Tutorial Letter 102/0/2023

COS2611

PROGRAMMING: Data Structures

Year module

Department of COMPUTER SCIENCE

IMPORTANT INFORMATION:

This tutorial letter contains the information pertaining to ASSESSMENT 2

Note: This is a fully online module. It is, therefore, only available on myUnisa.

BARCODE

CONTENTS

ASSESSMENT 2

Dear student, this tutorial letter contains the questions for Assessment 2.

This assessment requires you to plan, develop and implement workable C++ solutions. There are only two questions, question one is based on binary search trees (BST) and question two is based on the minimal spanning tree graph.

Download the folder and files for "Assessment 2 files" from Additional Resources.

1 Question 1 [20]

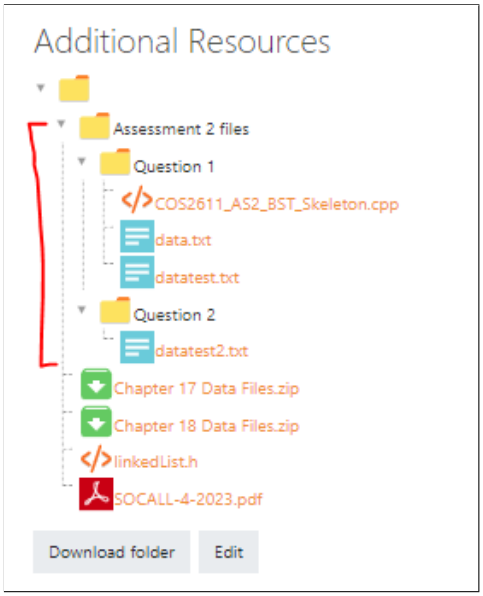
1.1 Pre-knowledge for Question 1

Study chapter 19 in your textbook on Binary Trees. Also, on myUNISA, see the videos and additional learning material that will help you understand the implementation of Binary Trees and specifically Binary Search Trees and Binary Tree Traversal algorithms (inorder traversal, preorder traversal and postorder traversal).

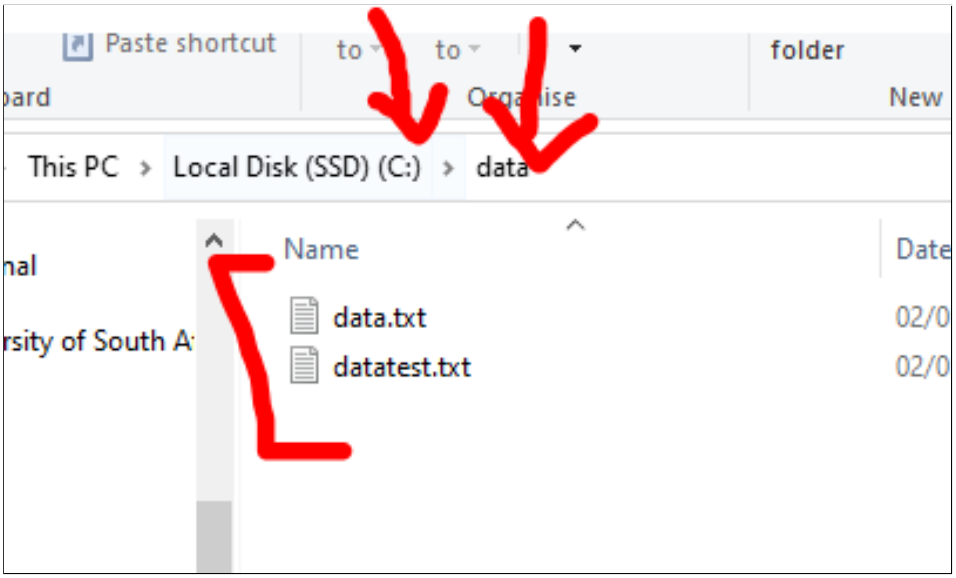
1.2 Scenario for BST

Create a C++ program that will display a binary search tree: inorder traversal, preorder traversal, postorder traversal and height of the tree. The nodes of the binary search tree will be read into the code using a text file.

Download from Additional Resources, the files for Assessment 2, Question 1. The file `COS2611_AS2_BST_Skeleton.cpp` contains a skeleton framework for binary search trees (BST) and two test files.



Create a folder in your C-drive: data. Copy the two files to this folder, `data.txt` and `datatest.txt`. Note variable `thePath` in the skeleton code. You may not change this, as the marker will use the same path and file names when testing your programme.



Rename and save the skeleton code with your details (replace 12345678 with your student number):

`12345678_COS2611_02_Q1.cpp`

Notice that in `main()` are the calling procedures to the classes `BST` and `TreeNode`.

Use the skeleton as a starting point for you. The `insertNode` function is given to you. This

will help you to get started.

You may not change the following classes and procedures:

- `class TreeNode`
- `readFromFileData(string thePath)`
- `main()`

You must use `main()` as is. The marker will use the same code and constructs as in `main()` to call the objects in the classes.

What you should provide: the code for the functions (methods):

(a) `heightOfTree` (5)

(b) `printPreorder` (5)

(c) `printInorder` (5)

(d) `printPostorder` (5)

1.3 Test your code

The data provided in the `data.txt` and `datatest.txt` can be used to test your code.

The data in `data.txt` contains the values for the nodes for the BST. There may be an unlimited number of these values. The program will open this file and read the values and use it to create the nodes for the BST.

The data in `datatest.txt` contains 3 values. Value 1 = first node to search for. Value 2 = second node to search for. Value 3 = node to delete.

When you test your code, the following will be printed for BST VALUES and HEIGHT OF TREE:

```
PRINT BST VALUES

PRINT INORDER TRAVERSAL

11 16 17 19 20 30 32 33 40 45 48 55 88 90 99


PRINT PREORDER TRAVERSAL

90 40 30 20 16 11 19 17 33 32 45 48 55 88 99


PRINT POSTORDER TRAVERSAL

11 17 19 16 20 32 33 30 88 55 48 45 40 99 90


HEIGHT OF TREE

6
```

Save your C++ solution as `12345678_COS2611_02_Q1.cpp` Replace 12345678 with your student number.

1.4 Assessment rubric question 1

The marker will use **a different set of data** to test your code. That is, the marker will compare the results for running your option `4.Print BST values` and option `5.Height of tree`, with the solution.

Marking of the code will be done as follows:

	Mark allocation
Code does not execute or no output	0
Code executes but the result is incorrect	1
Code executes and the result is partially correct	2
Code executes and the result is correct	5

2 Question 2 [10]

2.1 Pre-knowledge for Question 2

Before you continue with this question, study the chapter in your textbook on Graphs. Also, on myUNISA see the videos and additional learning material that will help you in understanding graph definitions, notations, the differences between adjacency matrices and adjacency lists and when to use which. In the graph traversals section, make sure that you understand depth-first traversal, breadth-first traversal and shortest path and that you are able to demonstrate it by hand. **The question in this assessment will focus on the minimal spanning tree as an abstract data type (ADT)**.

2.2 Case study

Company XFibre, is requested by the municipality to install fibre in a new neighbourhood. The company is constrained to bury the fibre cable only along certain paths. Some of the paths might be more expensive. The cost is represented as a weight on the edge. You need to write a C++ program that will determine the minimum spanning tree that will represent the least expensive path for laying the fibre cable.

To solve the above problem for Company XFibre, you may use AI (artificial intelligence) to help. As you are aware AI tools, such as ChatGPT[1], are becoming a valuable tool to help programmers in saving valuable time in the development of code for specific problems. Use an AI tool and generate C++ code that will implement Prim's algorithm using ADT with a time complexity of $O(n^2)$.

After you have downloaded the code from the AI tool, you will need to make some changes.

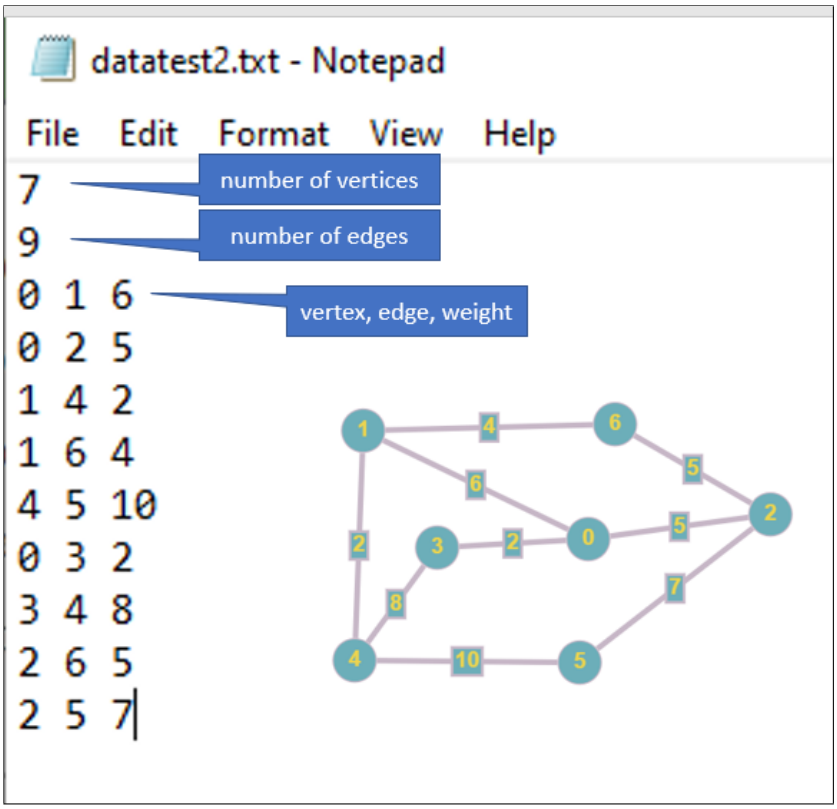
1. Insert as a comment your student name and number.
2. Add as a comment the question that you have asked the AI tool to generate the code for you.
3. Add as comments in your code, the updates and changes that you are making. This is

[1] https://chat.openai.com

important as the marker needs to see your input, changes, and updates to the code to make it work for this case study.

Adapt the code to allow you to read from a text file the test data. To test your code, download from Additional Resources, Assessment 2, Question 2 the `datatest2.txt` file and copy it to your C-drive: data folder (the same folder as in Question 1). The path in your code should refer to this text file. As with Question 1, the marker will use a different set of data to test your code.

The first value in `datatest2.txt`, represents the number of vertices, and the second value is the number of edges. The values thereafter represent combinations of (vertex, edge, and weight).



The output of your code should list the minimum spanning tree paths (see section 2.3 for an example).

2.3 Test your code

The data provided in the `datatest2.txt` can be used to test your code.

When testing your code, the following output can be expected:

```
Minimum Spanning Tree using Prim's Algorithm:
6 - 1
0 - 2
0 - 3
1 - 4
2 - 5
2 - 6
```

Save your C++ solution as `12345678_COS2611_02_Q2.cpp`, and replace 12345678 with your student number.

2.4 Assessment rubric question 2

The marker will use **a different set of data** to test your code. That is, the marker will compare the results for running your code with the solution.

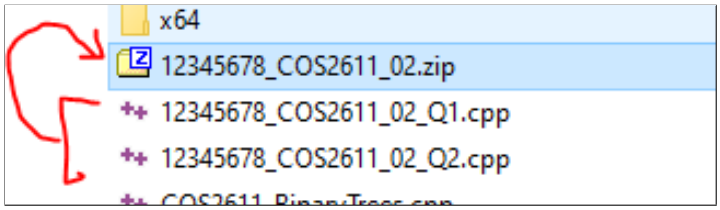
Marking of the code will be done as follows:

	Mark allocation
AI tool was used (see the AI question)	0 - 2 marks
Comments added to code to indicate the changes to the code	0 - 3 marks
Code does not execute or no output	0
Code executes but the result is incorrect	1
Code executes and the result is partially correct	2
Code executes and the result is correct	5

[TOTAL: 30 MARKS]

3 How to submit

When you have completed the code for both questions (cpp) [`12345678_COS2611_02_Q1.cpp` and `12345678_COS2611_02_Q2.cpp`, combine the two text files in one ZIP folder. Label the ZIP folder as `12345678_COS2611_02.ZIP`. [Note: should you have decided to include external classes or templates, then include them in the ZIP folder.] See the example below (replace 12345678 with your student number 🙂).



Upload the ZIP folder to myUNISA - Assessment 2.

Looking forward to your solutions!

Lecturer(s) for COS2611

DEPARTMENT OF COMPUTER SCIENCE