Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Algorithms and Imperative Programming

Date: Thursday 31st May 2018

Time: 09:45 - 11:45

Please answer all Questions.

Use a SEPARATE answerbook for EACH Question

This is a CLOSED book examination

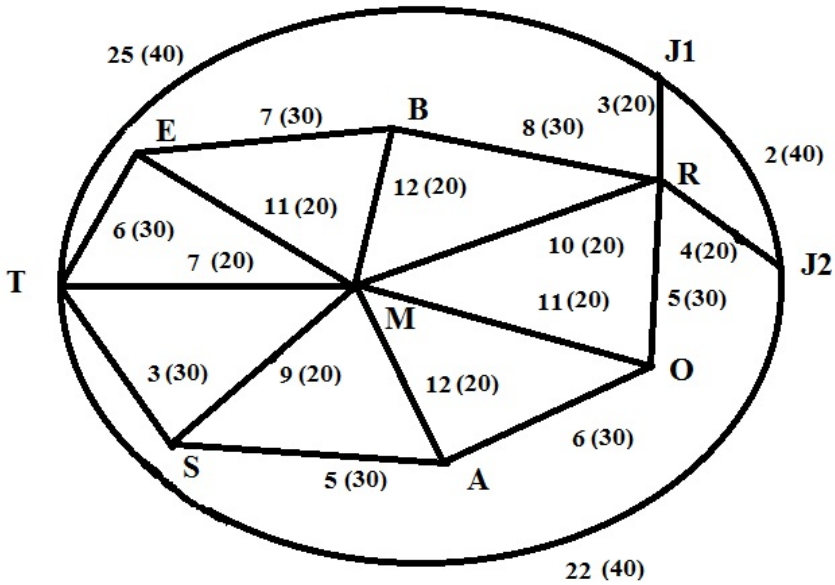
The use of electronic calculators is permitted provided they
are not programmable and do not store text

[PTO]

1. Graph algorithms.

   a) Write a pseudocode description of Dijkstra's algorithm for finding the shortest path between a given node and all other nodes in a connected, undirected, weighted graph. You may use in your answer a priority queue and the standard functions that operate on it. (5 marks)

   b) The following diagram shows main roads connecting towns in Greater Manchester:



   The values associated with the edges represent the distances in miles between the places and, in brackets, the average speed (in miles per hour) on these roads during the morning rush hour. A person lives in Rochdale (R) and works in Trafford (T). By applying Dijkstra's algorithm, find the path which will take *the shortest amount of time*, in the rush hour, between the towns R and T. Explain how you compute this path.

   (10 marks)

   c) Give the algorithmic complexity of Dijkstra's algorithm when applied to the problem of finding the shortest path between a given node and all other nodes in a connected, undirected, weighted graph with n nodes and e edges. In your answer, discuss the two different implementations of the priority queue: as a list and as a heap. Define sparse and dense graphs by expressing the relationship between n and e in such graphs. Based on this, explain the complexity of Dijkstra's algorithm with the heap implementation of a priority queue for the cases of sparse and dense graphs. (5 marks)

2. A combinatorial problem and dynamic programming.

   We are given a list of positive integers $P = [p_1, \ldots, p_n]$ of length $n \geq 0$, together with a positive integer s. We wish to determine whether a subset of the integers in P sums exactly to s. More formally, we wish to output *either* a list $B = [b_1, \ldots, b_n]$, with each $b_i \in \{0, 1\}$, such that

   $$\sum_{i=1}^{n} b_i \cdot p_i = s$$

   or the word `impossible`, if no such list B exists.

   a) Using pseudocode, write an algorithm which solves this problem by brute-force search. You must explain clearly how your algorithm works, including the role of all the variable used and, where possible, their meanings. In your explanation, you may find it useful to refer to the pseudocode using line-numbers. (8 marks)

   b) Give an upper complexity-bound for the algorithm given in Part 2a as a function of n. (2 marks)

   c) Write an algorithm in pseudocode which solves this problem using *dynamic programming*. You must explain clearly how your algorithm works, including the role of all the variables and, where possible, their meanings. In your explanation, you may find it useful to refer to the pseudocode using line-numbers. (8 marks)

   d) Give an upper complexity-bound for the algorithm given in Part 2c as a function of s and n. (2 marks)

3. Hashing techniques.

   Consider a hash table used for storing integer keys, with the following properties:

   - The initial table size is $n = 5$.
   - Quadratic probing is used for collision resolution.
   - The hash function is given by

     $$h(x) = (x + j^2) \mod n, \quad j = 0, 1, 2, \ldots$$

     where x is the integer key and n is the current size of the hash table.
   - We consider that an insertion fails if more than a half of the buckets are attempted (i.e. when the number of failed attempts is $\geq \lceil \frac{n}{2} \rceil$).
   - The table is enlarged by doubling its size when either: i) the load factor is > 0.6; or ii) when an insertion fails.

   (a) Present the process, step by step, for the insertion of the following keys, in the given order, into the hash table:

   6,16,36,56,76

   (12 marks)

   (b) Highlight any key that failed to be inserted during the insertion process in (a).

   (2 marks)

   (c) What is the size of the final hash table and what is its final load factor?

   (2 marks)

   (d) What can be concluded from the pattern of entries in the hash table when quadratic probing is used? (4 marks)

END OF EXAMINATION