
**DUBLIN INSTITUTE OF TECHNOLOGY**

# BSc. (Honours) Degree in Computer Science

**Year 2**

## SUMMER EXAMINATIONS 2014/2015

## ALGORITHMS & DATA STRUCTURES [CMPU2001]

MR. RICHARD LAWLOR
DR. DEIRDRE LILLIS
MR. PAUL COLLINS

FRIDAY 15TH MAY 4.00 P.M. – 6.00 P.M.

TWO HOURS

ATTEMPT THREE OUT OF FOUR QUESTIONS
ALL QUESTIONS CARRY EQUAL MARKS

ONE COMPLEMENTARY MARK FOR PAPER

1. **(a)** Provide a Java or C# class based on a linked list implementation for the abstract data type *Queue* showing data structure and method declaration, but not method body or definition. You need only concern yourself with the two significant queue methods and the data structure. You can assume the queue stores *int* values.

(6 marks)

**(b)** Give the implementation of the method *enQueue()*. Explain why you might use or not use a tail reference in your implementation. What is the complexity of enQueue() with and without the use of a tail pointer?

(9 marks)

**(c)** Use diagrams to show how an integer can be inserted into a sorted linked list. Write code for this insert method.

(12 marks)

**(d)** Write down a simple equation for a Stack in terms of *pop()* and *push()* that expresses the last-in-first-out (LIFO) behaviour of the Stack.

(6 marks)

2. **(a)** Explain the terms:

i) priority queue
ii) complete binary tree
iii) heap
iv) heap condition

(5 marks)

**(b)** Draw the following heap array as a two-dimensional binary tree data structure:

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a[k] | | 13 | 10 | 8 | 6 | 9 | 5 | 1 | | | | |

Also, assuming another array hPos[] is used to store the position of each key in the heap, show the contents of hPos[] for this heap.

(6 marks)

**(c)** Write in pseudocode the algorithms for the *siftUp()* and *insert()* operations on a heap and show how hPos[] would be updated in the *siftUp()* method if it was to be included in the heap code. Also write down the complexity of *siftUp()*.

(9 marks)

**(d)** By using tree and array diagrams, illustrate the effect of inserting a node whose key is **12** into the heap in the table of part (b). You can ignore effects on hPos[].

(6 marks)

**(e)** Given the following array, describe with the aid of text and tree diagrams how it might be converted into a heap.

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| b[k] | | 2 | 9 | 18 | 6 | 15 | 7 | 3 | 14 |

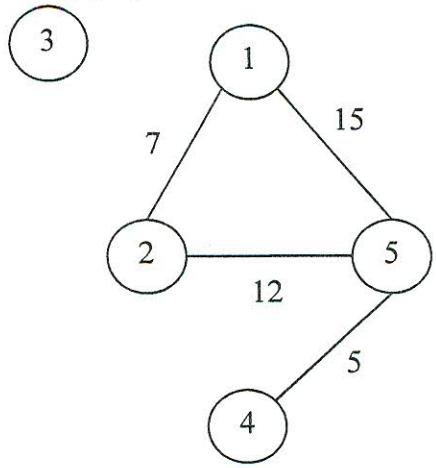
(7 marks)

**3. (a)** An industrial process requires the drilling of holes at predefined positions in a metal sheet. The holes are bored by a robot arm which moves the drill from position to position. When the drilling is complete, the robot arm returns to its initial position. The same set of holes are to be drilled in many metal sheets, so minimising the time spent in movement of the robot arm is essential.

How can this problem be related to graph theory and what kind of graph if any does it correspond to?

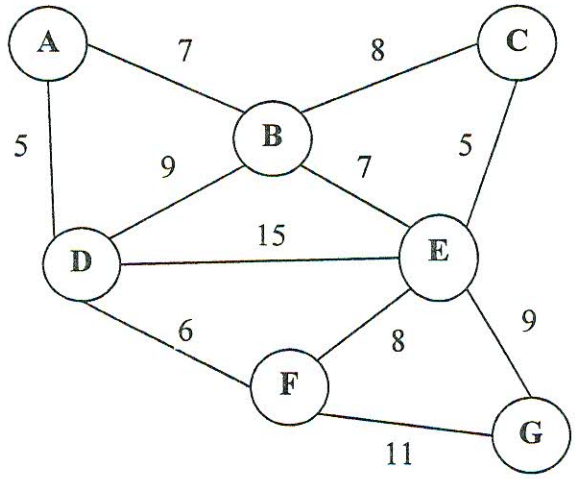
(5 marks)

**(b)** Suggest two possible representations for a weighted graph when it is to be stored in RAM, and mention if one representation is preferable in some circumstances and why. Then illustrate how the following graph would be stored in both representations.




(6 marks)

**(c)** Illustrate in detail how Prim's algorithm computes a MST for the graph below showing the heap contents and the state of the arrays distance[] and parent[] at each stage. It is not necessary to show the heap tree structure at each stage, just its contents.




(10 marks)

**(d)** Write Prim's algorithm in pseudocode for an adjacency lists representation of a sparse graph and discuss its complexity. What would be its complexity for a dense graph?

(12 marks)

**4. (a)** Show how binary search works when searching for ***17*** in the following array:

| 1 | 5 | 6 | 9 | 10 | 12 | 14 | 17 | 21 |
|---|---|---|---|---|---|---|---|---|

(5 marks)

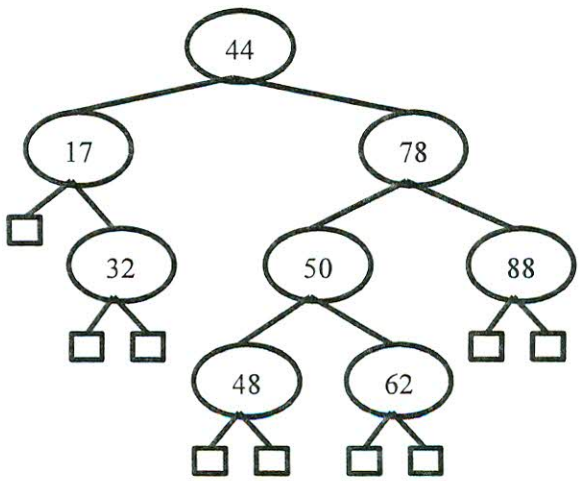
**(b)** What is a binary search tree (BST)? Mention any specific advantage or possible disadvantage. What is the complexity of searching a BST?

(6 marks)

**(c)** Write in pseudocode the algorithm for searching a BST.

(6 marks)

**(d)** Given the following binary search tree, show how it would be modified by inserting 54.




(5 marks)

**(e)** What is an AVL-tree? Include in your answer the idea of a *rotation*.
Show how the tree that results from inserting 54 in part (d) would be rebalanced if it were an AVL-tree.

(11 marks)