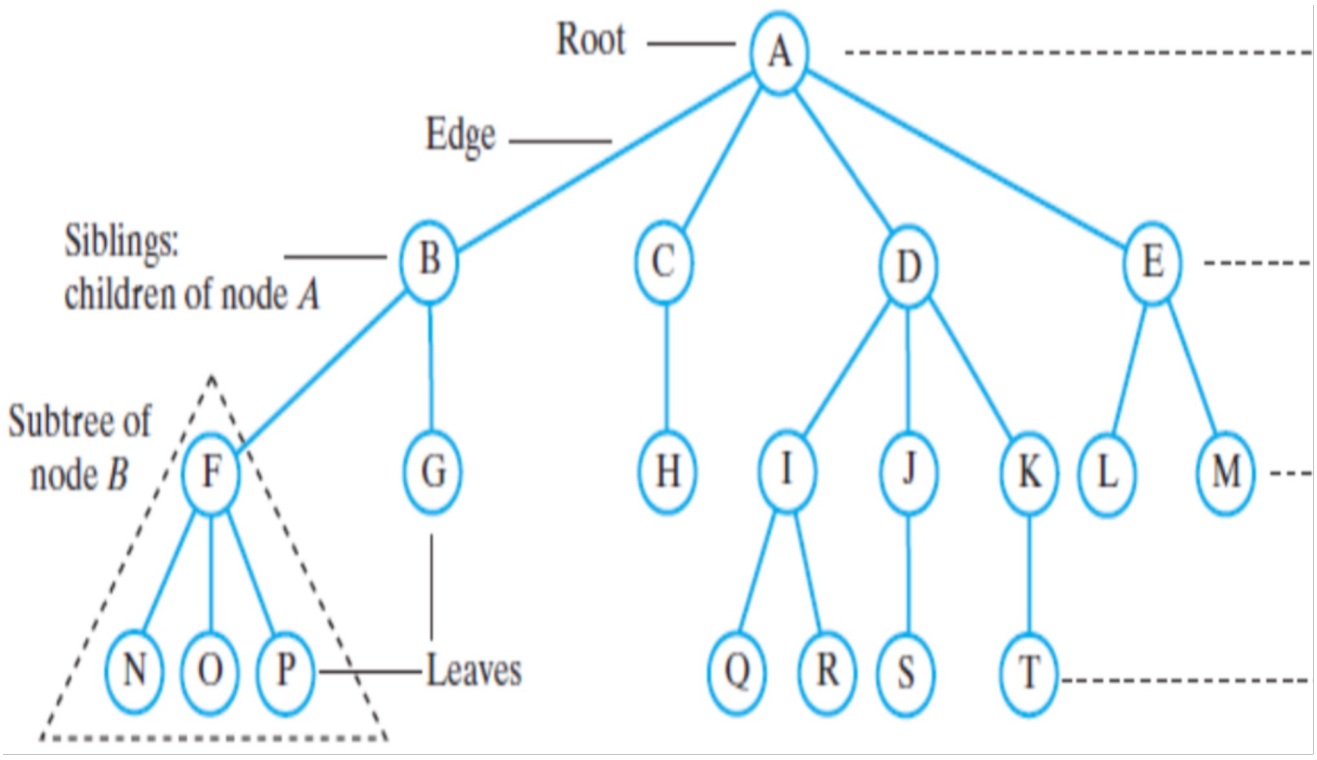
Trees and Binary Trees


Root
A
Edge
Siblings:
children of node A
B
C
D
E
Subtree of
node B
F
G
H
I
J
K
L
M
N
O
P
Leaves
Q
R
S
T

Trees:

- A natural way to think about many real-world concepts:
  - Employee hierarchy
  - Family trees
  - Directory-file structure

- Similar to Singly linked lists:
  - There is a starting node (root node)
  - Each node includes references to zero or more child nodes
  - Nodes don't typically have references to the parent nodes.
  - So we can only traverse the tree from a node to its children…not the other way around….
    - just like Singly linked lists

Linear vs. non-linear

- Linked lists arrays, stacks, queues, priority queues are linear.
- Trees are non-linear, because they allow branching.

Trees versus Graphs

- Trees can have at most one path from one node to a node farther down the tree
- Graphs can have multiple paths between nodes

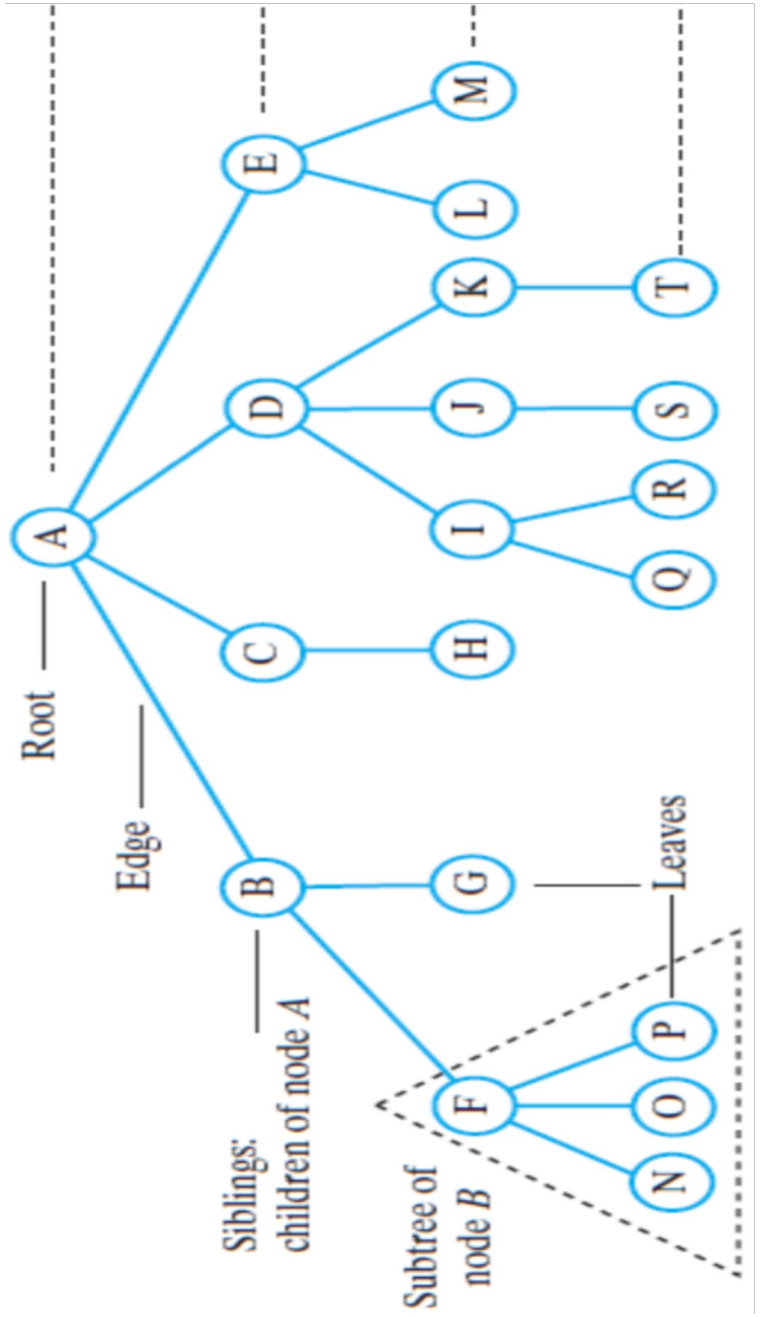
Tree terminology:

- Basics:
  - node
  - edge
- Kinds of nodes:
  - root
  - parent, child
  - ancestor, descendant
  - leaf, interior
- Tree properties:
  - level
  - Length
  - subtree
  - path

- Attributes of trees:
  - Binary trees
  - binary search trees / sorted trees
  - heaps

Algorithms for trees

- traversal algorithms for visiting all nodes
  - pre-order
  - post-order
  - in-order
  - level-order
- counting nodes
- Calculating length
- searching



- What is the length?
- How many paths are there from A to R? from R to A? from Q to T?

Can a tree have more than one root?

A. Yes
B. No

Can a tree have more than one root?

A. Yes
B. No

In a tree with 10 nodes, how many of those nodes are children?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many of those nodes are children?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many children does the root node have?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many children does the root node have?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many descendants does the root node have?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many descendants does the root node have?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many of those nodes are parents?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, how many of those nodes are parents?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, what is the largest possible length of the tree?

A. 0
B. 1
C. 9
D. 10
E. It can't be determined without more information

In a tree with 10 nodes, what is the largest possible length of the tree?

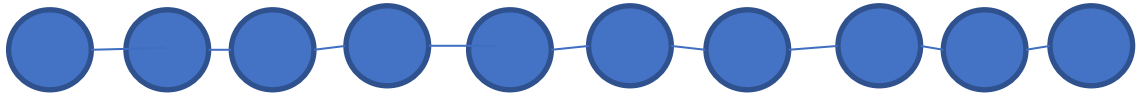
A. 0

B. 1

C. 9

D. 10

E. It can't be determined without more information

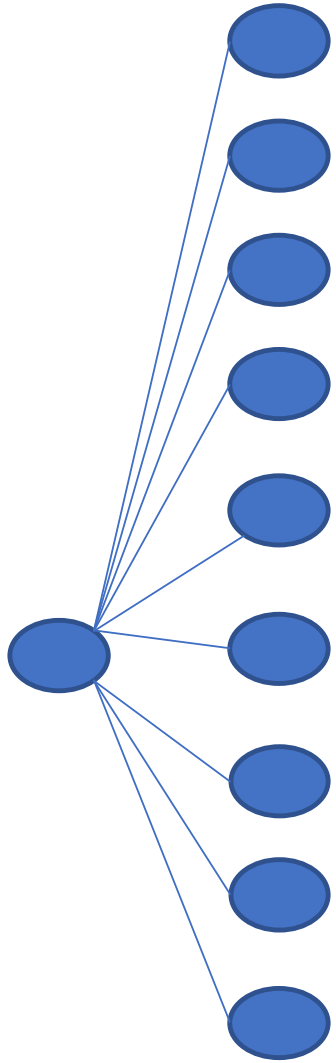


In a tree with 10 nodes, what is the smallest possible length of the tree?

A. 0
B. 1
C. 2
D. 5
E. It can't be determined without more information

In a tree with 10 nodes, what is the smallest possible length of the tree?

A. 0
B. 1
C. 2
D. 5
E. It can't be determined without more information



Given two nodes, A and B, in a tree: how many paths will there be between A and B?

A. At least 1
B. At most 1
C. Exactly 1
D. None of the above

Given two nodes, A and B, in a tree: how many paths will there be between A and B?

A. At least 1
B. At most 1
C. Exactly 1
D. None of the above

How to implement a general tree:

- Node class. Each node has:
  - Data
  - References to its children

Each Node Having One Child

```
class Node {
     int value;
     Node child;

     // constructor and methods
}
```

Each Node Having Three Children

```
class Node {
     int value;
     Node child1, child2, child3;

     // constructor and methods
}
```

Each Node Having multiple Children

```
class Node {
     int value;
     Node[] children;

     // constructor and methods
}
```

Binary trees:

- The same as the trees we've been talking about, with one restriction:
  - Each node can have at most two children.
    - Up child
    - Down child

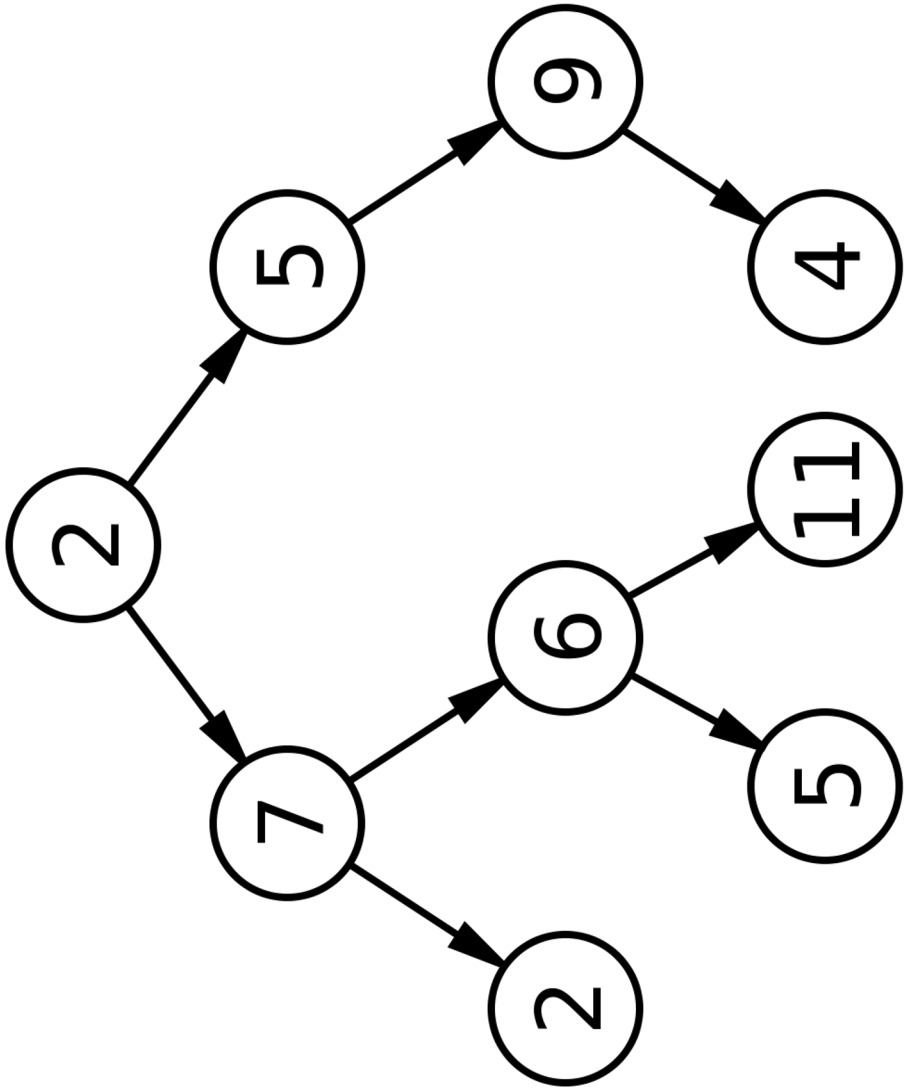
Binary tree node:

```
class Node {
     int value;
     Node down, up;

     // constructor and methods
}
```

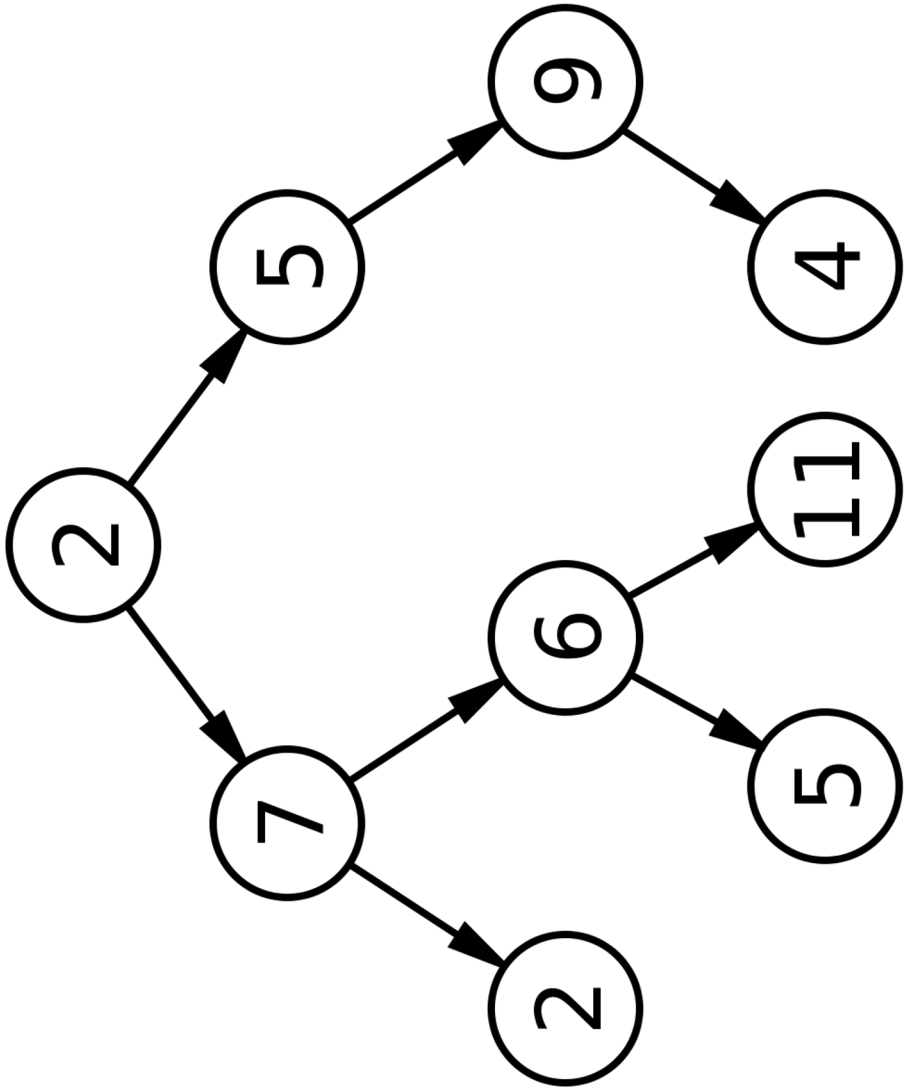
What is the length of this binary tree?

A. 3
B. 4
C. 9
D. None of these



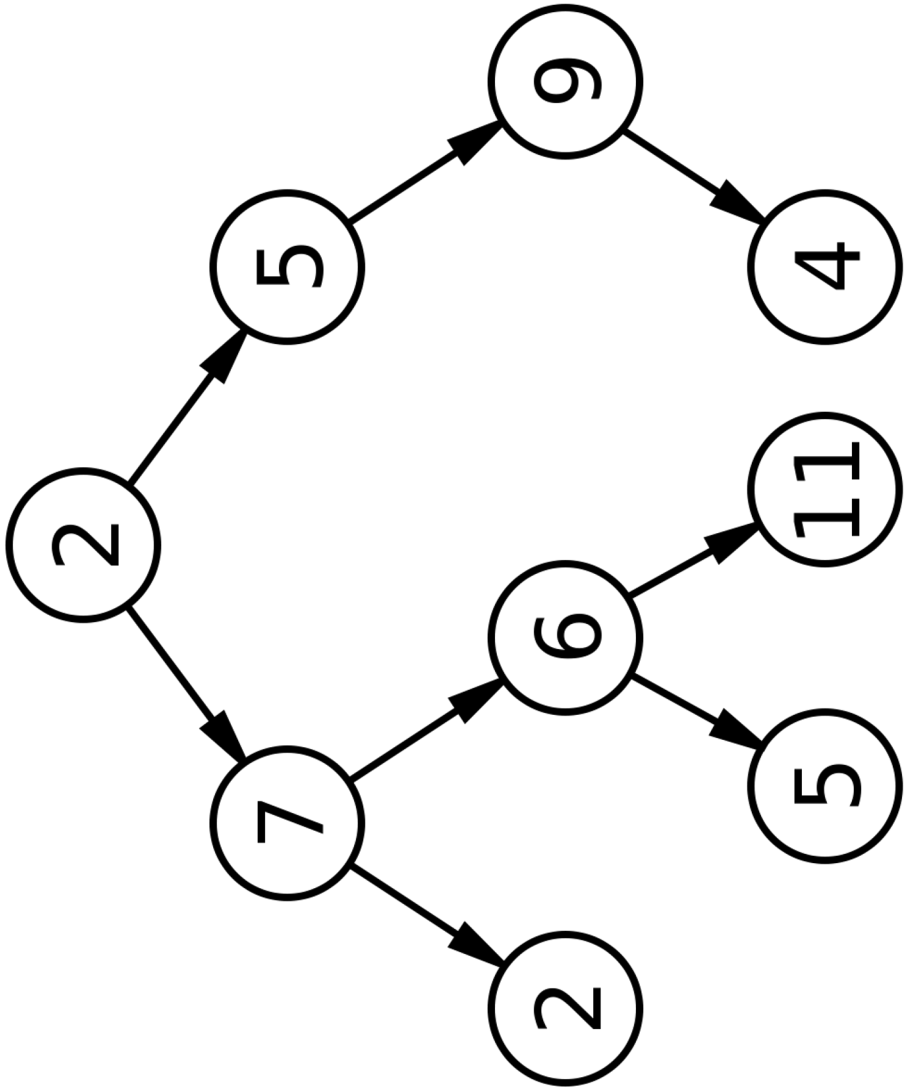
What is the length of this binary tree?

A. 3
B. 4
C. 9
D. None of these



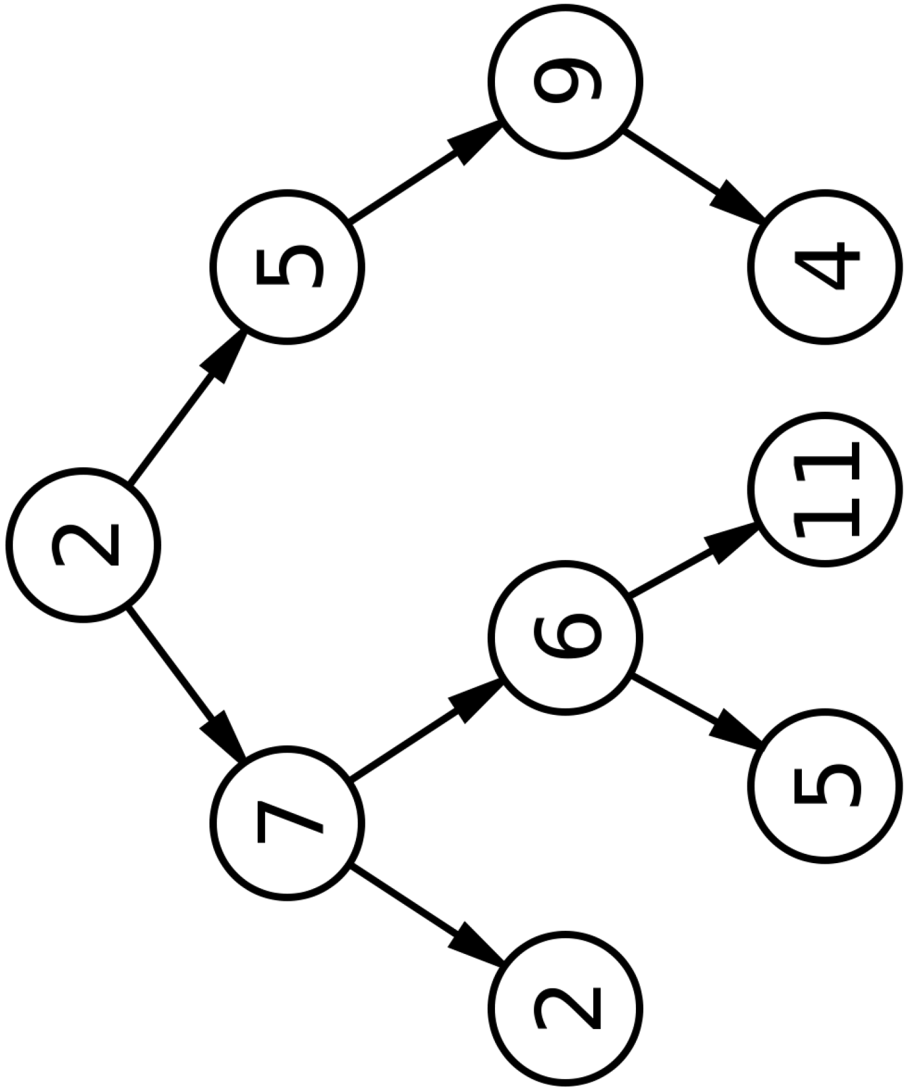
How many leaves are in this binary tree?

A. 3
B. 4
C. 5
D. None of these



How many leaves are in this binary tree?

A. 3
B. 4
C. 5
D. None of these



A binary tree has 30 nodes. What is its largest possible length?

A. 14
B. 15
C. 29
D. 30
E. None of these

A binary tree has 30 nodes. What is its largest possible length?

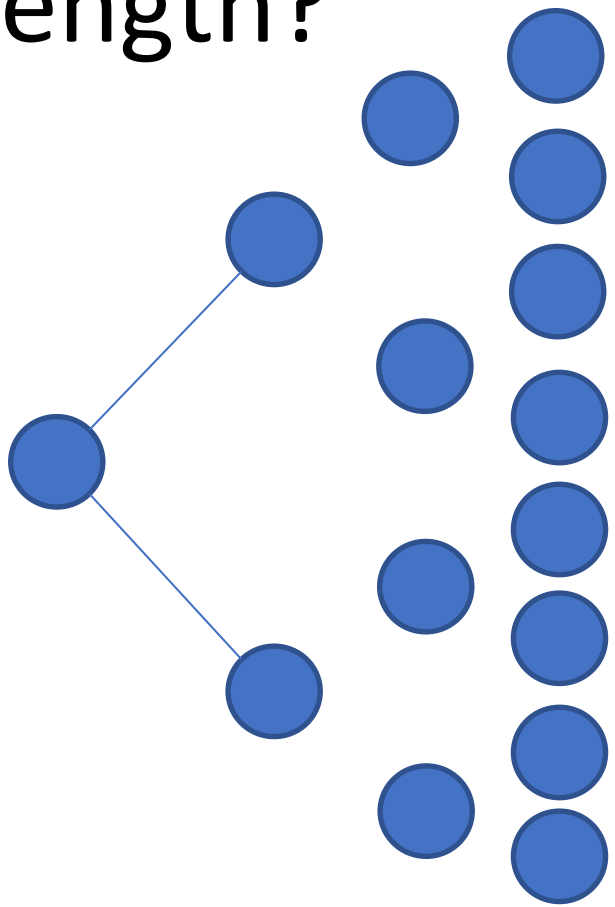
A. 14
B. 15
C. 29
D. 30
E. None of these

A binary tree has 30 nodes. What is its smallest possible length?

A. 3
B. 4
C. 5
D. 6
E. None of these

A binary tree has 30 nodes. What is its smallest possible length?

A. 3
B. 4
C. 5
D. 6
E. None of these

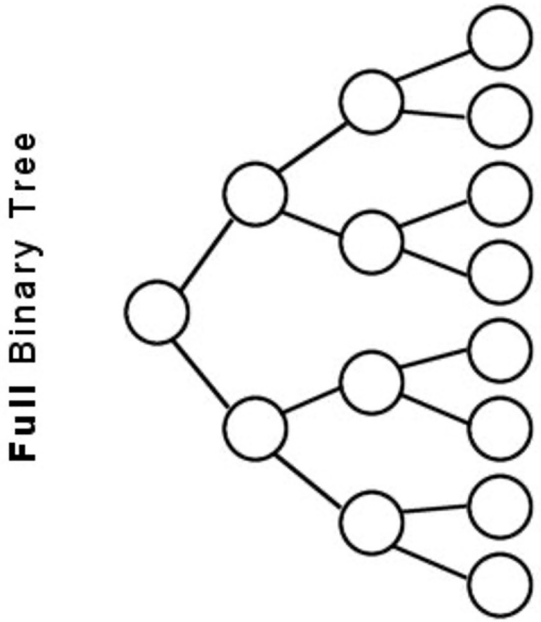


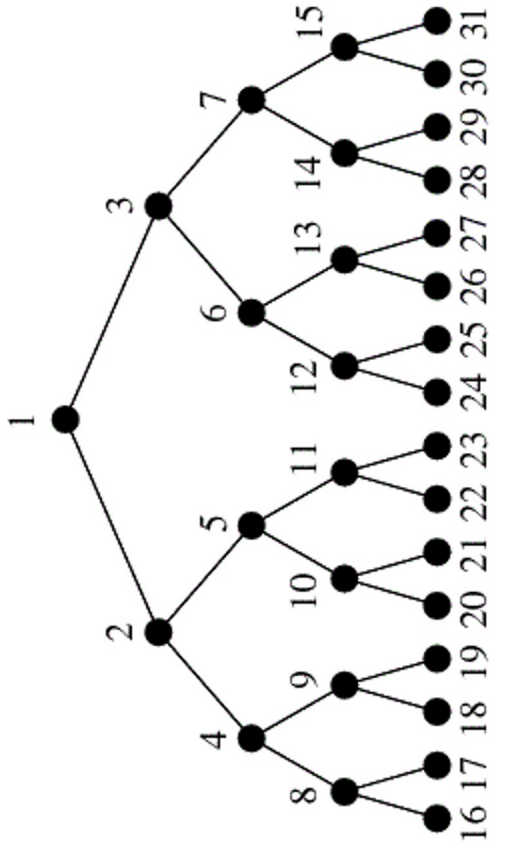
Attributes of Binary Trees

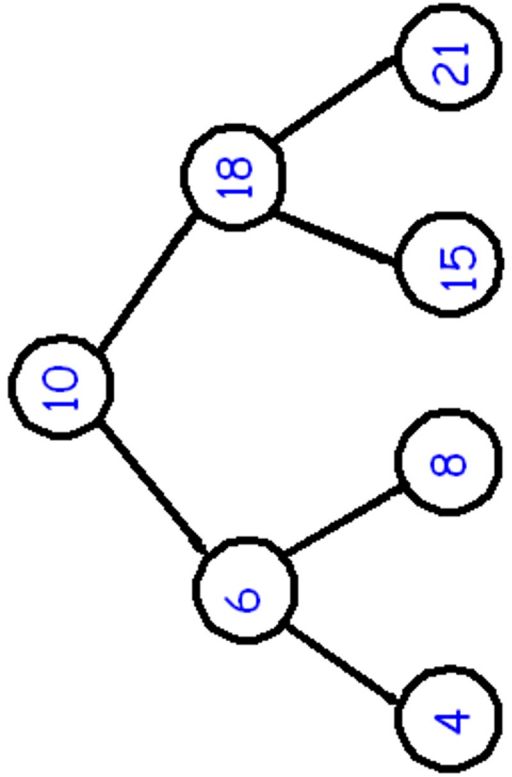
- There are a few speciall categories of binary trees
  - Full
  - Complete
  - Balanced

Full binary trees:

- Every level has all possible nodes
- All leaf nodes are at the righthand side
- Nodes on all the other levels have two children







How many nodes are in a full binary tree with a length of 6?

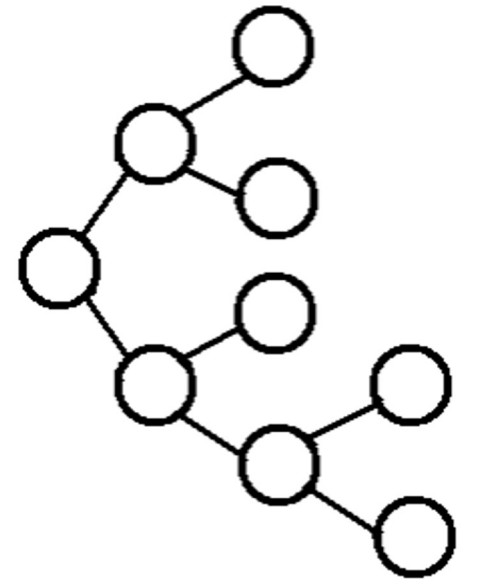
A. 15
B. 31
C. 63
D. 127

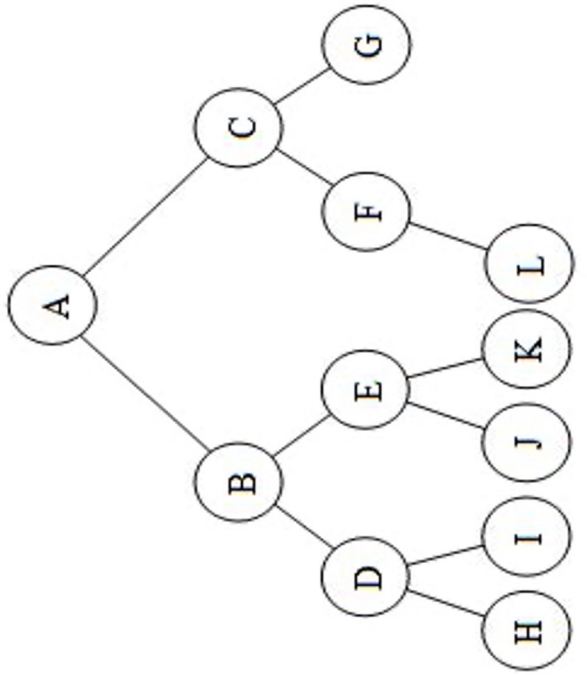
How many nodes are in a full binary tree with a length of 6?

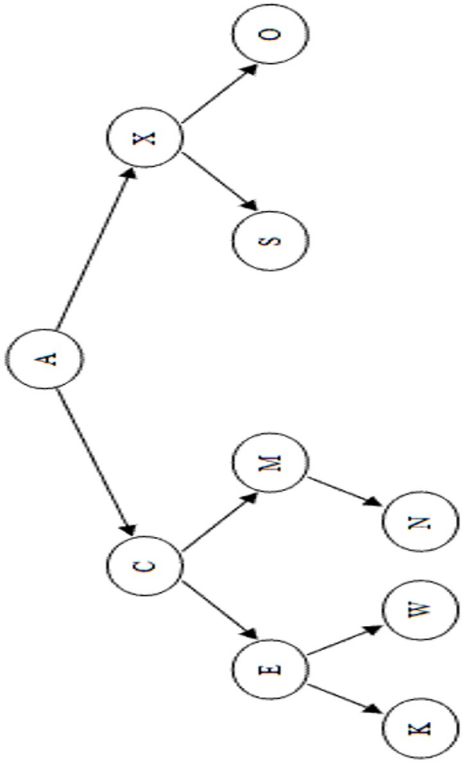
A. 15
B. 31
C. 63
D. 127

Complete binary trees:

- Except for the last level, the tree is full
- Looking at the last level, there is no gap in the bottom







What is the fewest number of nodes in a complete binary tree with length 6?

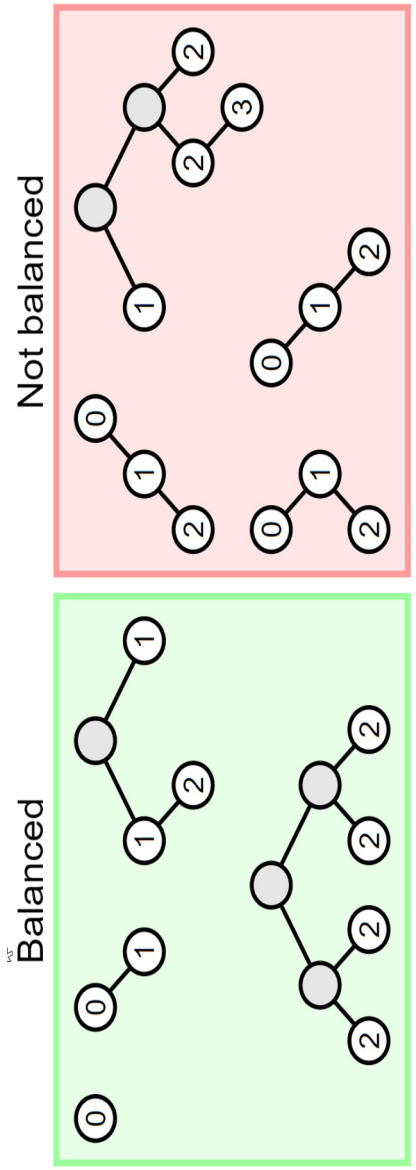
A. 31
B. 32
C. 63
D. 64
E. None of these

What is the fewest number of nodes in a complete binary tree with length 6?

A. 31
B. 32
C. 63
D. 64
E. None of these

Balanced binary trees:

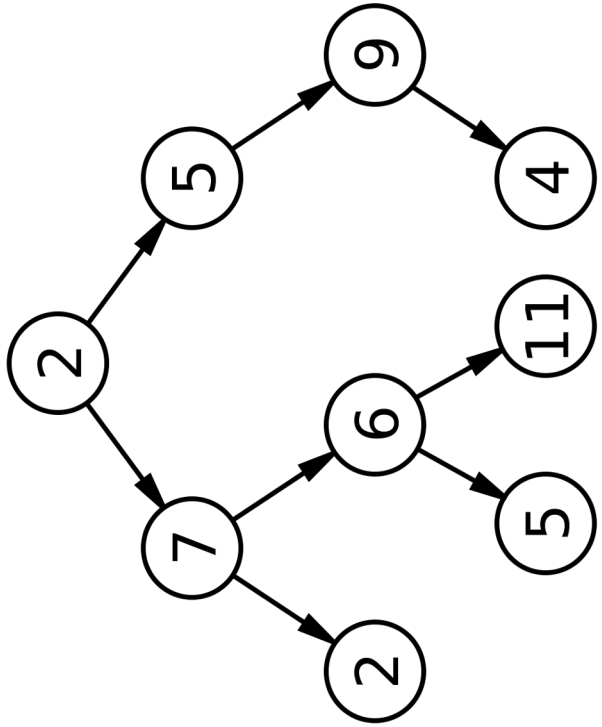
- Also known as length-balanced
- For each node, the length of its subtrees are either:
  - equal, or
  - differ by 1



http://www.growingwiththeweb.com/2015/11/check-if-a-binary-tree-is-balanced.html

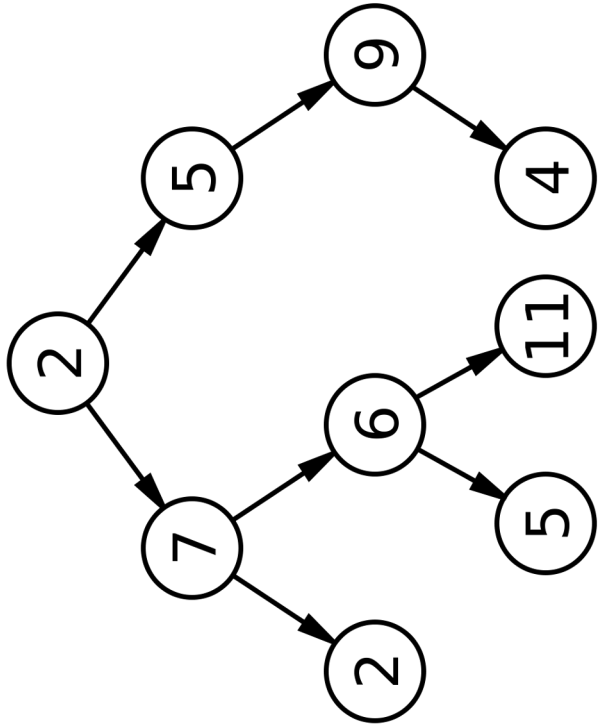
What happens if we call print(root)?

```
public void print(Node mynode) {
    Node current = mynode;
    while (current != null) {
        System.out.print(current.data + " ");
        current = current.up;
    }
}
```



What happens if we call print(root)?

```
public void print(Node mynode) {
    Node current = mynode;
    while (current != null) {
        System.out.print(current.data + " ");
        current = current.up;
    }
}
```



2 5 9

I have a binary tree.

The root's down subtree has 14 nodes.

The root's up subtree has 10 nodes.

How many nodes are in the tree?

A. 10

B. 14

C. 24

D. 25

Counting all nodes in a tree:

- The number of nodes in a tree with a given root is:
  - The number of nodes in root.down subtree, plus
  - The number of nodes in root.up subtree, plus
  - 1

- The above note suggests a recursive algorithm for counting all nodes in a tree: nodeCounter (root)

- `nodeCounter(mynode) = 1 + nodeCounter(mynode.down) + nodeCounter(mynode.up)`

Recursive Algorithms:

- Counting nodes
- Computing length
- Traversing all nodes

I have a binary tree.

The root's down subtree has a length of 7.

The root's up subtree has a length of 10.

What is the length of my binary tree?

A. 7
B. 8
C. 10
D. 11
E. 17
F. 18

I have a binary tree.

The root's down subtree has a length of 7.

The root's up subtree has a length of 10.

What is the length of my binary tree?

A. 7
B. 8
C. 10
D. 11
E. 17
F. 18

Computing the length of a tree:

- The length of a tree with a given root is:
  - the length of the largest subtree, plus
  - 1

- The above note suggests a recursive algorithm for Calculating the length of a tree: Length(root)

  - `Length(mynode) = 1 + max(Length(mynode.down), Length(mynode.up))`