

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLAISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

CS2516 Algorithms and Data Structures II

SAMPLE EXAM PAPER

1.5 Hours

The use of electronic calculators is permitted

Answer both questions

Total Marks: **80**

(~1 minute per mark)

**PLEASE DO NOT TURN THIS PAGE UNTIL
INSTRUCTED TO DO SO**

**PLEASE
ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER**

1. (40 marks)

In all parts of the question n refers to the length of the input sequence.

- (i) Write out the algorithm for Insertionsort applied to an array-based list of integers, using clear pseudocode, or python code operating on a python list. What is the worst-case time complexity of Insertionsort, in terms of the number of comparisons as a function of n ?
(5 marks)
- (ii) Write out a recursive algorithm for Quicksort applied to an array-based list of integers, using clear pseudocode or python code. What is the worst-case time complexity of Quicksort, in terms of the number of comparisons? What is the expected time complexity of Quicksort?
(10 marks)
- (iii) Show the main steps in a trace of QuickSort on the following sequence, showing the subproblems tackled in each recursive call:
[6 7 0 8 2 4 9 1 5 3]
(5 marks)
- (iv) Explain how the idea of Quicksort can be adapted to select what would be the k th ordered element of an input list, in average time $O(n)$.
(5 marks)
- (v) Suppose we have a list of A s and B s, and we want to sort the list so that all the A s come first. Write an efficient algorithm to do this. Marks will be based on correctness, clarity and efficiency.
(5 marks)
- (vi) Suppose now that our list consists of 0 s, 1 s and 2 s, and we want to sort the list so that all the 0 s come first, then the 1 s and finally the 2 s. Write an efficient algorithm to do this. Marks will be based on correctness, clarity and efficiency.
(10 marks)

2.

(40 marks)

- (i) The *Graph* ADT offers methods for querying a graph and for modifying a graph. It relies on two other ADTs: the *Vertex* and the *Edge*. What methods are offered by these two ADTs? Give a clear English language comment for each method.

(5 marks)

- (ii) Describe the *Adjacency Map* implementation of the *Graph* ADT and discuss the time and space complexity of the main Graph methods for this implementation.

(10 marks)

- (iii) What are the additional methods offered by the *Directed Graph* ADT, and the additional methods needed for the *Edge* ADT to support them?

(6 marks)

- (iv) Define a *Directed Acyclic Graph (DAG)*, and define a *Topological Sort* for a DAG.

(5 marks)

- (v) Give clear pseudocode for an algorithm which creates a topological sort for a directed graph if one exists.

(5marks)

- (vi) A directed graph can be used to model a task scheduling problem. Each task is represented by a vertex in a graph, and if task A must be completed before task B begins, then there is a directed edge from A to B. If two tasks are unrelated, and can be executed in any order, then there is no edge between them in the graph. A scheduling model is consistent if it is possible to sequence the tasks so that all priority constraints are satisfied. Suppose we have two different models of a scheduling problem: a single set of tasks, and two different directed graphs created over the task. Give clear pseudocode of an algorithm to determine whether or not the two scheduling models are compatible – that is, is there a sequence of the tasks that satisfies both sets of constraints. Explain why your algorithm guarantees a correct answer to the problem. State the complexity of your algorithm, assuming an Adjacency map (or adjacency list) implementation of the graph.

(10 marks)

Total: 80 marks

END OF PAPER