Discrete Mathematics Honors Examination
Swarthmore College
Monday, May 12, 1997

This exam has two parts. Part I focuses mostly on basic knowledge and technical skills, while Part II focuses on proof techniques and creativity. Provide answers and brief justifications of your solutions to the questions in Part I, but give more details in your solutions to questions in Part II. Do at least 5 of the questions in Part I, and at least 3 of the questions in Part 2.

## Part I.

1. Solve this recurrence relation using generating functions:

$$T(n) = 5T(n-1) + 6T(n-2),$$

$$T(0) = 2 \text{ and } T(1) = 5$$

.

2. For each of the following functions, give a recurrence relation that it satisfies.
   (a) $T(n) = n!$
   (b) $T(n) = n^2 + 5n - 3.$
   (c) $T(n) = 8^n - 2 * 3^n$

3. Ma and Pa Shoe have 8 children and no two children in the family can keep from fighting with each other, and so the only way to keep them quiet at dinner is to make sure that no two children sit next to each other. Fortunately, Ma and Pa have a round table which can accomodate as many people as they need. (For the following questions, assume that there is no head to the table, but circular orderings which are flips of each other are distinguishable.)
   (a) How many ways can Ma and Pa Shoe arrange a group of 8 adults and 8 children around their table so that no two children fight?
   (b) If one of the children go to a friend's house, then how many ways can Ma and Pa Shoe arrange 8 adults and 7 children around their table?

4. Consider the Stirling numbers of the second kind, $S(m, n)$, which denote the number of partitions of $m$ different objects into $n$ sets. Give a recurrence relation for $S(m, n)$, and prove its correctness.

5. A *derangement* of $1, 2, \ldots, n$ is an ordering of of $1, 2, \ldots, n$ such that no integer $i$ appears in the $i^{th}$ position in the ordering. Use the principle of inclusion-exclusion to give a formula for the number of derangements of $1, 2, ..., n$.

6. Suppose $\Pi_1, \Pi_2, \ldots, \Pi_k$ are decision problems, and suppose that for each $i = 1, 2, \ldots, k-1$, there is a polynomial time reduction from $\Pi_i$ to $\Pi_{i+1}$.

   (a) If $\Pi_4$ is solvable in polynomial time, what can you say about solvability in polynomial time for each of the other problems?

   (b) If $\Pi_4$ is solvable in polynomial time and $\Pi_1$ is NP-hard, what can you say about the question $P = NP$?

   (c) If $\Pi_1$ is solvable in polynomial time and $\Pi_4$ is NP-hard, what can you say about the question $P = NP$?

7. Let $L$ be any language over the alphabet $\Sigma = \{a, b\}$.

   (a) Is the language $L_1 = \{w \in L : |w| \leq 50\}$ regular?

   (b) Is the language $L_2 = \{w \in L : |w| \geq 50\}$ regular?

8. Is the language $L = \{a^n b^n c^k : n, k \geq 1\}$ context free?

**Part II.**

1. For a graph $G = (V, E)$, we define the degree of a node $v$, denoted $deg(v)$, to be $|\{w : (v, w) \in E(G)\}|$. We let $\Delta(G)$ be the maximum degree of any node in $G$. A *proper* vertex coloring is an assignment of colors to the nodes of the graph so that no edge connects nodes of the same color.

   (a) Prove or disprove: *For all such graphs $G$, it is impossible to properly vertex-color $G$ with fewer than $\Delta(G)$ colors.*

   (b) Prove or disprove: *For all such graphs $G$, it is possible to properly vertex-color $G$ with exactly $\Delta(G)$ colors.*

   (c) Prove or disprove: *For all such graphs $G$, it is possible to properly vertex-color $G$ with exactly $\Delta(G) + 1$ colors.*

2. The **Partition** problem is as follows:

   - Input: set of positive integers $\{x_1, x_2, \ldots, x_n\}$.
   - Question: is there a partition of $x_1, x_2, \ldots, x_n$ into two sets $A$ and $B$ such that the sum of the numbers in $A$ is the same as the sum of the numbers in $B$?

The **Knapsack** problem is as follows:

- Input: set of positive integers $\{x_1, x_2, \ldots, x_n\}$, and integer $B$.
- Question: is there a set $X \subseteq \{x_1, x_2, \ldots, x_n\}$ such that the sum of the numbers in $X$ is exactly $B$?

(a) Assume that Partition is NP-hard, and prove that Knapsack is NP-hard.

(b) Consider the following dynamic programming formulation for solving Knapsack. Let $M$ be a boolean matrix with rows indexed from $i = 0$ to $i = n$ and columns indexed from $j = 0$ to $j = B$, and let $M(i, j)$ be true if and only if there is a subset of $\{x_1, x_2, \ldots, x_i\}$ summing to $j$. Describe a dynamic programming algorithm based upon this matrix to solve the Knapsack problem, and analyze its running time.

(c) Comment on the question of $P = NP$? with respect to these two results.

3. If $G = (V, E)$ is a graph, $V' \subset V$ and $G'$ is a subgraph of $G$ which contains every edge of $G$ connecting two vertices of $V'$, then $G'$ is called the subgraph *induced* or *spanned* by $V'$. A *cycle* in a graph $G = (V, E)$ is a sequence of nodes $v_1, v_2, \ldots, v_k$ such that $(v_i, v_{i+1}) \in E$ for $i = 1, 2, \ldots, k-1$ and $(v_1, v_k) \in E$. A *chord* in the cycle $v_1, v_2, \ldots, v_k$ is an edge between two non-consecutive nodes in the cycle. A **triangulated** graph is a graph $G = (V, E)$ such that all cycles of size four or more contain chords (i.e. all vertex-induced cycles are of size at most three). Thus acyclic graphs and complete graphs are triangulated. A *simplicial* node in a graph $G$ is a node $v$ such that its neighbors form a clique (i.e a collection of nodes every two of which are adjacent). A *perfect elimination ordering* is an ordering of the vertices $v_1, v_2, \ldots, v_n$ such that each $v_i$ is simplicial in the subgraph of $G$ induced by $v_i, v_{i+1}, \ldots, v_n$. The following theorem can be proven: **Theorem: $G$ has a perfect elimination ordering (there can be more than one) if and only if $G$ is triangulated.**

(a) Prove only that a graph that has a perfect elimination ordering is triangulated.

(b) Given a perfect elimination ordering $v_1, v_2, \ldots, v_n$ for $G = (V, E)$, define $X(v_i) = \{v_j : j \geq i \text{ and } (v_i, v_j) \in E\}$ (so $X(v_i)$ depends upon the graph and upon the perfect elimination ordering). Prove that the maximum clique in a triangulated graph $G$ is of the form $\{v\} \cup X_v$.

(c) Use the fact that every triangulated graph has a perfect elimination ordering and the result of the previous question to prove that the size of the maximum clique in a triangulated graph can be determined in polynomial time.

(d) The chromatic number of a graph is the minimum number of colors needed to color the vertices of a graph so that no pair of adjacent nodes are assigned the same color. Prove that every the chromatic number of a triangulated graph is the size of its maximum clique, and give a polynomial time algorithm to compute the optimal coloring.

4. Let $G = (V, E)$ be a connected graph and let $w : E \to R$ be a function assigning real weights to the edges of $G$. A *minimum spanning tree* of $G$ is a subgraph $T = (V, E(T))$ of $G$ such that $T$ is a tree (i.e. connected and acylic graph) containing all the nodes of $G$ and minimizing $\sum_{e \in E(T)} w(e)$.

   Consider the following condition (which we will call the $*$-condition) on a (not necessarily minimum) spanning tree $T$ for $G$ *For all edges* $e \in E - E(T)$, *let* $\gamma_e$ *be the unique simple cycle in* $T \cup \{e\}$, *and let* $e'$ *be the heaviest edge in* $\gamma_e$. *Then* $w(e') \leq w(e)$.

   (a) Prove that every minimum spanning tree $T$ satisfies the $*$-condition.

   (b) Now suppose that all the weights in $E$ are distinct, so that $w(e) \neq w(e')$ for all $e \neq e'$. Show that any spanning tree $T$ which satisfies the $*$-condition is a minimum spanning tree. (Hint: consider a spanning tree $T'$ which has a smaller weight than $T$, and consider $E(T) \Delta E(T')$.)

   (c) Deduce from (a) and (b) above that the minimum spanning tree on a graph in which all the edge weights are distinct is unique.

5. State the *Polya-Redfield Theorem*, sketch the proof, and give a simple example of how it can be applied.

6. Let $R$ be a regular language over an alphabet $\Sigma$ and let $L^{1/2}$ and $L^2$ be the languages:

$$L^{1/2} = \{w \in \Sigma^* : ww \in R\}$$

   and

$$L^2 = \{ww \in \Sigma^* : w \in R\}$$

   (a) Is $L^{1/2}$ a context free language? Justify your answer.

   (b) Is $L^2$ a context free language? Justify your answer.