PAPER CODE NO. **COMP309**	EXAMINER : **Dr Michele Zito** DEPARTMENT : Computer Science Tel. No. **43705**

THE UNIVERSITY
of LIVERPOOL

JANUARY 2006 EXAMINATIONS

Bachelor of Arts: Year 3
Bachelor of Science: Year 3

EFFICIENT SEQUENTIAL ALGORITHMS

TIME ALLOWED: Two Hours and a half

INSTRUCTIONS TO CANDIDATES

Answer **four** questions only. Each question is marked out of 25.

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY *of* LIVERPOOL

1. Dynamic programming.

   (a) Define the *Weighted Activity Selection* problem. **(5 marks)**

   (b) Give a pseudo-code description of a recursive algorithm that computes the total weight of an optimal activity selection for an instance of the Weighted Activity Selection problem. **(5 marks)**

   (c) Describe the recursion tree built during the execution of the algorithm obtained as a solution to the previous question on the following input:

	A_1	A_2	A_3	A_4	A_5	A_6
s	1	3	2	4	3	6
f	2	4	5	6	7	9
w	9	1	3	2	7	1

   (9 marks)

   (d) Comment on how could the running time of the recursive algorithm referred to in question (b) be sped up. **(6 marks)**

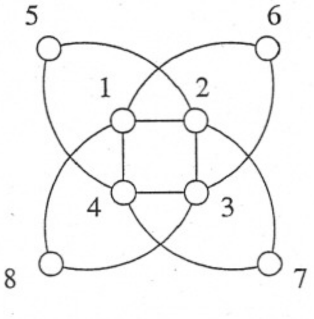
2. Knuth-Morris-Pratt algorithm (KMP).

   (a) Define the following terms: *prefix*, *suffix*. List all the prefixes and all the suffixes of the string ABACAB. **(5 marks)**

   (b) Compute the prefix function π for the pattern $P \equiv$ 124312441. **(5 marks)**

   (c) Simulate the execution of the KMP algorithm. In particular, list the sequence of values taken by the "state" variable q used in the pseudo-code of KMP when the pattern $P \equiv$ cbccbc is matched against the text $T \equiv$ cbccbacbcbccbccbcbb. (Hint: To simulate the algorithm correctly you will need to compute the prefix function π first). **(15 marks)**

3. (a) Prove (by splitting its vertex set into two disjoint independent sets) or disprove (by finding a triangle in it) that the graph in the figure below is bipartite. **(9 marks)**



   (b) Run the appropriate maximum cardinality matching algorithm on the graph in the figure above, assuming the initial matching is the set $M = \{\{1, 2\}, \{3, 4\}\}$. Draw all intermediate forests that you build. **(16 marks)**

THE UNIVERSITY
of LIVERPOOL

4. Turing machine models and space complexity.

    (a) Give a precise definition of a one tape deterministic Turing machine. **(6 marks)**

    (b) Explain in words (not more than 100) the notion of a *non-deterministic* Turing machine. **(5 marks)**

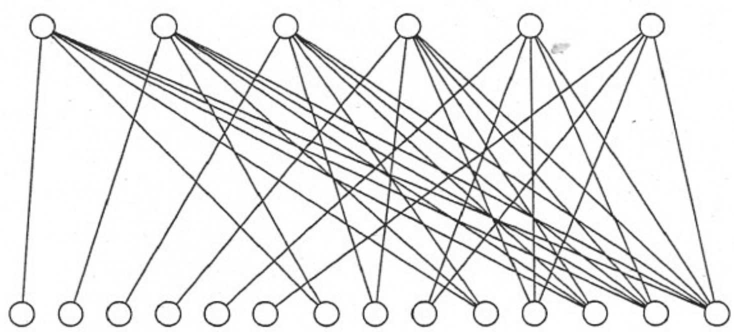
    (c) Given two strings T and P over $\{0, 1\}$ a valid shift is a position in T such that the string P can be matched in T starting from that position. Describe a non-deterministic Turing machine that solves the existence of a valid shift problem. **(10 marks)**

    (d) State Savitch's result on the relationship between the space complexity of deterministic and non-deterministic Turing machines. **(4 marks)**

5. Approximability of the Vertex Cover problem.

    (a) Define the VERTEX COVER problem. Your definition should express the fact that such problem is an optimisation problem (i.e. all four components in the definition of optimisation problem must be defined in the specific case). **(5 marks)**

    (b) Run the MM heuristic for VERTEX COVER on the graph below:

    What vertices are returned by such an algorithm?

    (Hints: just draw the matching edges and highlight the vertices of the resulting cover; alternatively label the vertices arbitrarily and state what the matching and the cover are). **(7 marks)**

    (c) Give a pseudo-code description of Savage's algorithm for finding a vertex cover in a graph based on building a depth-first spanning tree in the graph.

    (Hint: The toughest part of this question consist in stating what a depth-first spanning tree in a graph is.) **(8 marks)**

    (d) Run Savage's algorithm on the graph above. What vertices are returned by such an algorithm?

    (Hints: label the vertices of the graph in any way you wish; show the spanning tree and the cover vertices.) **(5 marks)**