

COMP2014J: Data Structures and Algorithms 2

Lab Worksheet 1 (Week 3): Trees

David Lillis (david.lillis@ucd.ie)

The goal of this worksheet is to get experience of how to write programs to deal with trees. Today you are working from the point of view of a programmer who *uses* data structures (not the programmer who creates them).

Before you start:

1. Install a Java Development Kit (preferably version 21) and IntelliJ IDEA installed.
2. Download the WS1-Trees.zip file from Brightspace.
3. Follow the instructions in the “How to import projects into IntelliJ IDEA” document (on Brightspace) to import this project. There is also a video explanation.
4. This project contains the following:
 - a. In the “src” folder, you will find three Java files to get you started with the three questions below, named Q1.java, Q2.java and Q3.java.
 - b. In the “doc” folder, you will find the API documentation for all the classes that are available to you. Open the file “index.html” to get the full list. There are some other data structures included in case they are helpful for you.
The ‘dsa.iface’ package contains interfaces for ADTs. The ‘dsa.impl’ package contains implementing classes.

Hint: In the following questions, you will often need to find the node that contains a certain value. You should start by writing a method to do this task, using a tree traversal algorithm. Note that the lecture notes give traversal algorithms for binary trees, but the trees in this worksheet are not binary trees. The pattern is the same, however. For example, in a preorder traversal, a node itself is visited and then all of its children are visited recursively.

Hint: You may also notice other tasks that you may need to perform several times. Consider writing a method for these situations also.

Question 1

For this question, write your code in the Q1.java file. The first line in the main() method loads a tree that stores Character objects.

- a) What is stored at the root of the tree?
- b) What are stored in the children of the root position?
- c) What is the depth of the position that stores “L”?
- d) What is the height of the tree?
- e) List the elements stored in any ancestors of the position that stores G.
- f) List the elements stored in any descendants of the position that stores B.
- g) List the elements that are stored at leaf (external) positions.
- h) Is (N,L) an edge?
- i) List the elements stored in the positions that are in the path from D to N.

Question 2

Write your code in the Q2.java file. The first line in the main() method loads a tree that stores Character objects.

- a) What is the height of the tree?
- b) What is the depth of the position that stores D?
- c) List the elements stored in the children of the position that stores B.
- d) List the elements stored in any siblings of the position that stores D.
- e) List the elements that are stored at external positions.
- f) What is the parent of the position that stores A?
- g) List the ancestors of the position that stores E.
- h) What is the size of the tree?

Advanced (Question 3)

If you are finished Question 1 and 2, and you would like more practice, you can do Question 3 below.

Write your code in the Q3.java file. The first line in the main() method loads a tree that stores String objects.

- a) Which is stored in the root position?
- b) What are stored in the internal positions?
- c) How many descendants does the position that stores "cs016/" have?
- d) How many ancestors does the position that stores "cs016/" have?
- e) What are the siblings of the position that stores "homeworks/"?
- f) Which positions are in the subtree rooted at the position that stores "projects/"?
- g) What is the depth of position that stores "papers/"?
- h) What is the height of the tree?