**CMP4272- DATA STRUCTURES AND ALGORITHMS**

**LAB SESSION-4: Trees**

**Objectives:**

* Identify computing applications where tree data structures are/can be used.
* Understand the list-of-lists, list-of-children, and parent pointer data structures for trees.
* Use tree data structures to represent data in computer programs.

1. **Background:**

A tree is an abstract model of a hierarchical structure. It consists of nodes with a parent-child relation.

Trees are both Abstract Data Type as well as Data Structure; we have an abstract idea of what a tree is, but there are also data structures for them.

Implementation of trees:

* List of lists
* List of children
* Parent pointer tree

1. **Complete the following exercises before you attempt the mandatory ones provided under Lab submission Exercises in the next section.**

**Exercise-1:** Trees can be represented as outlines, which are nested list structures. In Python, we can write lists with square brackets and nested lists with nested brackets. This gives us a simple data structure for trees, called a ***list-of-lists****.* This is particularly useful when each leaf node holds one data value, and the internal nodes hold no data.

**Write the python list of lists representing the tree structure given below**

A diagram of a network

Description automatically generated

*Answer:*

**Exercise-2:** The following Python code declares the Node class, and constructs a tree using the ***list-of-children*** data structure.

A screenshot of a computer program

Description automatically generated

1. Draw a node/link diagram for the tree.

*Answer:*

1. Find root.children[0]

*Answer:*

1. Draw a second node/link diagram showing the state of the tree after root.children.append(Node('h')) is executed.

*Answer:*

1. **Lab Submission Exercises:**

* **Submitting the solution of the following exercises (3, 4 and 5) is mandatory.**
* **Solutions that comprise of python code, must be well documented. (Include necessary comments)**

**Exercise-3: For the tree given below:**

A diagram of a network

Description automatically generated

1. List all the internal nodes.

P, B, N, Q

1. List all the leaf nodes.

W, V, H, C

1. What is the depth of node H?

2

1. What is the height of the tree?

2

1. List the descendants of node B?

W, N

1. Suppose and edge is drawn connecting node P and H. Discuss whether the resultant structure will still comply with the definition of a tree.

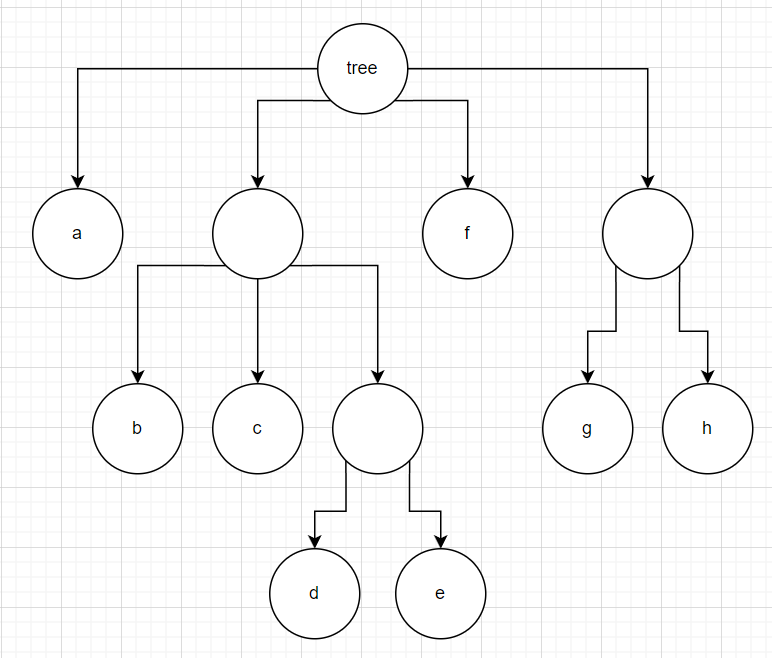
It will no longer be a tree. Each child node can only have 1 parent and not 2.

**Exercise-4:** The following Python code represents a tree as a *list-of-lists.*

A screenshot of a computer code

Description automatically generated

1. Draw a node/link diagram for the tree.



1. Find tree[1], tree[1][2] and tree[1][2][0].

[‘b’, ‘c’, [‘d’, ‘e’]]

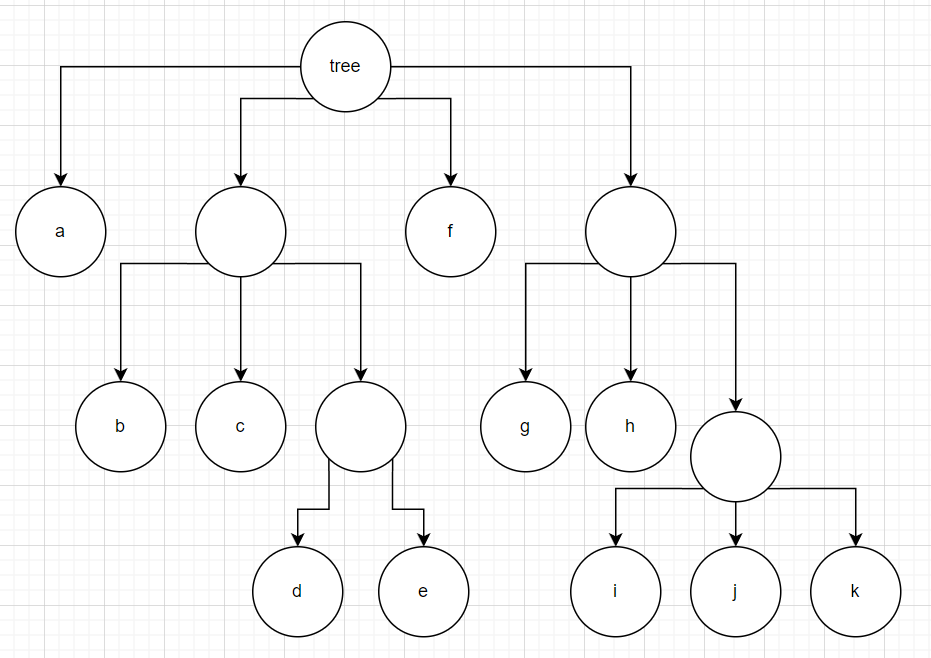
[‘d’, ‘e’]

d

1. Write code to access the node with the value 'h'.

tree[3][1]

1. Draw a second node/link diagram showing the state of the tree after tree[3].append(['i', 'j', 'k']) is executed.



**Exercise-5:** The following Python code declares the Node class, and constructs a tree using the ***list-of-children*** data structure.

A screenshot of a computer program

Description automatically generated

1. Write code to access the node with value 'g'.

*Answer*

1. Find root.children[0].value.

*Answer*:

1. **Moodle Submission:**

You are required to submit your solution in the word document.

Naming Format: **StudetName\_studentID.docx [ or other word formats]**

Example : AliceSmith\_514099.docx

**NOTE**

* It is important to complete the weekly labs in particular labs 2, 3, 4, 5 and 6 because it contains questions that are part of the coursework. (Weightage: 25%).
* Only one of these labs will be chosen randomly for marking, so it is important that you complete and submit each of these labs.
* Ideally, you should submit each lab within one week of that lab session.
* Solutions that comprise of python code, must be well documented. (Include necessary comments)