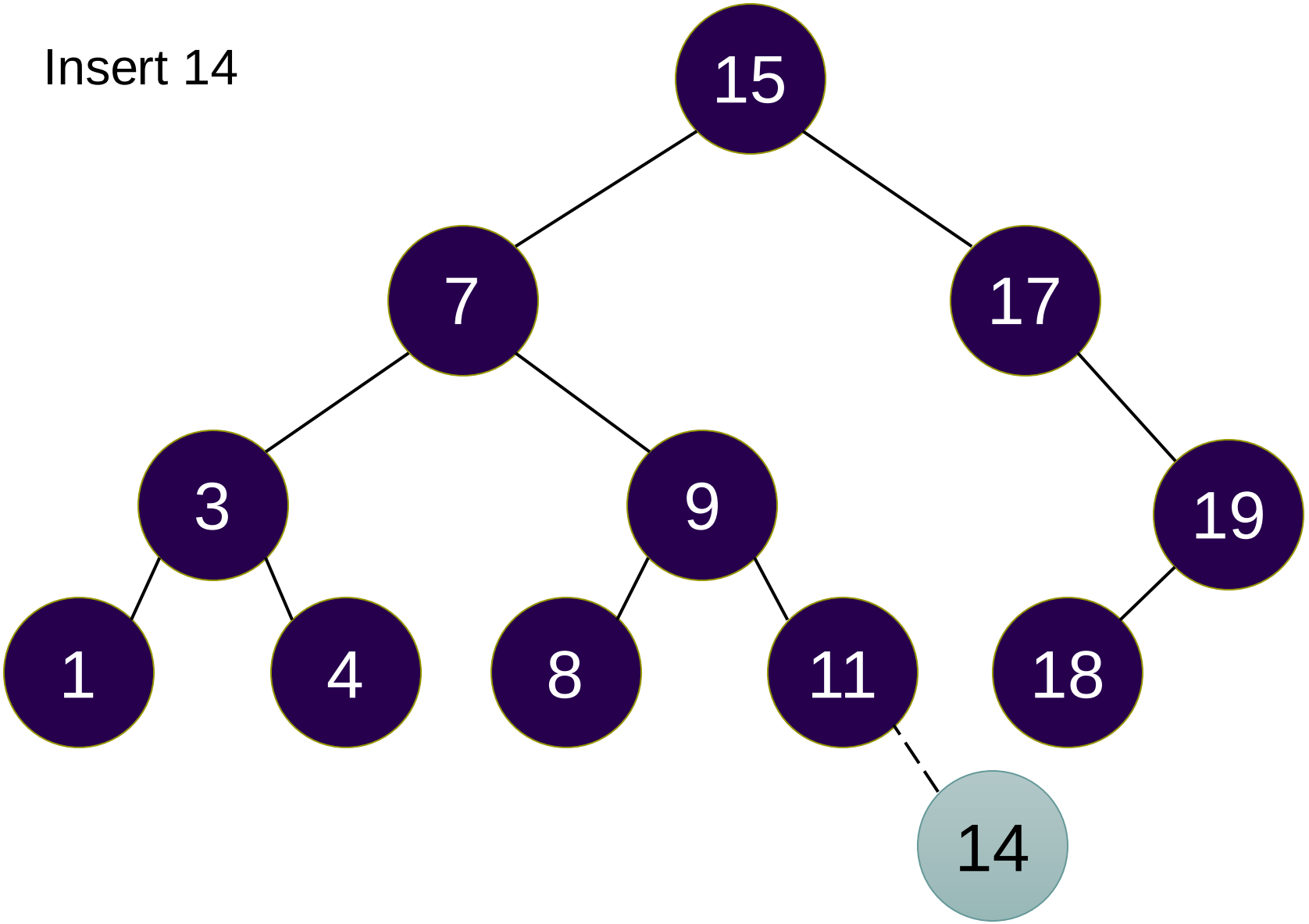
BST.Insert

Insert 18



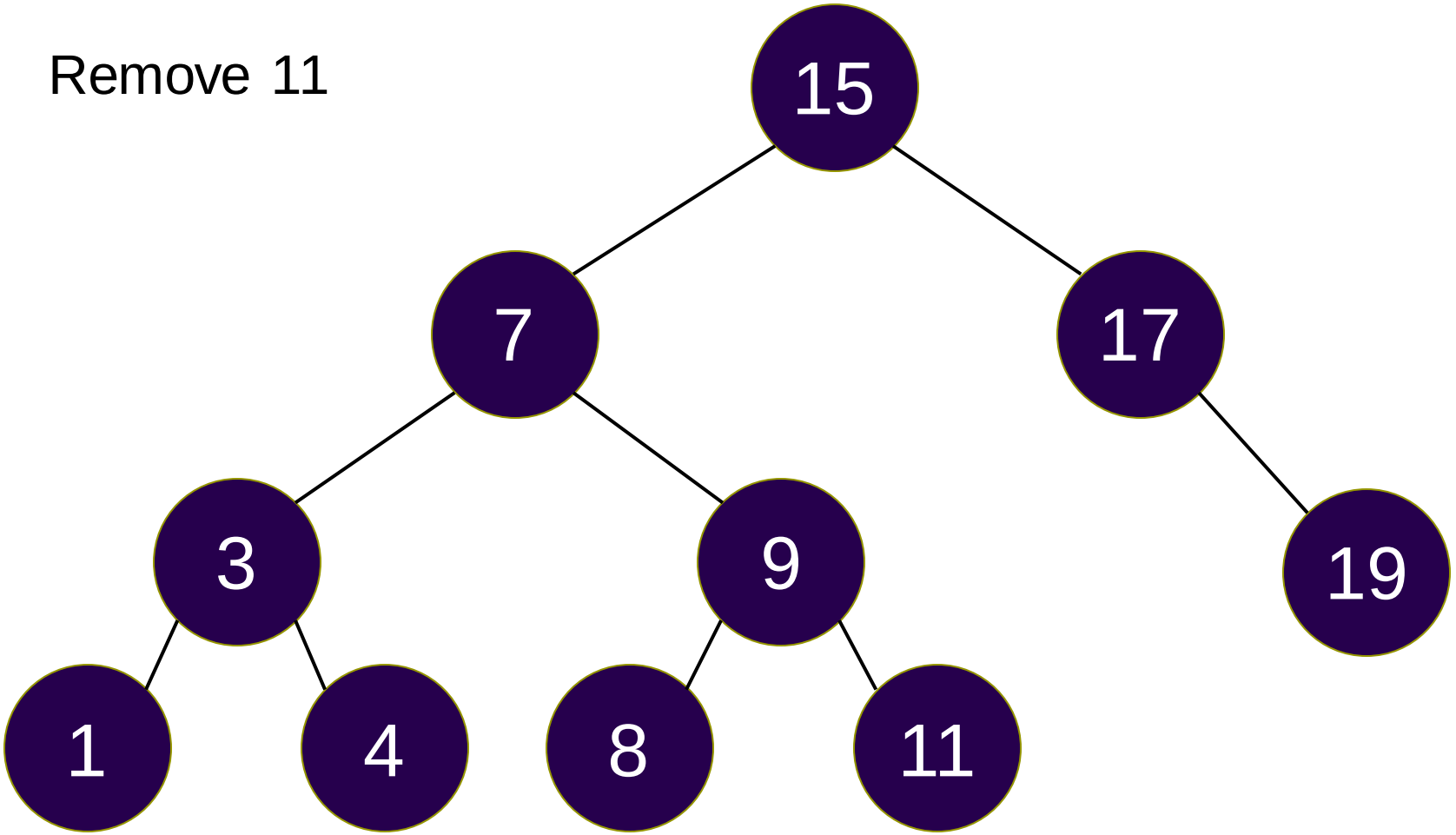
BST.Insert

Insert 14



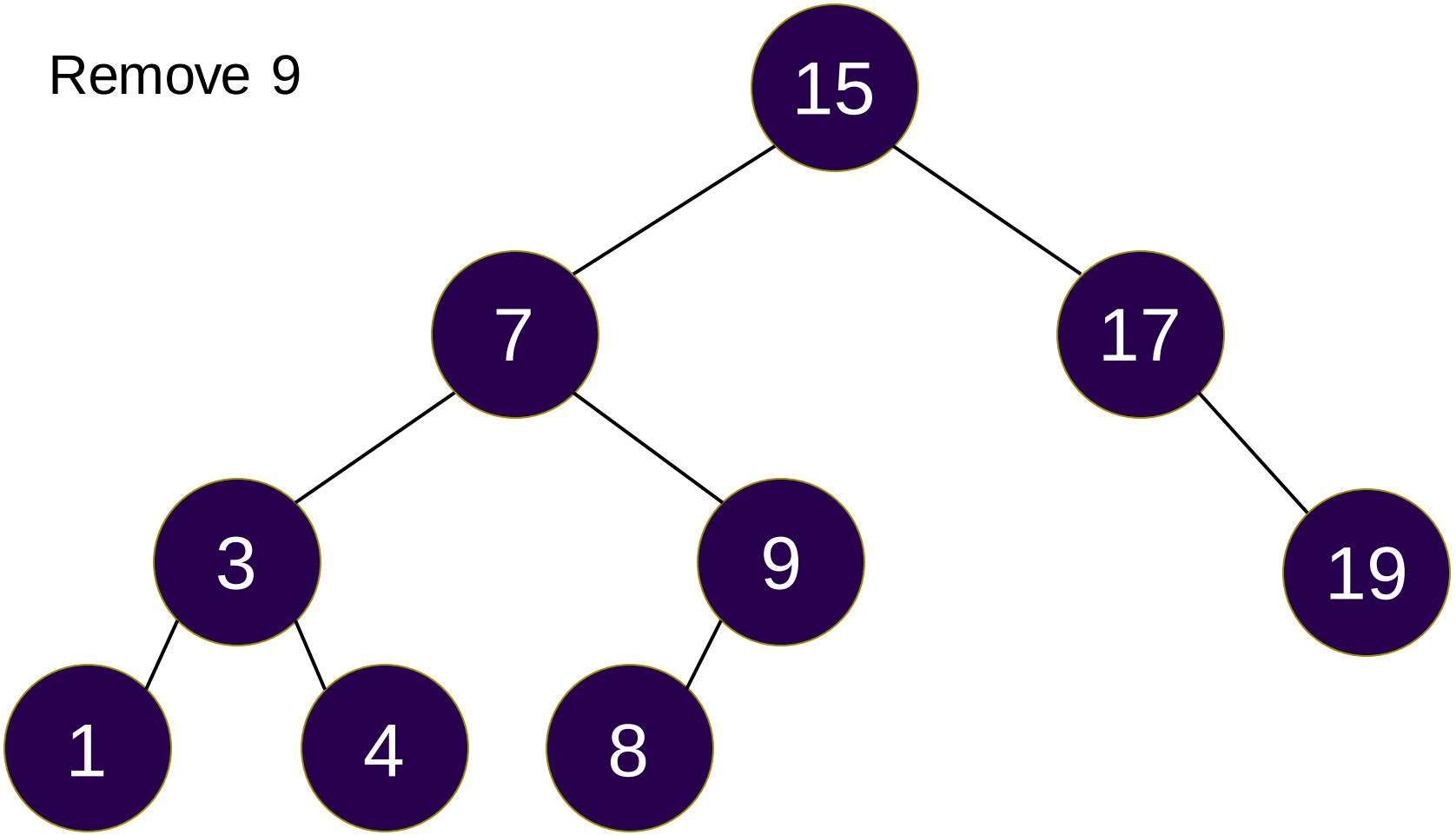
BST.Remove

Remove 11



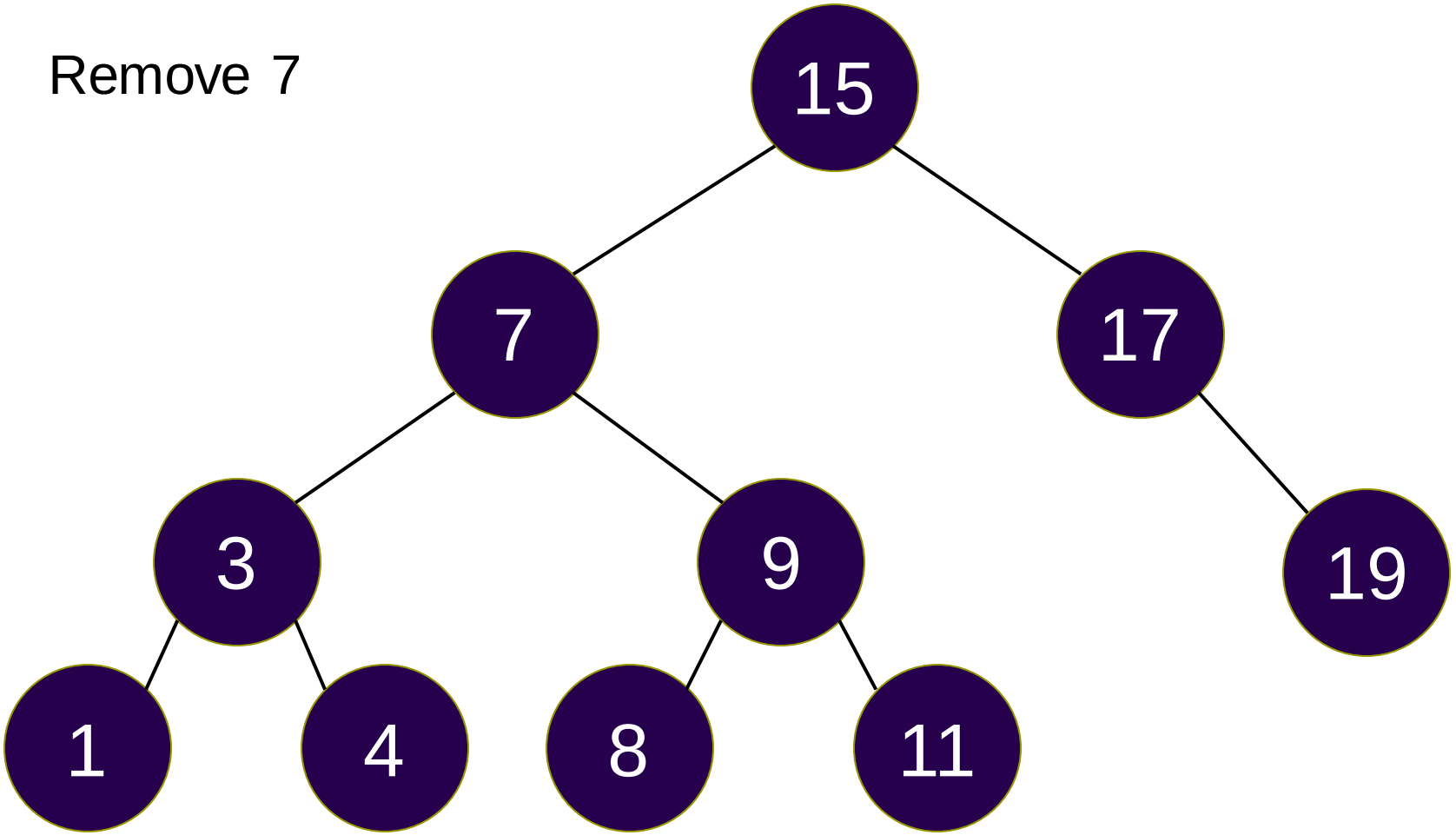
BST.Remove

Remove 9



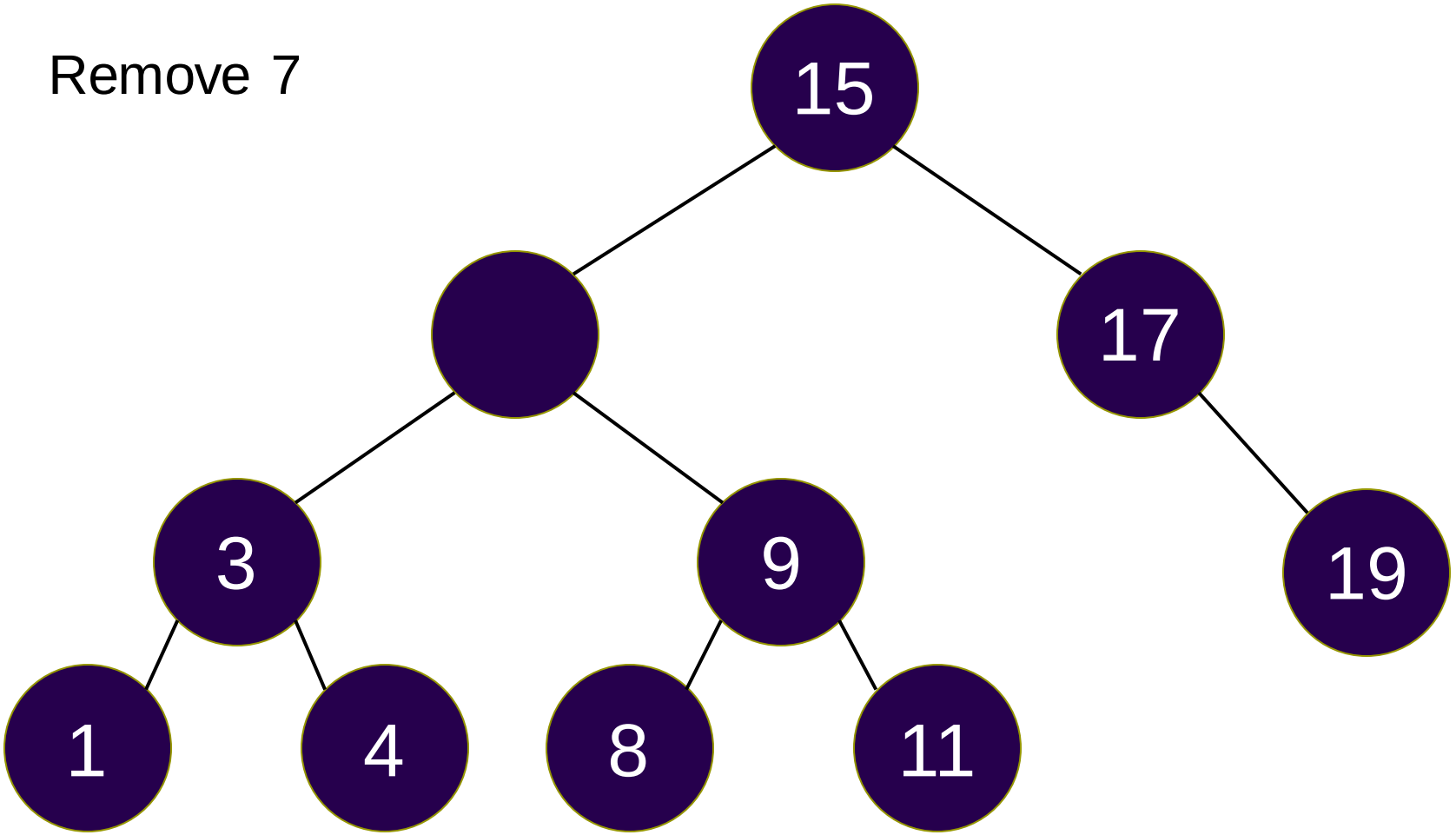
BST.Remove

Remove 7



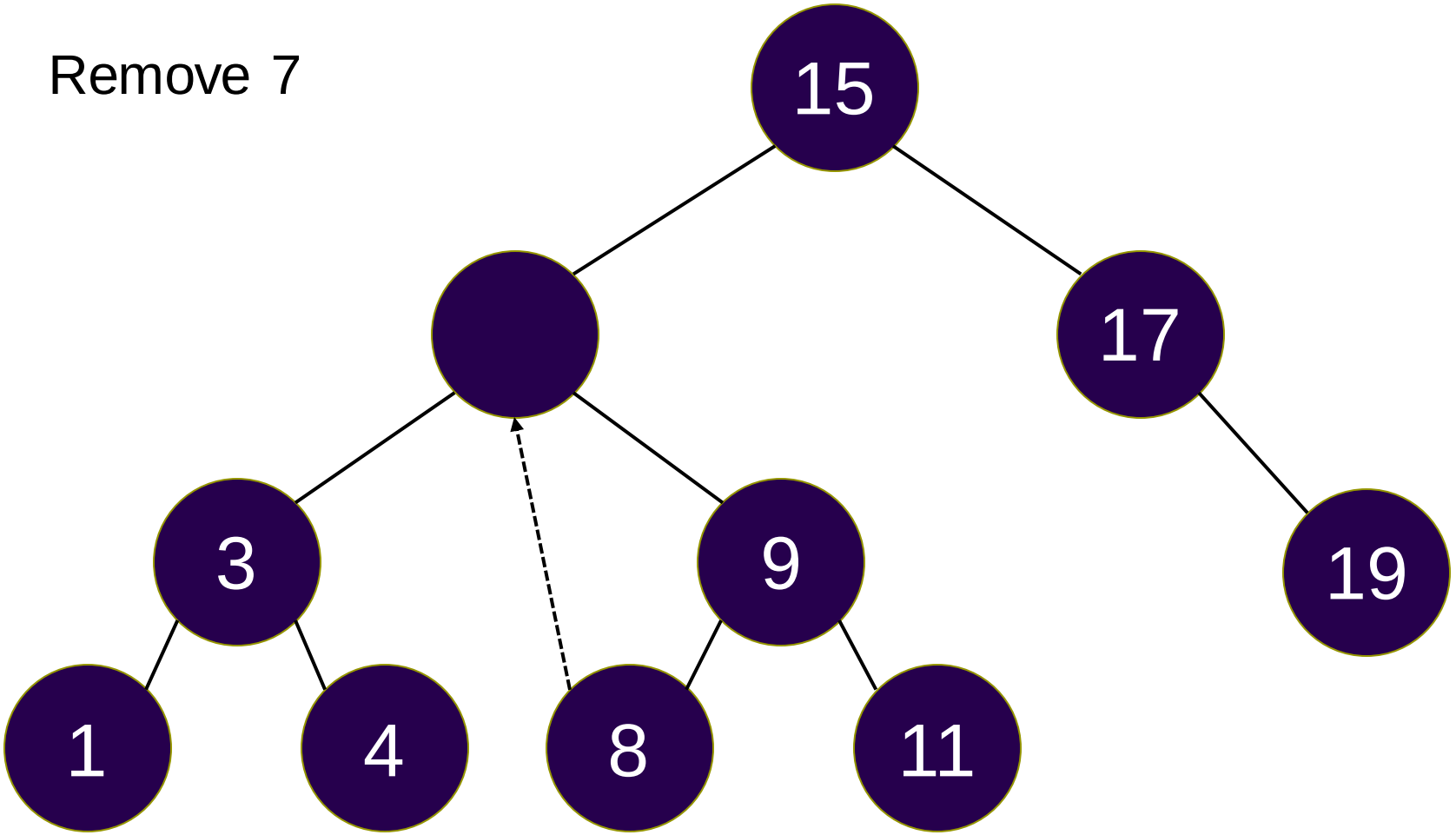
BST.Remove

Remove 7



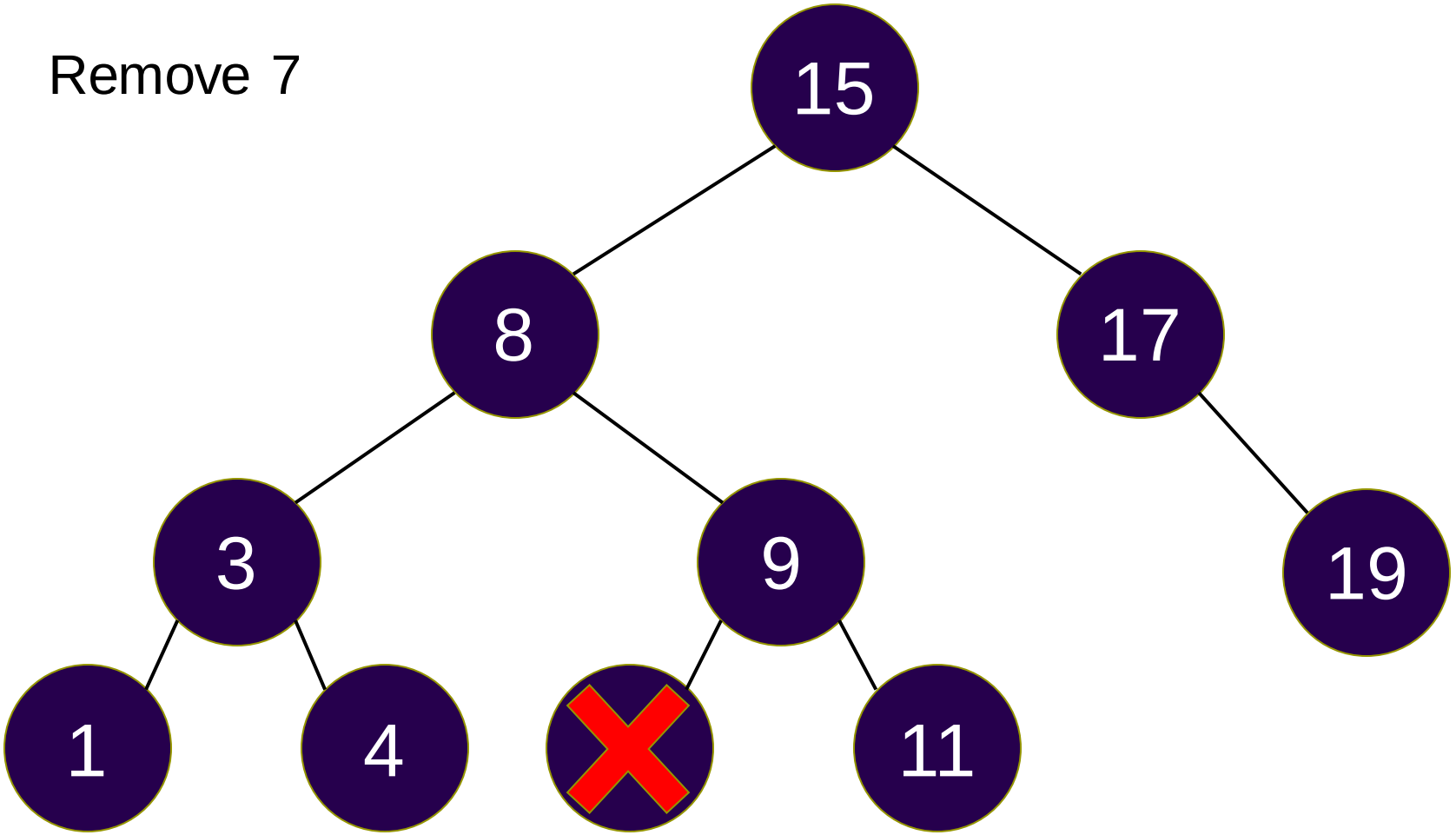
BST.Remove

Remove 7



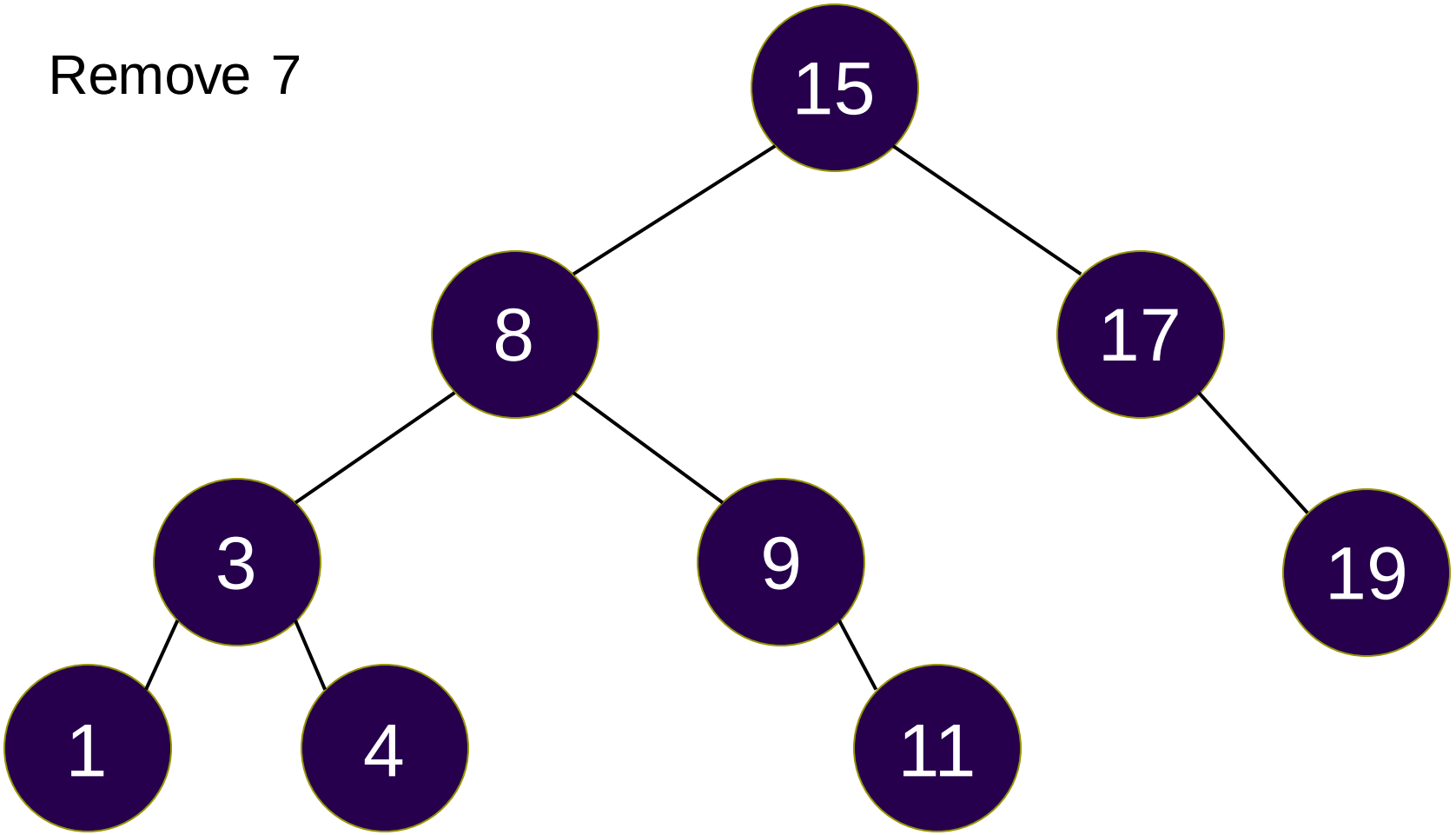
BST.Remove

Remove 7



BST.Remove

Remove 7



Analysis of BST Algorithms

- Operations:
  - Find
  - Insert
  - FindMin
  - FindMax
  - Delete
- All O(h) where h is the height of the tree
- h = O(n) (worst case)
- h = Ω(log n) (best case)
- How to guarantee O(log n) performance?