UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

FINAL EXAMINATIONS, APRIL 2001
Third Year

CSC 366S — THE THEORY OF COMPUTATION
Exam Type: C
Examiner — Allan Borodin

First Name: _____

Last Name: _____

Student # : _____

Section: (check one) ☐ L0101 ☐ L0102

---

*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 examination consists of six (6) questions on nine (9) pages including this title page. *When you receive the signal to start, please make sure that your copy of the examination is complete.* Answer each question directly on the exam 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** which part of your work should be marked.)

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Indeed erroneous or irrelevant statments can lead to deductions in your grade. Also, unreadable answers will be given zero (0); so write legibly. In your answers, you may use any theorems or facts given during the course, as *long as you state them clearly*. You must prove any other theorems or facts needed for your solution.

General hint: I tried to be careful to leave ample space on the exam paper to answer each question. If you find yourself using much more room than what is available, you're probably missing something. Also, if you cannot complete one of your solutions, at least give a *precise* outline of the steps required, in order to get part marks.

# 1:	_____/10
# 2:	_____/20
# 3:	_____/30
# 4:	_____/10
# 5:	_____/10
# 6:	_____/10
TOTAL:	_____/90

Good Luck!

CONT'D...

Question 1. [10 MARKS]

Let $G = (V, E)$ be a connected undirected graph, let $c : E \rightarrow \Re^{\geq 0}$ be a cost function on the edges of G, and let $T_0 = (V, E')$ be a subgraph of G (that is, $E' \subseteq E$) such that T_0 is a forest (i.e. T_0 has no cycles). Note: $\Re^{\geq 0}$ stands for the non negative reals.

Consider the following **Modified Kruskal's Algorithm:**

Sort the edges $\{e_1, \ldots, e_m\}$ of $E - E'$ so that: $c(e_1) \leq c(e_2) \leq \ldots \leq c(e_m)$
$T \leftarrow T_0$
for $i : 1..m$
  if $T \cup \{e_i\}$ has no cycle then
    $T \leftarrow T \cup \{e_i\}$
  end if
end for

We claim that this Modified Kruskal's algorithm will compute a minimum cost spanning tree T^* in G containing T_0. That is, amongst all spanning trees containing T_0, T^* has minimum cost.

In order to prove this claim we need to prove an appropriate inductive statement concerning T_i, the forest that has been computed at the end of the i^{th} iteration of the loop.

State an appropriate inductive statement and indicate how the above claim about the Modified Kruskals algorithm follows from the inductive statement. You do NOT have to prove the inductive statement.

Question 2. [20 MARKS]

We define the following "no consecutive items knapsack problem": There is a sequence of jobs $I = \{(w_k, g_k) | 1 \leq k \leq n\}$ and a knapsack bound C. A feasible subset $S \subseteq \{1, 2, \ldots n\}$ is a set sastisfying

- $\sum_{i \in S} w_i \leq C$ (as in the usual knapsack problem)
- $i \in S$ implies $i - 1 \notin S$ for $1 \leq i \leq n$.
  That is, the second condition states that if job i is placed in the knapsack then job $i - 1$ can not be in the knapsack.

We assume that C and each w_k is a positive polynomial size integer (i.e. polynomial in the number of jobs) and we want to construct a polynomial time dynamic programming DP algorithm for maximizing the profit achieveable by a feasible set of jobs.

- Define an appropriate semantic array $P[k, t]$ for $0 \leq k \leq n$, $0 \leq t \leq C$.
- Give a recurrence defining an array $\tilde{P}[k, t]$ such that $P[k, t] = \tilde{P}[k, t]$ for all relevant values of k and t. Briefly justify why your recurrence works.

Question 3. [30 MARKS]

Assume $P \neq NP$. Then (based on this assumption) for each of the following problems, state whether or not it has a polynomial time algorithm, and justify your answer. That is, give a brief outline of the algorithm or the reason why there is none. You may refer to any NP-complete or NP-hard problems given in the course. *No credit will be given for a guess without some reasonable justification. Some of the problems below are quite easy (given known facts) and others require a little more thought.*

Part (a) [6 MARKS]

Input: A set S of clauses such that each clause has an odd number of literals. Recall that a clause is a disjuntion of literals.

Output : YES iff there is a truth assignment which satisfies each clause in S?

Part (b) [6 MARKS]

Input: An undirected graph $G = (V, E)$.

Output: YES iff G has at least two disjoint 5-cliques.

Part (c) [6 MARKS]

Input: An undirected graph $G = (V, E)$ and an integer m.

Output: YES iff G has at at least two disjoint m-cliques.

Part (d) [6 MARKS]

Input: A set of (unit profit) intervals $I_1, \ldots I_m$.

Output: YES iff there is a unique optimal feasible (i.e. the jobs can be simulataneously scheduled on one machine) subset S^* of these intervals. That is, S^* is a feasible subset of intervals and any other (different) feasible subset S' of intervals has cardinality $|S'| < |S^*|$.

Part (e) [6 MARKS]

Consider the following "almost all job scheduling problem":

Input: A set of m jobs $J_1, \ldots J_m$ with $J_k = (r_k, d_k, p_k)$ specifying the release time, deadline and processing time of job J_k. All input parameters are positive integers represented in binary.

Output: YES iff there is a feasible schedule (on one machine) of these jobs such that at least m-1 jobs are scheduled.

Question 4. [10 MARKS]

Consider the following "3-way PARTITION" problem:

Input: A set m positive integers (represented in binary) $\{b_1, b_2, \ldots, b_m\}$.

Output: YES iff there is a disjoint partition of $\{1, \ldots, m\}$ into three sets S_1, S_2, S_3 such that $\sum_{i \in S_1} b_i = \sum_{i \in S_2} b_i = \sum_{i \in S_3} b_i$.

Show that this "3-way PARTITION" problem is *NP*-complete. In particular, show that it is in *NP* and that it is $NP - hard$ (by defining and provng an appropriate polynomial time transformation from some known *NP*-hard problem).

Question 5. [10 MARKS]

Consider the language $L = \{\langle M \rangle | \mathcal{L}(M) = \{ww^R | w \in \Sigma^*\}\}$. That is, L is the set of representations of Turing machines M such that the language accepted by M is precisely the set of palindromes.

Part (a) [5 MARKS]

Show that L is not semi-decidable. Establish an appropriate transformation from a language known to be not semi-decidable.

Part (b) [5 MARKS]

Show that $\bar{L}$ is not semi-decideable. Establish an appropriate transformation from a language known to be not semi-decidable.

Question 6. [10 MARKS]

Consider the language $L = \{\langle M \rangle | \mathcal{L}(M) \subseteq \{ww^R | w \in \Sigma^*\}\}$. That is, L is the set of representations of Turing machines M such that the language accepted by M is contained in the set of palindromes.

Show that $\bar{L}$, the complement of L is semi-decidable but not decidable.