Algorithms and Data Structures for Data Science

lab_trees

CS 277

March 1, 2024

Brad Solomon

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Learning Objectives

Explore structure and use of a binary tree

Practice building programs by applying small functions to solve more complex problems

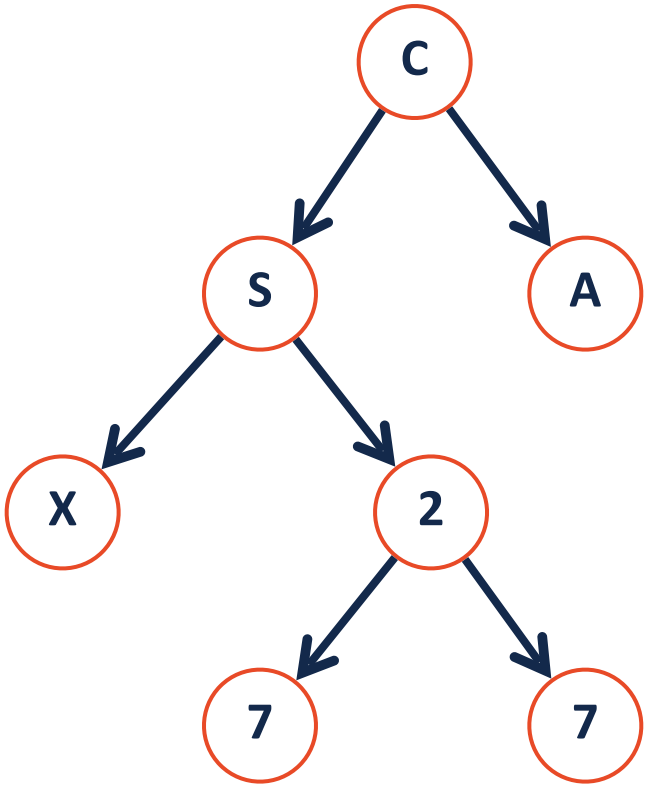
(Binary) Tree Recursion

A **binary tree** is a tree T such that:

$T = None$

or

$T = treeNode(val, T_L, T_R)$

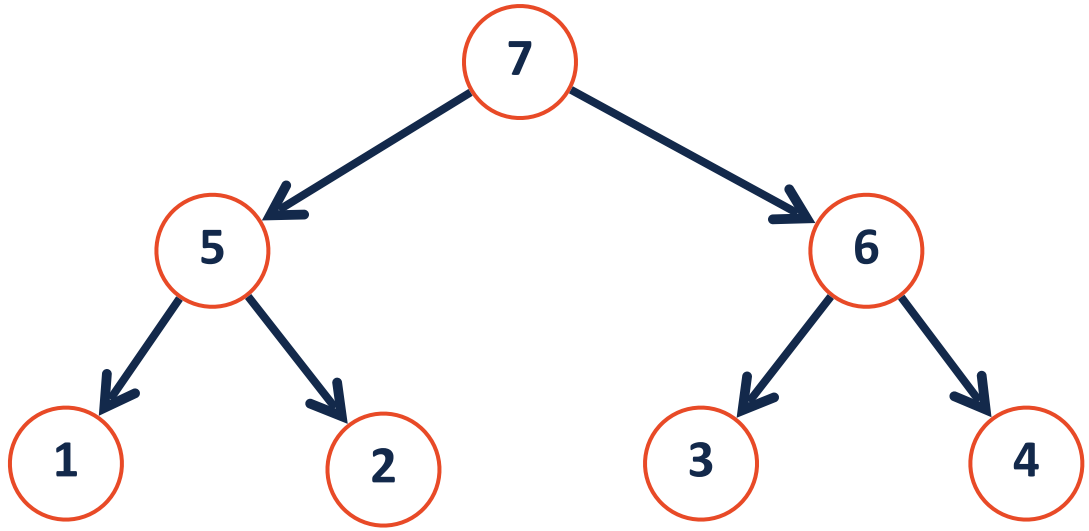



```
1  class treeNode:
2      def __init__(self, val, left=None, right=None):
3          self.val = val
4          self.left = left
5          self.right = right
```

```
1  class binaryTree:
2      def __init__(self):
3          self.root = None
4
5
```

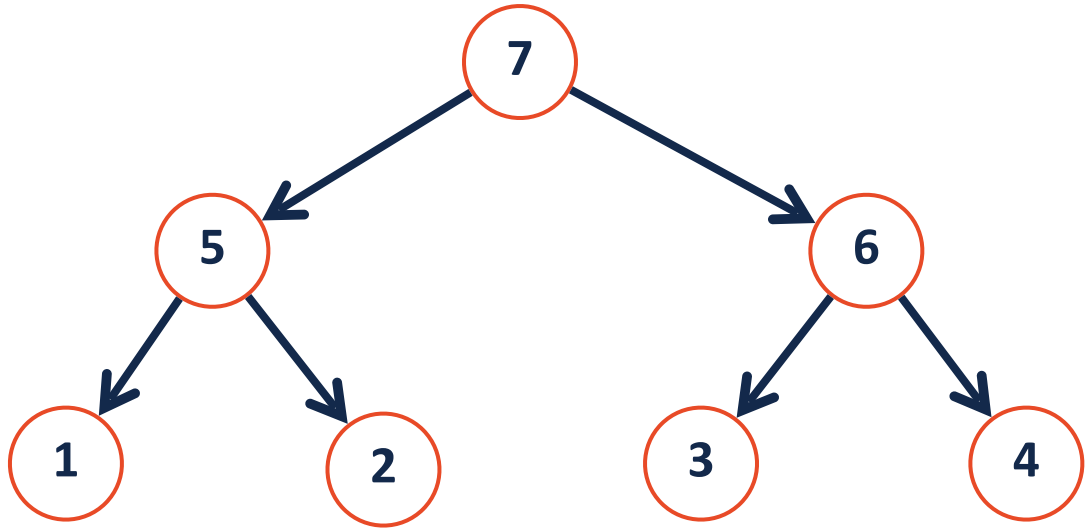
In-Order Traversal

1) Recurse left
2) Get current nodes value
3) Recurse right



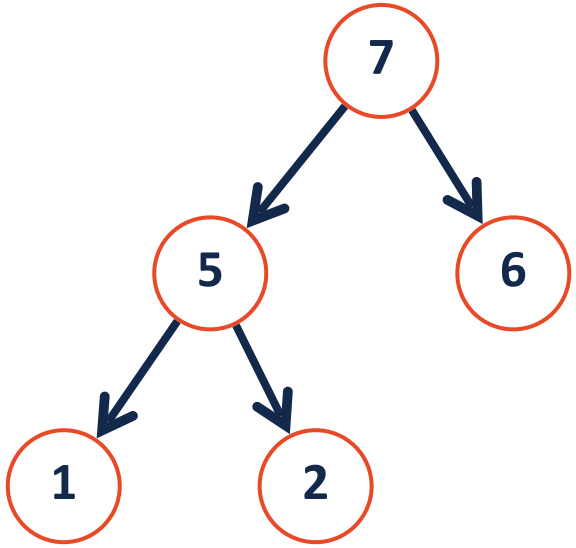
Post-Order Traversal

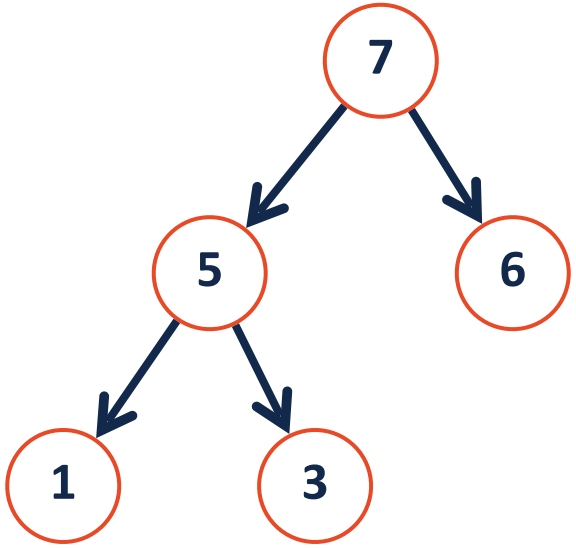
1) Recurse left
2) Recurse right
3) Get current nodes value

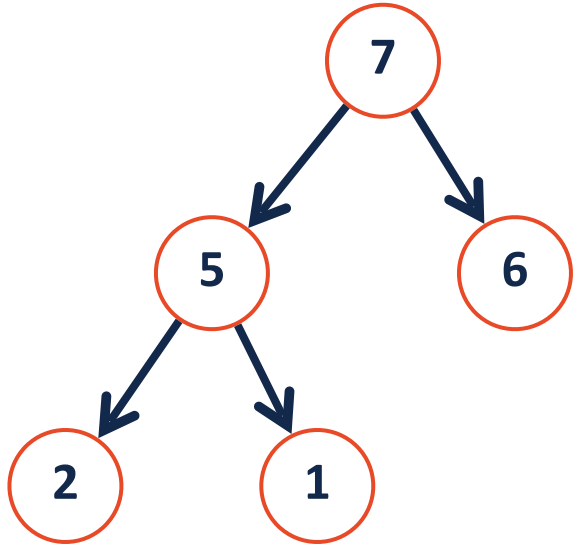


Are Equal

How can we tell if two trees are equal?

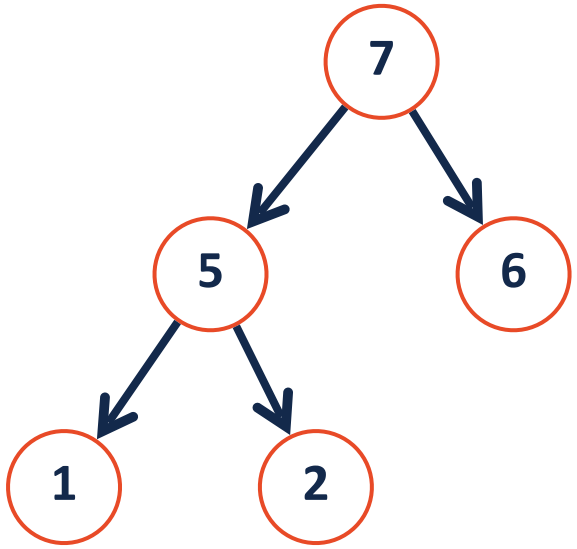


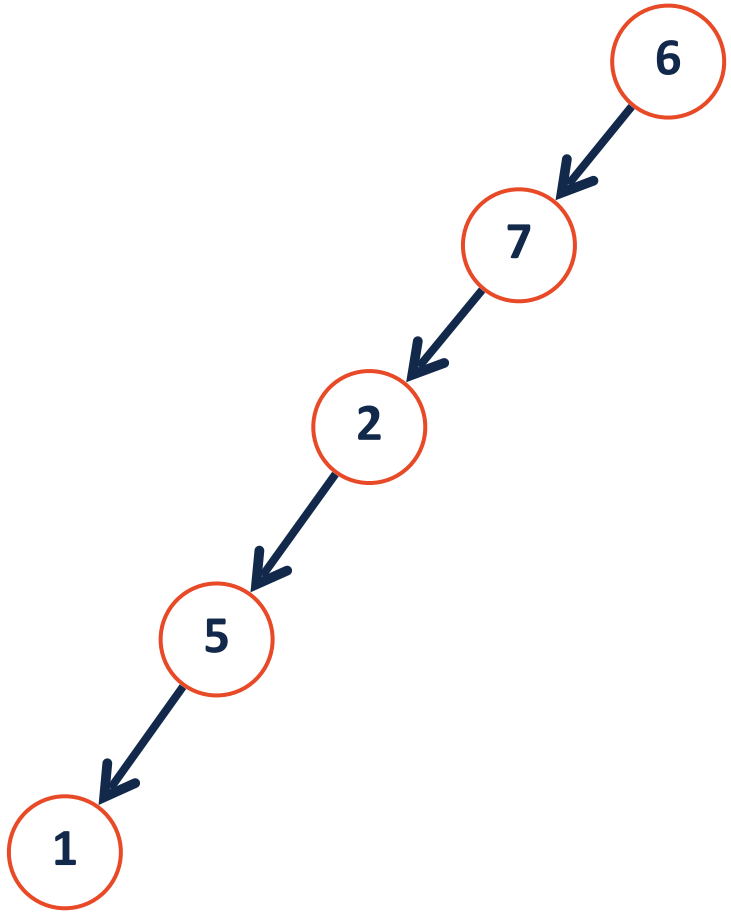




Are Equal

One traversal doesn't work.

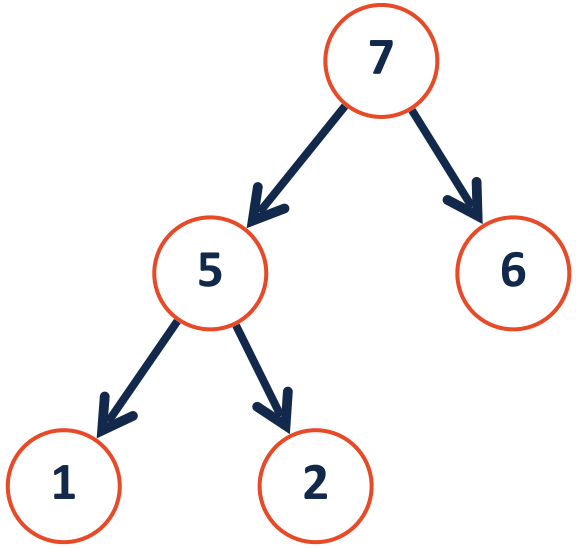




pathToNode()

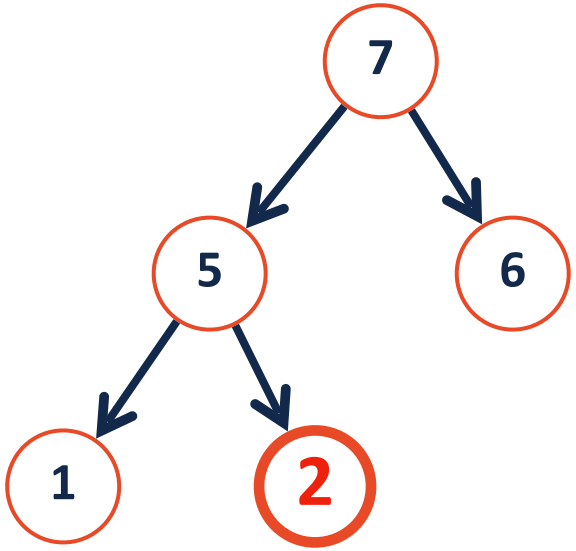
`pathToNode(<t>, 2)`

How can we return the path to a specific node?



pathToNode Visualized

```
1  def pathToNode(<7>):
2
3
4
5
6
```




```
1  def pathToNode(<5>):
2
3
4
5
6
```

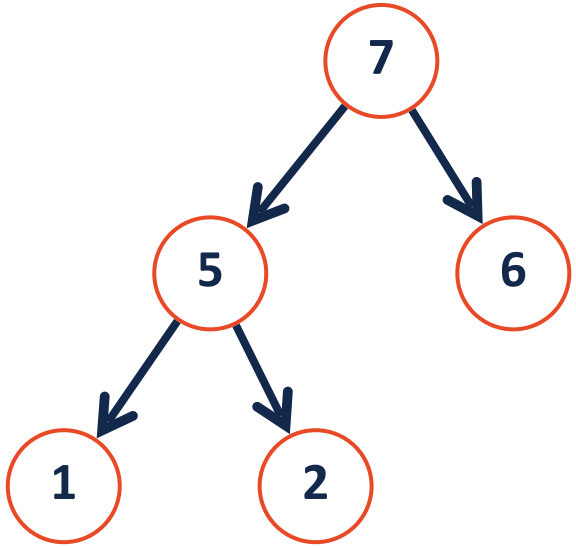
```
1  def pathToNode(<6>):
2
3
4
5
6
```

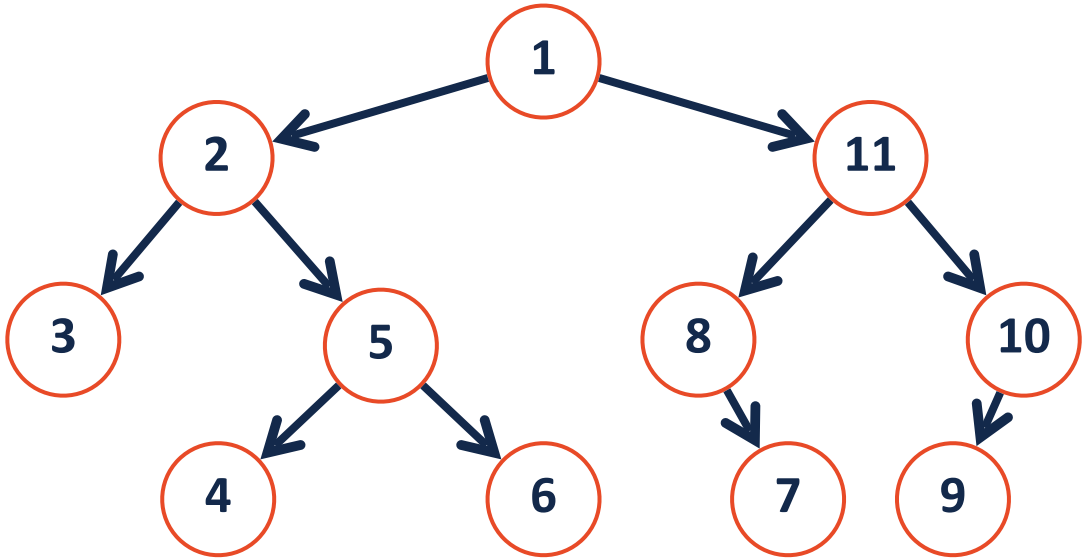
```
1  def pathToNode(<1>):
2
3
4
5
6
```

```
1  def pathToNode(<2>):
2
3
4
5
6
```

distBetweenNodes()

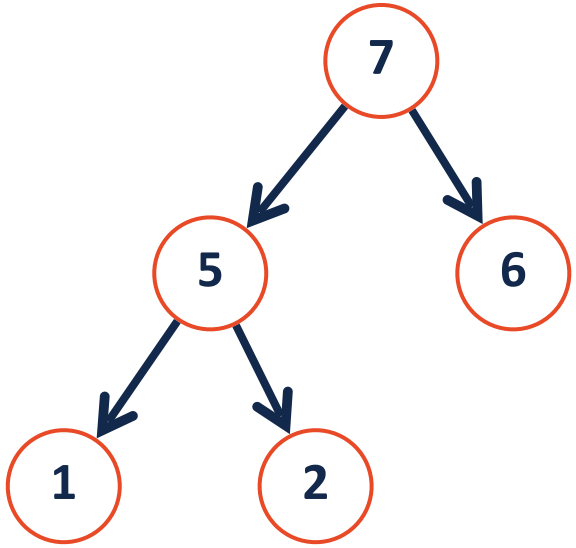
I want to compute the smallest distance between nodes.

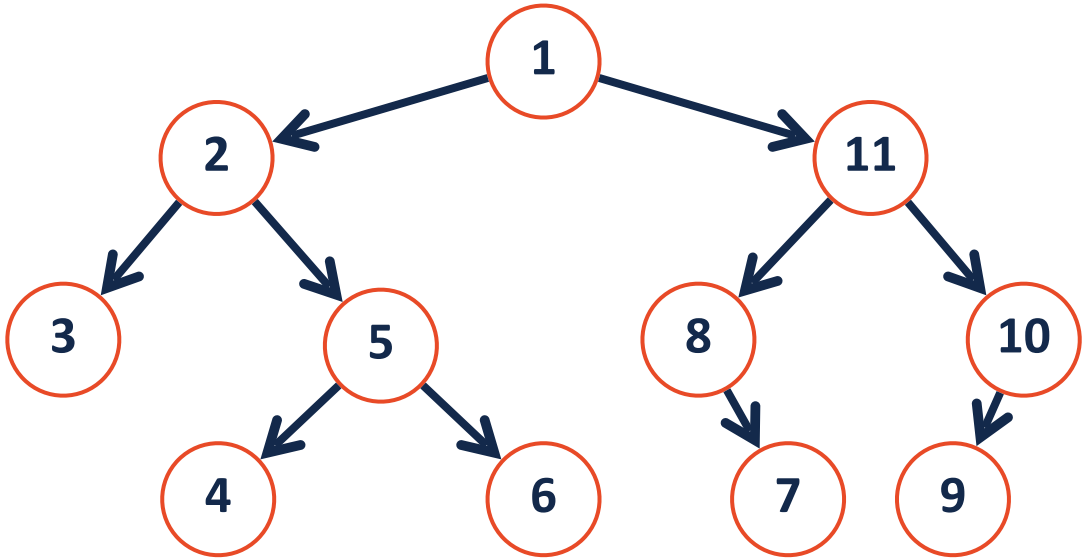




Lowest Common Ancestor

The **LCA** is the mutual parent of both nodes with the greatest depth.

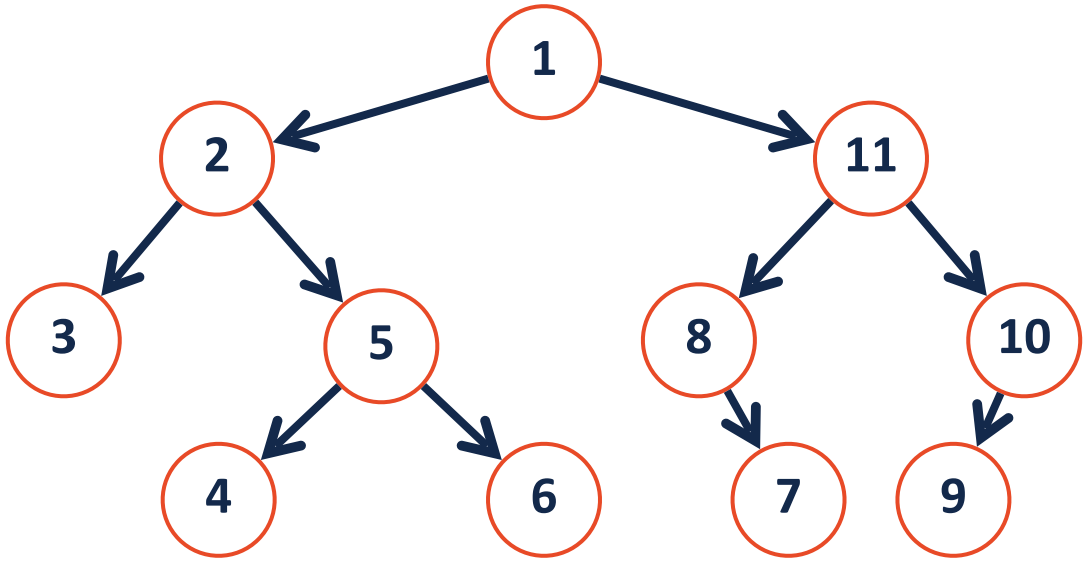




distBetweenNodes()

`dbn(<t>, 5, 7)`

I want to compute the smallest distance between nodes.



distBetweenNodes()

`dbn(<t>, 3, 6)`

I want to compute the smallest distance between nodes.