PAPER CODE NO. **COMP523**	EXAMINER: Aris Filos-Ratsikas DEPARTMENT: Computer Science	Tel. No. 0745514272

FIRST SEMESTER EXAMINATIONS 2019/20
Advanced Algorithmic Techniques

TIME ALLOWED : TWO AND A HALF Hours

INSTRUCTIONS TO CANDIDATES

NAME OF CANDIDATE .... SEAT NO ....

USUAL SIGNATURE ....

READ THE FOLLOWING CAREFULLY:

1. Answer **FOUR** questions. Each question is worth 25 marks. If more questions are answered, the 4 questions with the highest marks will be considered.
   For complexity analysis use asymptotic notation, always provide a justification for the complexity formulas and for the correctness of your arguments.
2. The exam mark is based on the overall number of correctly answered questions. The more questions you answer correctly the higher your mark, incorrectly answered questions do not count against you.
3. Enter your name and examination number IN PENCIL on the computer answer sheet according to the instructions on that sheet.
4. When you have completed this exam paper, read the instructions on the computer answer sheet carefully and transfer your answers from the exam paper. Use a HB pencil to mark the computer answer sheet and if you change your mind be sure to erase the mark you have made. You may then mark the alternative answer.
5. At the end of the examination, be absolutely sure to hand in BOTH this exam paper AND the computer answer sheet.
6. Calculators are permitted.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

UNIVERSITY OF LIVERPOOL

1. A. Consider a directed graph with nodes $\{1, 2, 3, 4, 5, 6\}$ and edges:
$(3, 2), (5, 3), (4, 6), (3, 1), (1, 4), (4, 1), (5, 6), (1, 3), (4, 5)$

(A1) Give the representation of this graph as:

(i) an adjacency list [4 marks]
(ii) an adjacency matrix. [4 marks]

(A2) Find a BFS directed spanning tree rooted at node 4 and represent it as:

(iii) an adjacency list; [4 marks]
(iv) an adjacency matrix. [4 marks]

B. (B1) Describe a deterministic algorithm with running time $O(m + n)$ for checking whether a given directed graph $G = (V, E)$ with n nodes and m edges, represented as an adjacency list, is a Directed Acyclic Graph (DAG) or not. [6 marks]

(B2) Give an argument for the correctness of your solution and the running time, and analyse the auxiliary memory used by the algorithm. [3 marks]

2. A. (A1) Define the interval scheduling problem for a given set of m intervals on the line. [2 marks]

(A2) Discuss the notions of approximation algorithms and approximation ratio, for both minimisation and maximisation problems. [4 marks]

(A3) Consider the following greedy algorithms for choosing which interval to schedule next: EST (earliest starting time), SI (smallest interval) and MO (minimum number of overlapping intervals). Provide instances of the problem showing that the approximation ratios of these algorithms are at least $m - 1$, 2 and $4/3$ respectively. [4 marks]

B. (B1) Define the maximum independent set problem on a graph G. [2 marks]

(B2) Provide a polynomial-time reduction from the interval scheduling problem to the maximum independent set problem. Discuss the implications of this reduction with respect to the NP-hardness or the polynomial-time solvability of the problems. [5 marks]

(B3) Consider the interval colouring problem, in which we are given m intervals and our goal is to colour them using as few colours as possible, under the restriction that overlapping intervals should not receive the same colour. Design a greedy algorithm that solves this problem optimally and argue about its correctness. [8 marks]

3. **A.** (A1) Describe the QUICKSORT algorithm for sorting a sequence of n comparable elements, stored in an array A. [2 marks]

   (A2) Consider the randomised version of QuickSort, where the pivot element is chosen uniformly at random among the n elements. Prove formally that the expected running time of this version of the algorithm is $O(n \log n)$. [7 marks]

   (A3) Provide a deterministic version of QuickSort that runs in worst-case running time $O(n \log n)$. Provide an argument for its running time. [6 marks]

   B. (B1) Propose a randomised algorithm for finding whether there is a value that occurs in more than $n/3$ entries of a given array of length n. The solution must perform $O(n)$ comparisons and return the correct answer with probability at least $1/2$. Analyse its correctness and argue that it has the required running time. [7 marks]

   (B2) Modify the algorithm of the previous question to provide a correct answer with probability at least 0.99, with no change in the asymptotic running time. Provide an argument for its correctness and its running time. [3 marks]

4. **A.** (A1) Define the decision versions of the maximum independent set and interval scheduling problems. Explain how a polynomial time algorithm for solving the decision versions can be used to design a polynomial time algorithm for computing the value of the optimal solution of the maximisation versions. [3 marks]

   (A2) Let INDEPENDENT SET be the decision version of maximum independent set defined above. Prove that INDEPENDENT SET is NP-complete. [8 marks]

   B. (B1) Formulate the maximum independent set problem on a graph G as an Integer Linear Program and write its LP-relaxation. [3 marks]

   (B2) Suppose that we round the relaxation to obtain an approximation algorithm for the problem. Prove that the approximation ratio of any such algorithm is $\Omega(n)$, where n is the number of nodes of the graph G. [4 marks]

   (B3) Does the lower bound in the previous question preclude the existence of an $o(n)$-approximate algorithm for the problem? Justify your answer. [3 marks]

   (B4) Design a greedy, polynomial-time, 4-approximate algorithm for the maximum independent set problem on graphs of degree at most 3.

   [4 marks]

5. **A.** (A1) Give the definition of a (feasible) flow in a flow network. [3 marks]

(A2) Describe the Ford-Fulkerson algorithm for finding the maximum flow on a given flow network. What type of algorithm is the Ford-Fulkerson algorithm? Justify your answer. [4 marks]

(A3) What is the running time of the Ford-Fulkerson algorithm assuming a network with integer capacities as inputs? Is it a polynomial time algorithm? [4 marks]

(A4) Describe an implementation of the algorithm which runs in time $O(nm^2)$ where n is the number of nodes and m is the number of edges of the network. You do not need to provide an argument for the running time. [4 marks]

B. A closure of a directed graph G is a set of nodes with no outgoing edges. The maximum weight closure problem is the problem of finding a closure of maximum weight, in a node-weighted graph G (with positive or negative weights).

(B1) Is the maximum weight closure problem NP-hard or polynomial time solvable? [2 marks]

(B2) In the former case, give a polynomial time reduction from the maximum independent set problem. In the latter case, provide a polynomial time algorithm for solving the maximum weight closure problem. [8 marks]