University of Toronto
Faculty of Arts and Science
August 2003 Examinations

CSC 364H1S

Duration - 3 hours
No Aids Allowed

There are 7 questions worth a total of 80 marks.
Answer all questions on the question paper, using backs of pages for scratch work.
Check that your exam book has 11 pages (including this cover page).

PLEASE COMPLETE THIS SECTION:

Name ______________________________
(Please underline your family name.)

Student Number ______________________________

Section (please circle): L5101 (Mohammad Salavatipour) L5201 (Antonina Kolokolova)

FOR USE IN MARKING:

1. __________/15
2. __________/10
3. __________/10
4. __________/10
5. __________/15
6. __________/10
7. __________/10

Total:__________/80

[15] 1. **Greedy algorithms**

The following problem is a variation of scheduling in which there are no deadlines or profits.

Suppose you are given a set $\{a_1, \ldots, a_n\}$ of jobs, where job a_i requires $p_i \in \mathbb{N}$ units of processing time to complete, once it has started. For a schedule S (which contains all the jobs), let c_i be the completion time of a job a_i, that is the time at which job a_i started plus the number of units p_i it needs to finish. Your goal is to find a schedule containing all the jobs in which the average completion time $\frac{1}{n}\Sigma_{i=1}^{n} c_i$ is minimal.

[2] (a) Give an example with two jobs and two possible schedules with different average completion time.

[5] (b) Describe a greedy algorithm you would use to solve this problem.

[3] (c) State the definition of *promising* that you can use to prove the correctness of your algorithm.

[5] (d) Briefly outline the proof of the correctness of your algorithm.

[10] 2. **Dynamic programming**

You are given a DNA string y of length m over alphabet $\{A, T, C, G\}$, and "genes" strings $x_1, \ldots, x_n$ of lengths $k_1, \ldots, k_n$ respectively, also over the alphabet $\{A, T, C, G\}$, where each x_i has an associated probability measure $0 < p_i < 1$. The goal is to find the most probable way of concatenating (with repetitions allowed) some strings from $\{x_1, \ldots, x_n\}$ to obtain y, if one exists. The probability of a sequence of strings is a product of the probabilities of individual strings.

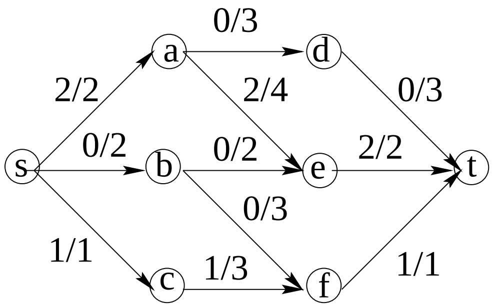
For example, take $y = AGAGTC$, $x_1 = TC$, $x_2 = AG$, $x_3 = AGAG$ with probabilities $p_1 = 0.3, p_2 = 0.7, p_3 = 0.4$. Then the two possible mappings of y are $y = x_2 x_2 x_1$ with probability 0.147 and $y = x_3 x_1$ with probability 0.12. Your algorithm should find the first mapping, since it has higher probability.

[4] (a) Give the definition of an array A that you will use to solve the problem. (Hint: use a one-dimensional array and consider prefixes of y of increasing length.) How can you get the probability of the most probable sequence from this array A? How can you check if there is no solution?

[6] (b) Give a recurrence to compute A, including the initialization. Briefly explain how your recurrence works.

[10] **3. Network Flows**

[5] (a) Specify an augmenting path in the flow network below by listing the sequence of the nodes in the path. What is the value of the flow after applying this augmenting path?



[5] (b) In a certain college, there are n instructors and m different courses that the college can offer. A course j can have up to k_j sections offered in one semester. For every instructor i, there is a list of courses $C_i \subseteq \{1, \ldots, m\}$ that this instructor can teach. In one semester, an instructor can teach no more than five sections of (possibly different) courses. This is similar, but not the same as the problem from assignment 2.

State how you can find a maximal number of *sections* of all courses that the college can offer by using flow networks. That is, give a flow network modeling the problem, and state how to use it to get the solution.

[10] 4. **Search-to-decision reduction**

The search problem LargestSubsetSum is defined as follows. Given a set of natural numbers $\{a_1, \ldots, a_n\}$ and a natural number t, find a set $S \subseteq \{a_1, \ldots, a_n\}$ such that $\Sigma_{a_j \in S} a_j = t$ and, in addition, $|S|$ is maximal. For example, if the input is $\{1, 2, 3, 5\}, t = 6$, then the maximal size S is $\{1, 2, 3\}$, not $\{1, 5\}$.

[3] (a) Define a corresponding **NP** decision problem LSSD that can be used to solve LargestSubsetSum (note that it is **not** the same as SubsetSum).

[3] (b) State how to find the *number of elements* in the solution to LargestSubsetSum (that is, find maximal $|S|$ for which $\Sigma_{a_j \in S} a_j = t$) by reduction to LSSD that you just defined. Your reduction should make no more than $O(\log n)$ calls to the decision problem LSSD, where n is the number of elements $a_1 \ldots a_n$.

[4] (c) Now state how to find a solution S of maximal size, again by reduction to LSSD. You can use your answer from the previous subquestion.

[15] 5. **NP-completeness**

For each of the decision problems below, either show it is in **P** or show it is **NP**-complete. To show it is in **P**, outline a polynomial-time algorithm which solves it. (You may refer to any polynomial-time algorithm discussed in this course.) To show that it is **NP**-complete, pick an **NP**-complete problem from the the list on the last page of this exam, state what you are reducing to what and give a construction for a general instance.

You do not need to show the problem is in **NP** or give the correctness proof in both directions.

[5] (a) **s-t HamPath**

Input: $\langle G, s, t\rangle$ where G is an undirected graph and s, t are nodes in G.

Output: "Yes" if G has a simple path from s to t containing all vertices; "no" otherwise.

[5] (b) **Large clique**

Input: An undirected graph G.

Output: "Yes" if G has a clique of size at least $n - 3$ where n is the number of vertices in G; "no" otherwise.

[5] (c) Do only **one** of the following two problems.

Even partition
Input: $\langle a_1, ..., a_n \rangle$, where $a_1, ..., a_n$ are nonnegative integers given in binary.
Output: "Yes" if there is a partition of the elements into two sets such that the sums of elements of both sets are equal and at least one set has an even number of elements; "no" otherwise.

Double-SAT
Input: $\langle F \rangle$, where F is a propositional formula in conjunctive normal form.
Output: "Yes" if F has at least two satisfying assignments; "no" otherwise.

[10] 6. **Properties of languages**

[5] (a) Let $L_1 \in \mathbf{NP}$ and $L_2 \in \mathbf{P}$. Prove that $L_3 \in \mathbf{NP}$, where $L_3 = \{x | x \in L_1 \text{ and } x \notin L_2\}$.

[5] (b) Is it true that every subset of a decidable language is decidable? Prove or give a counter-example.

[10] 7. **Undecidability**

Consider the following language:

$$L = \{\langle M \rangle | M \text{ is a Turing machine which accepts a string consisting of only 0's}\}$$

[5] (a) Prove that L is semi-decidable.

[5] (b) Prove that L is not decidable. You can use any of the undecidable problems from the last page of this exam for your undecidability proof.

LIST OF **NP**-COMPLETE PROBLEMS:

3SAT
Input: $\langle F\rangle$, where F is a propositional formula in conjunctive normal form with at most three literals per clause.
Output: "Yes" if F is satisfiable, "no" otherwise.

HamCycle
Input: $\langle G\rangle$, where G is an undirected graph.
Output: Does G have a Hamiltonian cycle, that is, a cycle containing all vertices?

PARTITION
Input: $\langle a_1, ..., a_n\rangle$, where $a_1, ..., a_n$ are nonnegative integers given in binary.
Output: "Yes" if there is a set $S \subseteq \{1, ..., n\}$ such that $\sum_{i\in S} a_i = \sum_{i\notin S} a_i$; "no" otherwise.

SubsetSum
Input: $\langle a_1, ..., a_n, t\rangle$, where $a_1, ..., a_n, t$ are nonnegative integers given in binary.
Output: "Yes" if there is a subset $S \subseteq \{1, ..., n\}$ such that $\sum_{i\in S} a_i = t$; "no" otherwise.

GKD (General Knapsack Decision)
Input: $\langle w_1, ..., w_n, g_1, ..., g_n, C, B\rangle$, where $w_1, ..., w_n, g_1, ..., g_n, C, B$ are positive integers given in binary.
Output: "Yes" if there is a set $S \subseteq \{1, ..., n\}$ such that $\sum_{i\in S} w_i \leq C$ and $\sum_{i\in S} g_i \geq B$; "no" otherwise.

3COL
Input: $\langle G\rangle$, where G is an undirected graph.
Output: "Yes" if there is a vertex colouring of G using at most three colours such that no two adjacent vertices get the same colour; "no" otherwise.

Clique
Input: $\langle G, B\rangle$, where G is an undirected graph and B is a positive integer.
Output: "Yes" if G has a clique of size B; "no" otherwise ($V' \subseteq V$ is a clique if every two nodes of V' are connected by an edge in G.)

IS (Independent Set)
Input: $\langle G, B\rangle$, where G is an undirected graph and B is a positive integer.
Output: "Yes" if G has an independent set of size B? ($V' \subseteq V$ is an independent set if no two nodes in V' are connected by an edge in G.)

LIST OF UNDECIDABLE LANGUAGES

$A_{TM} = \{\langle M, w\rangle | M$ is a Turing Machine and $w \in \mathcal{L}(M)\}$
$HB = \{\langle M\rangle | M$ is a Turing Machine and M halts on blank input$\}$
$NE = \{\langle M\rangle | M$ is a Turing Machine and $\mathcal{L}(M) \neq \emptyset\}$
$T = \{\langle M\rangle | M$ is a Turing Machine and M halts on every input$\}$
$INF = \{\langle M\rangle | M$ is a Turing Machine and M accepts infinitely many strings$\}$
$ALL = \{\langle M\rangle | M$ is a Turing Machine and $\mathcal{L}(M) = \Sigma^*\}$
$DIAG = \{\langle M\rangle | M$ is a Turing Machine and $\langle M\rangle \notin \mathcal{L}(M)\}$