

Duration: **1 hour**
 Aids Allowed: **NONE** (in particular, no calculator)

Student Number: _____

Last Name: _____

First Name: _____

Lecture Section: (circle one)	LEC01 M1, T10, W9 (Prof. Corneil)	LEC02 T4, W11, F1 (Prof. Pitt)	LEC03 W10, R3, F9 (Prof. Pitt)
---	---	--------------------------------------	--------------------------------------

Tutorial Section: (circle one)	TUT01 M10, GB405 (Mohammad)	TUT02 M10, GB412 (Lap Chi)	TUT03 R1, BAB024 (Periklis)	TUT04 R1, HA401 (Spyros)	TUT05 F10, BA2165 (Spyros)	TUT06 F10, BA2175 (Periklis)
--	-----------------------------------	----------------------------------	-----------------------------------	--------------------------------	----------------------------------	------------------------------------

*Do **not** turn this page until you have received the signal to start.*
 (In the meantime, please fill out the identification section above,
 and read the instructions below *carefully*.)

This term test consists of 4 questions on 6 pages (including this one), printed on one side of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete.*

Answer each question directly on the test paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of the page and *indicate clearly the part of your work that should be marked.*

You are strongly encouraged to write the test using a pen, because re-marking requests will *not* be granted for tests written using a pencil. (Simply cross off any part of your work that you do not want to be marked.)

If you are unable to answer a question (or part of a question), you will get 20% of the marks for the question (or part of the question) if you state clearly that you do not know how to answer. Note that you will *not* get those marks if your answer contains contradictory statements (such as “I do not know how to answer” followed or preceded by parts of a solution that have not been crossed off).

General Hint: We were careful to leave ample space on the test paper to answer each question.

MARKING GUIDE

1: _____ / 8

2: _____ / 7

3: _____ / 6

4: _____ / 9

TOTAL: _____ / 30

Good Luck!

Question 1. [8 MARKS]

Recall the *Task Scheduling Problem* from the previous test.

Input: A set of tasks $T = \{T_1 = (s_1, f_1, g_1), T_2 = (s_2, f_2, g_2), \dots, T_n = (s_n, f_n, g_n)\}$, where $s_i \in \mathbb{N}$ is the start time, $f_i \in \mathbb{N}$ is the finish time, and $g_i \in \mathbb{R}^+$ is the gain for task T_i (each task is well-defined, i.e., $s_i < f_i$ for all i).

Output: A subset of tasks $S \subseteq T$ such that every task in S can be scheduled on a single processor and the total gain of S is maximum. (For each i , task T_i can only be scheduled to start at time s_i and finish at time f_i , and no two tasks in S can overlap, although it is allowed to start one task at the same time that another task finishes.)

To use a Dynamic Programming algorithm to solve this problem, we first sort the tasks into non-decreasing finish time and then create a 2-dimensional table A , where $A[i, j]$ stores the maximum gain that can be achieved by scheduling tasks T_1, T_2, \dots, T_i such that all tasks complete no later than time j . On completion of the Dynamic Programming algorithm, the maximum gain possible is stored in the bottom right entry of A , namely $A[n, f_n]$.

We now want to use this table to output a schedule that achieves the maximum gain.

Part (a) [6 MARKS]

Assume that the values in array A have already been computed correctly. Write an algorithm in pseudo-code that takes A as input and uses it to output a schedule with the maximum gain.

Part (b) [2 MARKS]

Briefly justify that your algorithm is correct.

Question 2. [7 MARKS]

Consider the problem of assigning tutors to serve as invigilators for course examinations.

Each tutor, on reading the examination schedule, determines which courses he/she is able to invigilate. (Obviously, no tutor would try to invigilate two examinations held at the same time!) The set of tutors and courses may be considered as a bipartite graph with vertex set $T = \{t_1, t_2, \dots, t_m\} \cup C = \{c_1, c_2, \dots, c_n\}$ together with edges from T to C such that edge (t_i, c_j) indicates that tutor i can invigilate course j . Furthermore, we restrict the problem so that each tutor can invigilate at most 2 examinations and course c_j requires at most x_j invigilators (for $1 \leq j \leq n$).

The problem is to construct an assignment of tutors to courses that maximizes the number of tutors assigned.

Part (a) [5 MARKS]

Show how to solve this problem by translating it into a Network Flow problem, *i.e.*, describe clearly how to construct a network from an input to the problem given above.

Part (b) [2 MARKS]

Give a brief justification that the maximum flow in your network yields an assignment of tutors to courses that obeys the constraints of the problem and maximizes the number of tutors assigned.

Question 3. [6 MARKS]**Part (a)** [3 MARKS]

Prove the transitivity of $\xrightarrow{\text{poly}}$, i.e., show that for all languages X, Y, Z , if $X \xrightarrow{\text{poly}} Y$ and $Y \xrightarrow{\text{poly}} Z$, then $X \xrightarrow{\text{poly}} Z$.

Part (b) [3 MARKS]

Suppose that we have shown a language X to be NP -complete, and that for part of our proof we showed $Y \xrightarrow{\text{poly}} X$ for some language Y known to be NP -complete. Does $X \xrightarrow{\text{poly}} Y$? Justify.

Question 4. [9 MARKS]

Consider the following WEIGHTED VERTEX COVER problem.

Input: A graph $G = (V, E)$ with weighted *vertices* (i.e., each vertex $v \in V$ is assigned an integer weight $w(v) \in \mathbb{N}$) together with an integer target $K \in \mathbb{N}$.

Output: Is there a vertex cover of G with total weight at most K , i.e., is there a set of vertices $C \subseteq V$ such that every edge has at least one endpoint in C ($\forall (u, v) \in E, u \in C$ or $v \in C$) and the total weight of C is at most K ($\sum_{v \in C} w(v) \leq K$)?

Prove that WEIGHTED VERTEX COVER is *NP*-complete. You may refer to any of the problems listed on the last page of this test and use the fact that all of them are known to be *NP*-complete.

(HINT: The point of this question is to test that you understand and remember the steps for showing that a problem is *NP*-complete, by doing it on a specific problem. Therefore, make sure that you explain what you are doing; in particular, you may find it helpful to start by writing an outline of the steps that you will follow. Also note that you may not need all of the space below and on the next page.)

[more space on the next page...]

Question 4. (CONTINUED)

List of NP-complete problems.

In your solutions, you may refer to these problems and use the fact that they are NP-complete.

HAMILTONIAN CYCLE: Given a graph $G = (V, E)$, does G contain a Hamiltonian Cycle (*i.e.*, a cycle that visits each vertex exactly once)?

HAMILTONIAN PATH: Given a graph $G = (V, E)$, does G contain a Hamiltonian Path (*i.e.*, a path that visits each vertex exactly once)?

VERTEX COVER: Given a graph $G = (V, E)$ and a positive integer K , does G contain a vertex cover of size at most K (*i.e.*, is there a set of vertices $C \subseteq V$ such that $|C| \leq K$ and every edge of G has at least one endpoint in C)?

GRAPH COLOURING: Given a graph $G = (V, E)$ and a positive integer K , is G K -colourable (*i.e.*, is there a colouring of the vertices of G using at most K colours such that adjacent vertices have different colours)?

Total Marks = 30