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

Student Number: |_|_|_|_|_|_|_|_|_|

Last Name: ____________________

First Name: ____________________

Lecture Section: **(cirle one)**	LEC01 Prof. Corneil	LEC02 Prof. Pitt	LEC03 Prof. Pitt

Tutorial Section: **(cirle 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 3 questions on 5 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: ______/16

3: ______/ 6

TOTAL: ______/30

Good Luck!

Question 1. [8 MARKS]

Let $G = (V, E)$ be a connected graph with a cost $c(e) \in \mathbb{N}$ associated with each edge $e \in E$. Let $S \subseteq E$ be a subset of the edges. Given G and S, we want to find a spanning tree T of G such that:

(i) every element of S is in T (*i.e.*, $S \subseteq T$);
(ii) T has minimum cost among all spanning trees satisfying (i).

Part (a) [2 MARKS]

Does such a tree T always exist? Briefly justify.

Part (b) [5 MARKS]

If such a tree T does exist, design an algorithm to find T. You can write your algorithm in pseudo-code, or describe *clearly* how to modify an existing algorithm.

Part (c) [1 MARK]

What is the complexity of your algorithm? Briefly justify.

Question 2. [16 MARKS]

Consider the following *Task Scheduling Problem*.

Input: A set of tasks $\{T_1 = (s_1, f_1, g_1), T_2 = (s_2, f_2, g_2), \ldots, T_n = (s_n, f_n, g_n)\}$, 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 set 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 exactly and finish at time f_i exactly, and no two tasks in S can overlap, although it is allowed to start one task at the same time that another task finishes.)

Part (a) [2 MARKS]

What parameter would you use to sort the tasks for a Greedy algorithm? Be specific and briefly justify.

Part (b) [2 MARKS]

Give an example to show that this Greedy algorithm will not always produce an optimum schedule.

Part (c) [3 MARKS]

What 2-dimensional table would you use to store intermediate values for a Dynamic Programming algorithm? State precisely the value stored in the i, j-th entry of your table (including what specific subproblem it corresponds to).

Part (d) [1 MARK]

What value of your table stores the maximum profit that can be achieved?

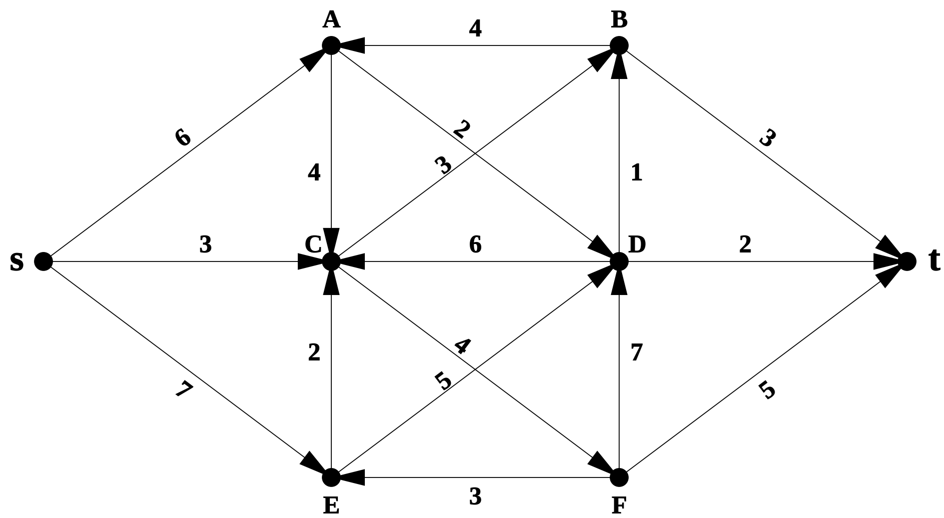
Question 2. (CONTINUED)

Part (e) [8 MARKS]

Write an algorithm in pseudo-code to fill out the values in your table from part (c). Briefly justify that your algorithm is correct.

Question 3. [6 MARKS]

Use the Ford-Fulkerson algorithm to compute a maximum flow in the network below (where each edge has the indicated capacity). State precisely each augmenting path that you use; you **must** use the path given below the picture as your first augmenting path. To justify that your final flow is maximum, exhibit an appropriate cut in the network and state the relationship between the value of your flow and the capacity of your cut.



First augmenting path: $s \xrightarrow{0/3} C \xrightarrow{0/3} B \xrightarrow{0/4} A \xrightarrow{0/2} D \xrightarrow{0/2} t$. Residual capacity: ___ (*fill in the value*)

Total Marks = 30