Beijing-Dublin International College

SEMESTER 2 EXAMINATION - 2017/2018

School of Computer Science

COMP2003J Data Structures and Algorithms 2

Prof. Pádraig Cunningham
Dr. David Lillis *

Time Allowed: 120 minutes

Instructions for Candidates

Answer any 2 questions. All questions carry equal marks.

BJUT Student ID: _____ **UCD Student ID:** _____

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge： _____ **(Signature)**

Instructions for Invigilators

No special instructions.

Question 1:

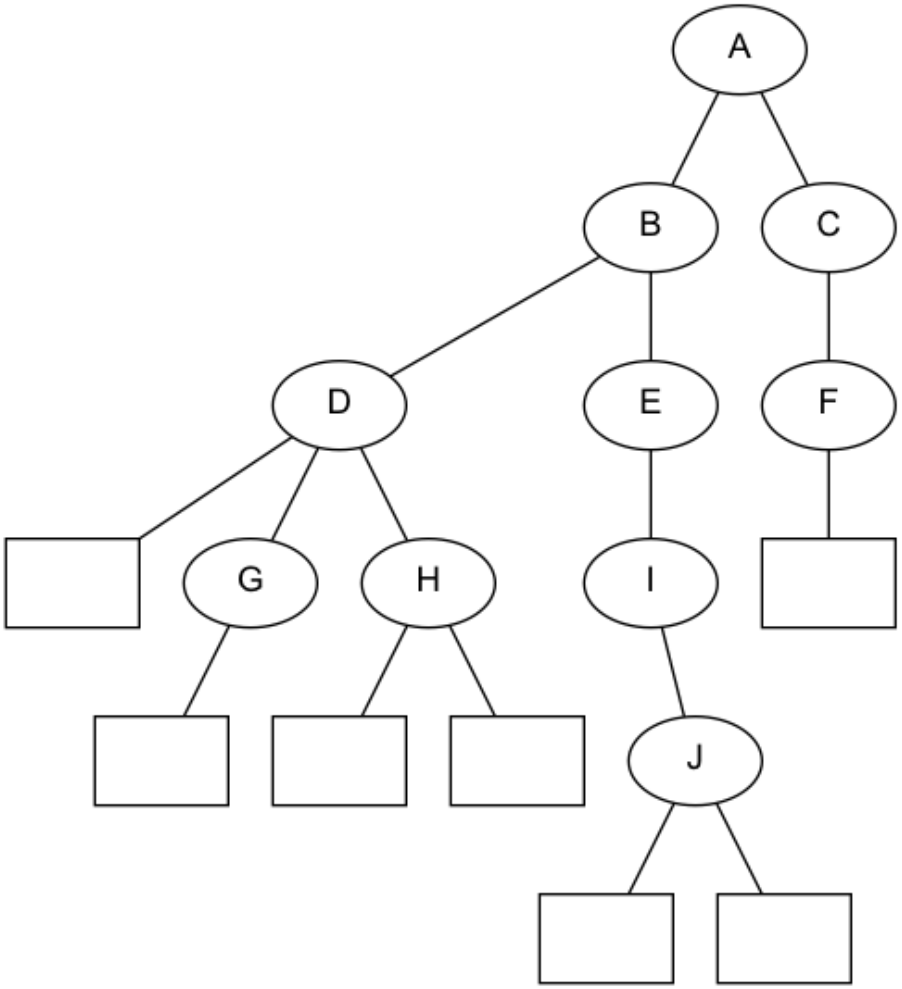

Figure 1

(a) Study the tree in Figure 1 and answer the questions that follow.

(i) List the siblings of G.

(ii) What is the depth of D?

(iii) What is the degree of node J?

(iv) What is the height of the tree?

(v) List the descendants of D.

(vi) Is (D,H) an edge? Explain your answer.

(vii) Is (D,B,A,C) a path? Explain your answer.

(viii) List the nodes that are in the subtree that is rooted at E.

[8 marks]

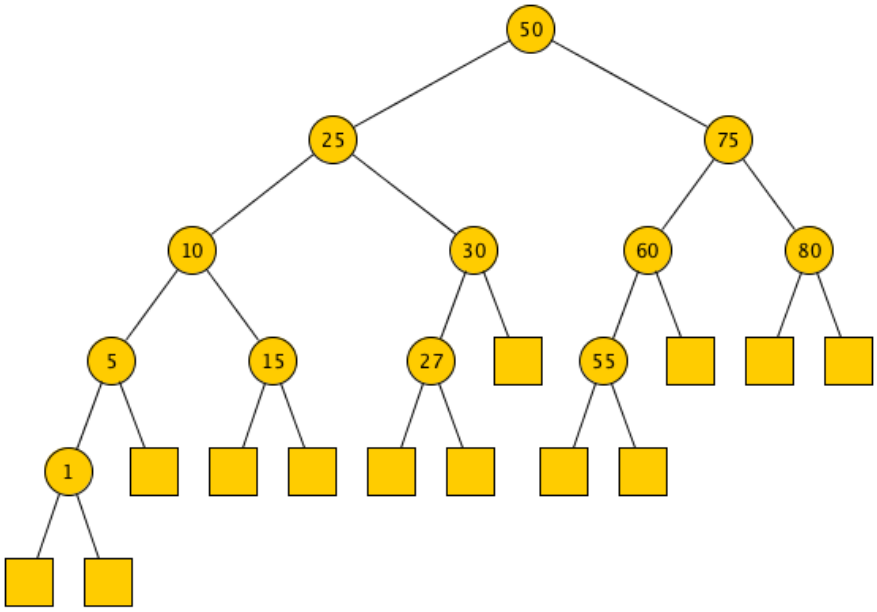

Figure 2

(b) Assume that the tree in Figure 2 is an **AVL Tree**. Draw the state of the tree after performing the following operations. In your answer, you should show the tree's state after each step and mention any restructuring that is required.

(i) Remove 80

(ii) Remove 15

(iii) Insert 78

(iv) Insert 28

(iv) Remove 55

[12 marks]

(c) Describe how a value is inserted into a Splay Tree. Include in your answer a discussion of how the tree is restructured.

[10 marks]

(d) Assume that the tree in Figure 2 is a Splay Tree. If the next operation is to insert 4, draw trees to show all restructuring operations that will happen and state what restructuring operations these are.

[6 marks]

(e) Briefly explain the differences between a *tree*, a *binary tree*, a *proper binary tree* and a *binary search tree*.

[4 marks]

(f) With regard to analysing the performance of an *AVL Tree*, discuss each of the following aspects:

(i) Space requirements.

(ii) Time complexity of the insert() and remove() functions.

[6 marks]

(g) In a proper binary tree implementation, what does it mean to "expand an external node"?

[4 marks]

[Total 50 marks]

Question 2:

(a) Give definitions of the following graph concepts: *parallel edges*, *directed graph*, *endpoints of an edge*, *path*, *self-loop*, *spanning forest*.

[10 marks]

(b) One implementation of a *priority queue* ADT is to use a *heap*.

(i) Describe the priority queue ADT and the core operations it has.

[5 marks]

(ii) Draw the heap that would be obtained after each of the following operations (your answer should show 10 heap diagrams).

insert(20), insert (12), insert(6), insert(16), insert(11),

insert(18), remove(), remove(), insert(1), remove()

[10 marks]

(iii) In a heap, what is the time complexity of a *downheap* operation? Explain your answer.

[3 marks]

(c) An array-based list is an appropriate way to implement a *Complete Binary Tree*. Explain in detail the advantages of using this type of implementation type compared to using a linked structure.

[6 marks]

(d) This part relates to *Huffman Encoding*.

(i) What is Huffman Encoding used for?

[2 marks]

(ii) Using the string “huffman trees are great” as an example, describe in detail how a Huffman Tree is created.

[11 marks]

(iii) Using the tree you created in part (ii), what is the code that represents the string “tears”?

[3 marks]

[Total 50 marks]

Question 3:

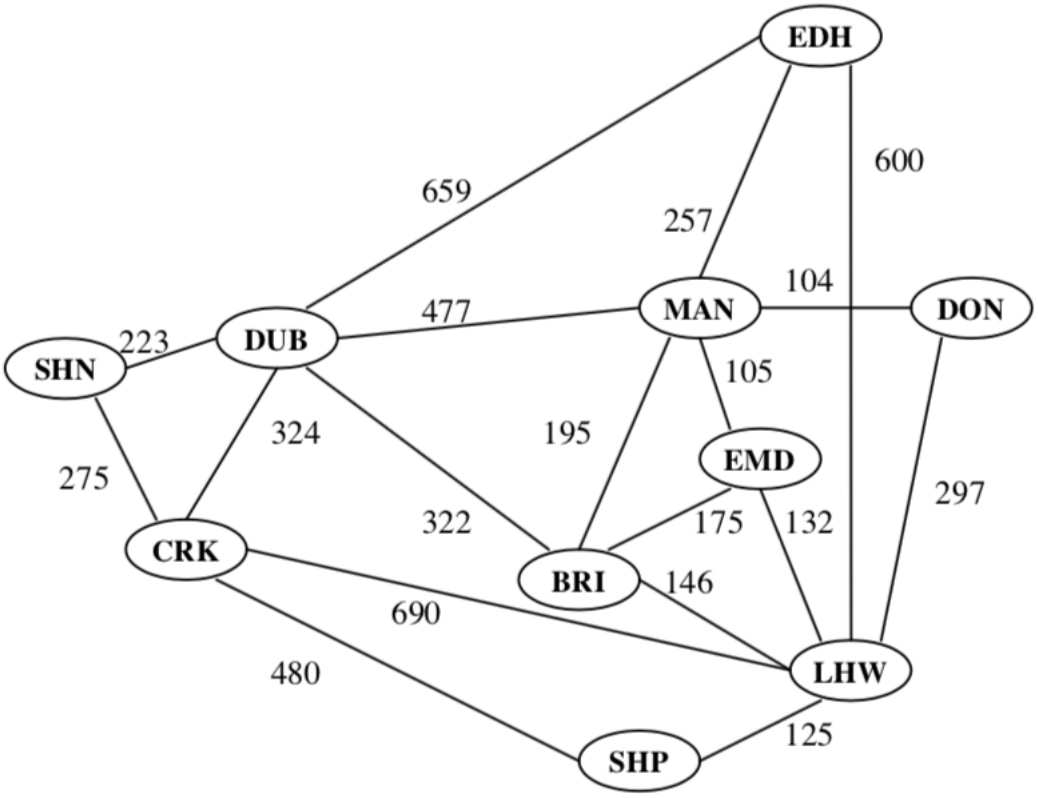

Figure 3

(a) This part relates to *Minimum Spanning Trees*.

(i) What is a Minimum Spanning Tree? How is this different to a *shortest distance tree* such as the one Dijkstra's Algorithm produces?

[4 marks]

(ii) Using the graph in Figure 3 as an example, show how Kruskal's algorithm can be used to compute a Minimum Spanning Tree. In your answer, you must explain each step that you take.

[10 marks]

(b) This part relates to graph traversals.

(i) Using the graph in Figure 3 as an example, show a *Breadth First Search* graph traversal works. In your answer, you must explain each step that you take.

[10 marks]

(ii) Breadth First Search runs in O(*n+m*) time (where *n* is the number of vertices in a graph, and *m* is the number of edges). Explain why this is the case.

[6 marks]

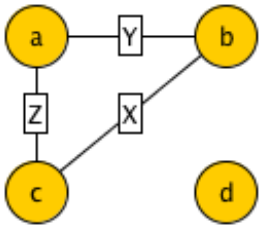

Figure 4

(c) Draw a diagram to show how the graph in Figure 4 can be represented using an *adjacency list* structure. In your answer, describe each object type and data structure used to explain its purpose and what data it stores.

[12 marks]

(d) When implementing a graph, using an *adjacency list* structure instead of an *edge list* structure results in better time complexity for the *incidentEdges(v)*, *areAdjacent(v,w)* and *removeVertex(v)* operations. Explain why this is the case.

[8 marks]

[Total 50 marks]