PAPER CODE NO.	EXAMINER : Dariusz Kowalski	Tel. No. 795 4285
COMP523	DEPARTMENT : Computer Science	

UNIVERSITY OF LIVERPOOL

RESIT EXAMINATIONS 2018/19

Advanced Algorithmic Techniques

TIME ALLOWED : Two and a Half Hours

INSTRUCTIONS TO CANDIDATES

Answer **FOUR** questions.
Each question is worth 25 marks.
For complexity analysis use asymptotic notation, always provide a justification for the complexity formulas and for correctness of your arguments.

Question 1

1 A. Consider the undirected graph with nodes $\{1, 2, 3, 4, 5, 6\}$ and edges:
$\{2, 6\}, \{5, 2\}, \{3, 5\}, \{6, 4\}, \{4, 5\}, \{2, 3\}, \{1, 6\}$

Give the representation of this graph as:

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

Find a BFS spanning tree rooted at node 2 and represent it as:

(iii) An adjacency matrix, [4 marks]
(iv) An adjacency list. [4 marks]

1 B. Design a time-efficient algorithm for checking if a given undirected graph $G = (V, E)$ with n nodes and m edges have all cycles of even length at least 4. [5 marks]

Analyse the time complexity and the additional memory used by this algorithm. [4 marks]

Question 2

2 A. Propose an efficient deterministic algorithm, which uses recursion, for finding a value that occurs in largest number of copies in a given sorted array of n numbers, unless every value occurs not more than $\sqrt{n}$ times (in such case the algorithm should return null). [6 marks]

Provide an argument supporting the correctness of your algorithmic solution. [3 marks]

Write down and justify the recursive formulas describing the time complexity of the recursive parts of your algorithm. [3 marks]

Give the asymptotic time complexity of your algorithm by solving the above recursive equations. [4 marks]

2 B. Give an implementation of Priority Queue using heaps. Describe the heap structure and the pseudo-codes of supporting operations. [9 marks]

Question 3

3 A. Explain the differences between the class of Monte Carlo randomised algorithms and the class of Las Vegas randomised algorithms. [4 marks]

Propose a randomised algorithm for finding whether there is a value that occurs in more than $n/4$ entries of a given array of length n. The solution must perform $O(n)$ comparisons, where n is the length of the array, and return the correct answer with probability at least $1/5$. Analyse its correctness and argue that it has the required time complexity. [12 marks]

3 B. Suppose there are three stations, each with a single packet, starting the slotted Aloha protocol at the same time. Compare the expected time until successful transmission of all three packets in two cases: when the probability of transmission is $1/2$ with the case when it is $1/4$. [9 marks]

Question 4

4 A. Discuss the equation P=NP, including the definition of each side of the equation and the meaning and some consequences of the equation. [4 marks]

Give a polynomial time algorithm for computing a minimum vertex cover set in a given complete rooted binary tree of $n = 2^k - 1$ nodes. Prove that it computes a minimum vertex cover set correctly. [8 marks]

What is the asymptotic time and memory complexity of this algorithm? [4 marks]

4 B. Give the 2-approximation polynomial time algorithm for solving the Minimum Weighted Vertex Cover problem based on integer programming technique. [6 marks]

Prove that this algorithm is a 2-approximation algorithm. [3 marks]

Question 5

5 A. Define the weighted interval scheduling problem for a given set of weighted intervals on the line. [2 marks]

Describe an optimal solution to the weighted interval scheduling problem based on dynamic programming and argue about its optimality and time complexity. [8 marks]

Design and analyse a greedy algorithm solving the weighted interval scheduling problem for inputs in which all intervals have weights equal to 2. [6 marks]

5 B. Consider the problem of finding the maximum number of edge-disjoint paths in a given undirected graph G: for two distinguished nodes s, t find the maximum number of paths from s to t which go through different edges. Is this problem NP-hard? If yes, give a polynomial-time reduction from the problem of finding a maximum independent set in a graph. If not, give a polynomial-time algorithm finding edge-disjoint paths that form a solution to the considered problem of the maximum number of edge-disjoint paths in a given graph. [9 marks]