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

COLAISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

2016/17

Semester 2 -- Summer 2017

CS2516 Algorithms and Data Structures II

Dr Helen Purchase
Professor Cormac Sreenan
Professor Ken Brown

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 selection sort 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 selection sort, 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, and clearly explain the role of the *pivot*. What is the worst-case time complexity of Quicksort, in terms of the number of comparisons?

(10 marks)

(iii) Show the main steps in a trace of QuickSort on the following sequence, showing each list that is passed as input to a recursive call, and each list that results from the recursive call. Choose the first element of each sub-list as the pivot.

[12 15 29 6 27 10 32 21 24]

(5 marks)

(iv) What is the expected complexity of Quicksort? Explain the role of pivot selection in achieving this.

(3marks)

(v) What is the best possible worst-case complexity for any comparison-based sorting algorithm? How could a sorting algorithm avoid comparing elements? How could we extend this idea to sorting lists of integers, where we know the range of the integers?

(7 marks)

(vi) Suppose we have two unsorted python lists of integers A and B, each of length n. We want to determine if, for some given input integer z, there are elements x of A and y of B such that $x + y = z$. Given an efficient algorithm for correctly answering whether x and y exist. What is the worst-case complexity of your algorithm? Justify your answer. Answers will be judged on algorithm clarity, efficiency, and the complexity justification.

(10 marks)

2. **(40 marks)**

(i) Describe the *Adjacency Map* implementation of the *Graph* ADT. If n is the number of vertices, and m is the number of edges, explain the time complexity of the following Graph methods for this implementation.

get_edges(x) – return a list of all edges incident on x
get_edge(x,y) – return the edge between vertices x and y
add_vertex(x) – add vertex x to the graph
add_edge(x,y) –add an edge between vertices x and y
remove_vertex(x) – remove vertex x and all edges incident on it.

(12 marks)

(ii) Give clear pseudocode (or Python code) for an implementation of Depth-first search traversal of an arbitrary undirected graph

(4 marks)

(iii) Define a *Directed Acyclic Graph (DAG)*,

(3 marks)

(iv) Prove that any *DAG* must have at least one vertex with in-degree = 0

(3 marks)

(v) Define a *Topological Sort* for a DAG

(3 marks)

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

(5marks)

(vii) An *articulation* vertex in a graph is a vertex such that if it was removed, the graph would no longer be connected. Give clear pseudocode for an algorithm which determines whether or not a connected undirected graph has an articulation vertex. What is the worst-case complexity of your algorithm? Justify your answer. Answers will be judged on algorithm clarity, efficiency, and the complexity justification.

(10 marks)

Total: 80 marks

END OF PAPER