Quiz 2

Due Mar 12 at 23:59   **Points** 10   **Questions** 10

Available Mar 5 at 0:00 - Mar 12 at 23:59   **Time Limit** None

Allowed Attempts Unlimited

Instructions

You can take this quiz as **many** times as you like before its due date.

Your mark will be based on your **last attempt**. All previous attempts will be ignored.

If you're an overseas student connecting via VPN (or just want the some extra assurance), download **Q2-orderedtree-fig.jpg** (https://canvas.sydney.edu.au/courses/48384/files/29695446/download?wrap=1) ⬇ (https://canvas.sydney.edu.au/courses/48384/files/29695446/download?download_frd=1) before starting, in case it doesn't load properly in the quiz itself.

<div align="center">

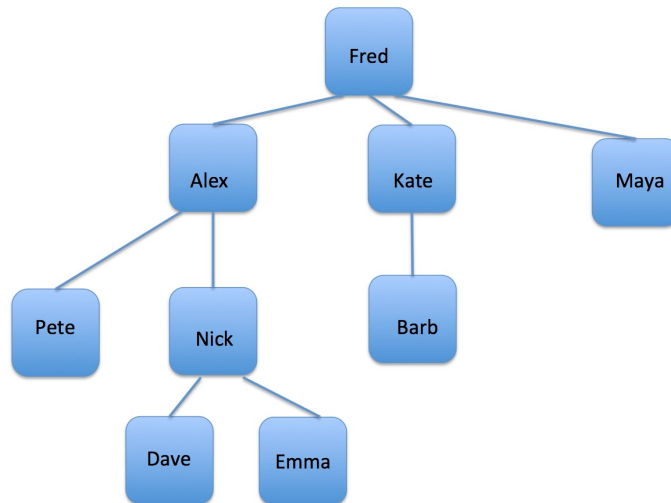⎡ **Take the Quiz Again** ⎤

</div>

Attempt History

	Attempt	Time	Score
LATEST	**Attempt 1**	14 minutes	7 out of 10

⚠ Correct answers are hidden.

Score for this attempt: **7** out of 10
Submitted Mar 7 at 9:38
This attempt took 14 minutes.

Incorrect	**Question 1**	**0 / 1 pts**

Consider the ordered tree shown below. What is the height of the tree?

- ○ 3
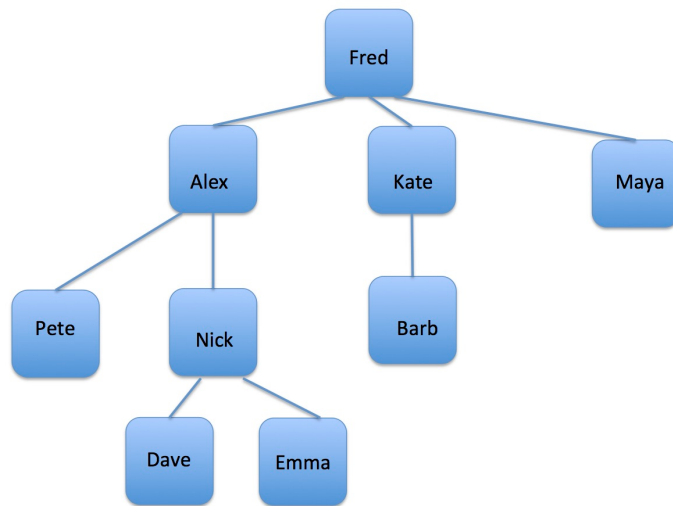- ◉ 4
- ○ 2
- ○ 1

Question 2

1 / 1 pts

Consider the ordered tree shown below. What is the depth of the node labeled Barb?

- ⦿ 2
- ○ 1
- ○ 3
- ○ 0

Question 3   1 / 1 pts

How many children can a node have in a binary tree?

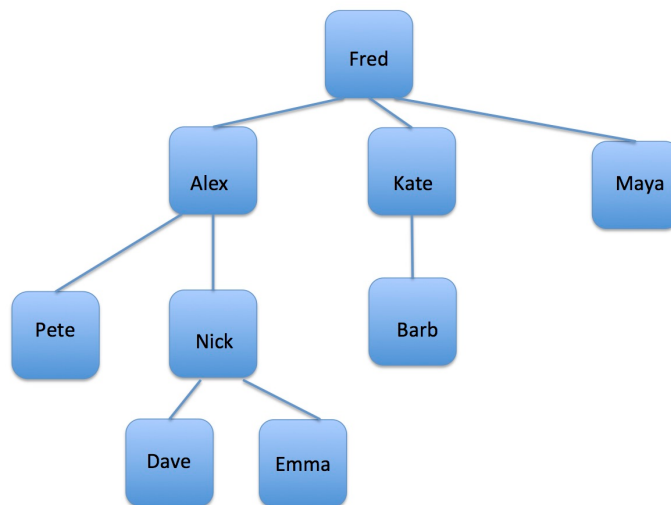- ○ A node can have 1 or 2 children

- ⦿ A node can have 0, 1 or 2 children

- ○ A node can have 0 or 2 children

○ A node must have 2 children

Question 4

Consider the ordered tree shown below. What is the label on the last node in the pre-order traversal of the tree?
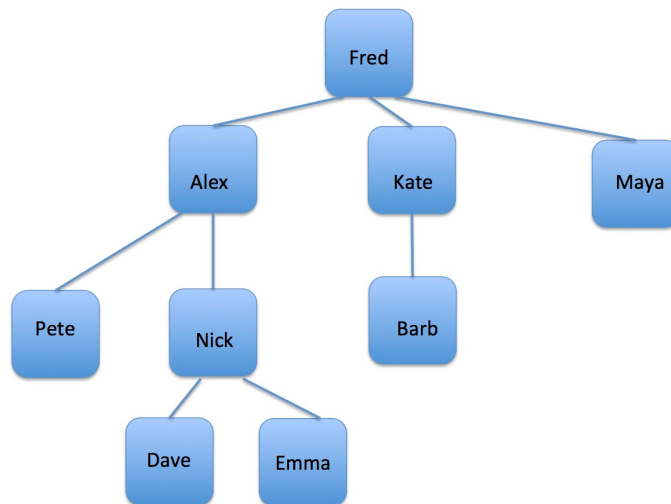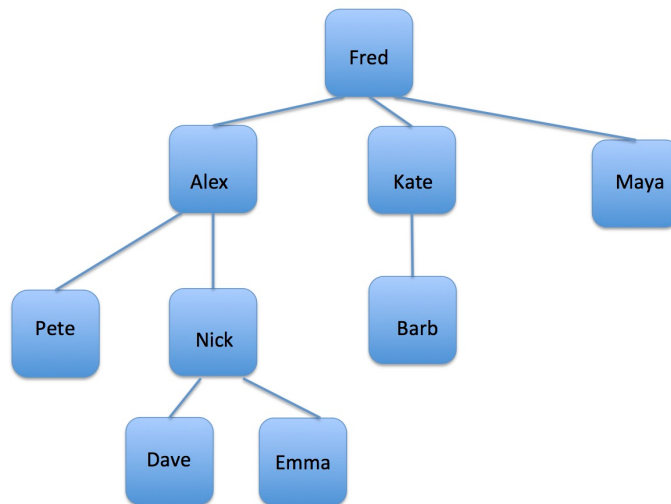

○ Emma

○ Pete

◉ Maya

○ Fred

Consider the ordered tree shown below. What is the label of the second node in a post-order traversal of the tree?

- ◉ Dave
- ○ Kate
- ○ Fred
- ○ Alex

Consider the ordered tree shown below. What is the label of the second

node in a pre-order traversal of the tree?

○ **Alex**

○ Pete

○ Kate

○ Fred

Question 7

Suppose we have a recursive function defined on a binary tree. The function takes as input a node in the tree, calls another function bar on the node that performs some work and then makes recursive calls either to the the left or the right subtree provided they are not empty.

def foo(u):

bar(u)

flip coin

if coin is heads and u.left is not empty **then**

    foo(u.left)

if coin is tails and u.right is not empty **then**

    foo(u.right)

Suppose bar(u) takes O(1) time to execute. What is the time complexity of running foo from the root a tree with n nodes.

- ○ O(log n)

- ◉ O(n)

- ○ O(n²)

- ○ O(1)

Question 8

1 / 1 pts

Suppose we have a recursive function defined on a binary tree. The function takes as input a node in the tree, calls another function bar that performs some work on the node and then makes recursive calls to itself on the left and right subtrees provided they are not empty.

def foo(u):

  bar(u)

  if u.left is not empty **then**

    foo(u.left)

if u.right is not empty **then**

    foo(u.right)

Suppose bar(u) takes O(1) time to execute. What is the time complexity of running foo from the root of a tree with n nodes. Select the tightest bound that holds.

- ⦿ O(n)

- ○ $O(n^2)$

- ○ $O(n^3)$

- ○ O(1)

Incorrect

Question 9

0 / 1 pts

Consider an initially empty tree T. Now perform n consecutive insert operations into T. What is the depth of the root of T after the insertions?

- ⦿ n-1

- ○ log n

- ○ n

- ○ 0

Incorrect

Question 10

0 / 1 pts

Suppose we have a recursive function defined on a tree. The function takes as input a node in the tree, calls another function bar that performs some work on the node and then makes recursive calls to itself on the parent provided it has one.

def foo(u):

    bar(u)

    if u.parent is not empty **then**

        foo(u.parent)

Suppose bar(u) takes $O(depth(u))$ time to execute. What is the time complexity of running foo from the root a tree with n nodes. Select the tightest bound that holds.

- ○ O(1)

- ○ $O(n^2)$

- ◉ O(log n)

- ○ O(n)