Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Algorithms and Imperative Programming

Date: Tuesday 4th June 2013

Time: 14:00 - 16:00

Please answer THREE Questions from the FOUR Questions provided

Use a SEPARATE answerbook for EACH Section

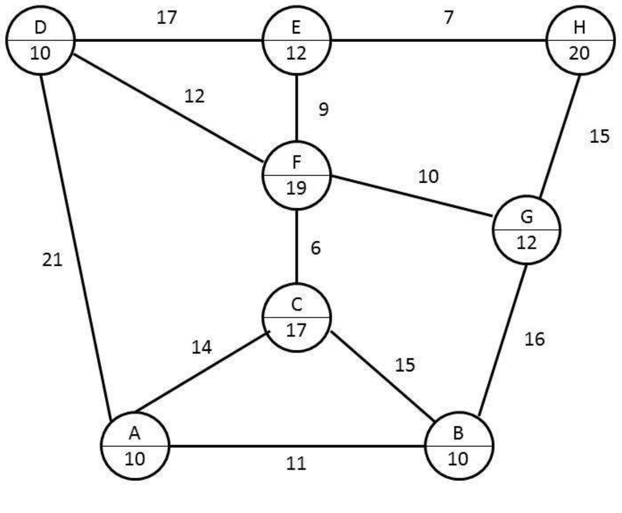
This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text.

Section A

1. a) Give a pseudo-code description of Dijkstra's algorithm for finding the shortest path between a given node and all other nodes in a weighted undirected graph. Assume in your answer that the functions for handling a priority queue that is required by Dijkstra's algorithm are defined. Explain how Dijkstra's algorithm can be generalised to find the shortest paths between all pairs of nodes. Describe a modification of this approach which cuts the execution time by a half. (5 marks)

   b) Consider the following weighted undirected graph:



i) Show the progression of Dijkstra's algorithm, step by step, and draw the content of a priority queue at each step on the example of finding the shortest path between the nodes A and H. (For this part ignore the numbers at the graph nodes). (8 marks)

ii) A passenger needs to go from the town A to the town H. They can cover up to 14 distance units per day and can only spend a night in a town (i.e. at a node). The cost of an overnight stay in each town is given in each node of the graph. Suggest a heuristic algorithm for finding the optimal path between the towns A and H. In your answer consider two optimality criteria: the shortest path and the cheapest path. (7 marks)

2. a) Explain what the *fractional knapsack upper bound* is, how it is calculated, and (briefly) how it is used in branch and bound algorithms for 0/1 knapsack problems. (You do not need to define the 0/1 knapsack problem). (6 marks)

b) Consider the following 0/1 Knapsack Problem instance.

	items					
index	1	2	3	4	5	6
value	3003	591	174	274	83	47
weight	1001	197	87	137	83	47
value:weight ratio	3	3	2	2	1	1

Knapsack capacity $C = 1185$

i) If items are presented to the greedy algorithm in the given ordering (descending by value:weight), show that it does not find the optimal solution. (You need to give an explicit argument or show step-by-step what prevents an optimal solution from being found).

(3 marks)

ii) Propose a variation to the greedy algorithm that would find the optimal solution to this instance, given this initial ordering of items, and show it works.

(4 marks)

iii) Comment on the efficiency of applying dynamic programming to this problem instance (e.g. compared with greedy or branch-and-bound). Explain your answer with regard to the particular nature of the problem instance.

(3 marks)

c) Explain how Bellman's Principle of Optimality,

"A problem is said to satisfy the Principle of Optimality if the subsolutions of an optimal solution of the problem are themselves optimal solutions for their subproblems"

is exploited in the tabular dynamic programming algorithm for solving 0/1 Knapsack. For full marks you should indicate what the subsolutions and subproblems are in the 0/1 Knapsack problem, why we know they are optimal, and how the Bellman Principle applies to allow building up of solutions to larger subproblems. (You do not need to give a full description of the tabular DP algorithm, but you should refer to the decisions and data structure it uses).

(4 marks)

Section B

3. Consider the following sequence of the keys:

```
25,12,39,17,14,43,53
```

We wish to insert these numbers in the given order into:

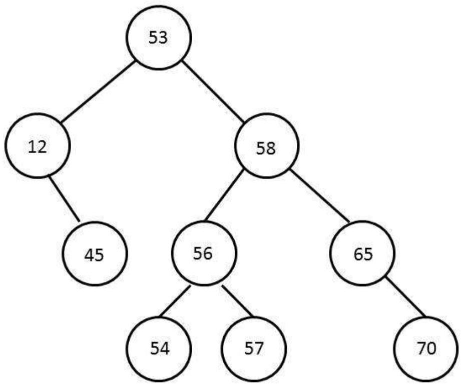
- a hash table of size 11 (with the elements denoted from 0 to 10);
- an AVL tree.

a) Show the resulting hash table if the hash function $h(x) = (2x+3) \bmod 11$ is used and the collisions are handled by linear probing. (5 marks)

b) Show the resulting AVL tree. (5 marks)

c) How many comparisons are required to find an element in a hash table (with linear probing) of length n which contains m elements ($m \leq n$). Discuss the best and the worst case scenario. Discuss the complexity (in terms of the number of comparisons) for the same operation in the case of an AVL tree. (5 marks)

d) Show the steps of removing the element with the key 45 from the AVL tree given below. (5 marks)



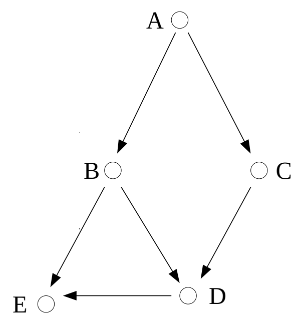
4. Graphs and graph algorithms.

a) Consider the following two representations of finite directed graphs:

- Adjacency lists, and
- Adjacency matrices.

i) Explain briefly what these representations are. (2 marks)

ii) Show how the following graph can be presented using the two representations. (2 marks)



b) Consider the following two traversal methods for finite directed graphs:

- Depth-First Search (DFS), and
- Breadth-First Search (BFS).

i) In each case, explain, briefly but clearly, what the traversal method is and how it applies to finite directed graphs. (3 marks)

ii) Give one example of a DFS traversal, and one example of a BFS traversal, of the example graph in Part (a)(ii) above. Your answers should be lists of nodes, starting from the node labelled *A*. (2 marks)

c) Give an algorithm for performing DFS traversals of finite directed graphs which numbers the nodes in the order they are first encountered. You may either present a program or express the algorithm in pseudocode.

What is the worst-case time complexity of your algorithm in terms of the number of nodes N and edges E? Explain your answer. (5 marks)

d) For finite *undirected* graphs, a *2-colouring* of a graph is an allocation of one of two colours (say, red and green) to each of the nodes of the graph, so that, whenever two nodes are linked by an edge, they are allocated *different* colours.

Some graphs have a 2-colouring and some don't. Give an example of an undirected graph which does and one which doesn't. Explain your answer. (2 marks)

How can a DFS algorithm be used to determine whether or not a finite undirected graph has a 2-colouring? Your answer should be a clear explanation of the principle involved and the steps of your DFS algorithm (you need not give a program or pseudocode). (4 marks)

END OF EXAMINATION