Practice Quiz 1 Solutions

Problem 1 [6 points] **Quantifiers**

Consider the following proposition.

$$\forall x \forall y \; (x < y \rightarrow \exists z \; (x < z \wedge z < y))$$

(a) [2 points] Prove or disprove this proposition, assuming that the universe is the set of integers.

The proposition is false. In the case where $x = 0$ and $y = 1$, there does not exist an integer z such that $x < z$ and $z < y$.

(b) [4 points] Write the negation of this proposition. Your solution may *not* use the $\neg$ symbol, but may use any of the relations $=$, $<$, $>$, $\leq$, or $\geq$.

$$\exists x \exists y \; (x < y \wedge \forall z \; (x \geq z \vee z \geq y))$$

Problem 2 [10 points] **Induction**

Define the matrix

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Let M^n denote the product of n copies of the matrix M. Use ordinary induction to prove that for all $n \geq 1$, the upper right entry of M^n is F_n, the n-th Fibonacci number. State your inductive hypothesis clearly.

For reference, the standard formula for the product of 2×2 matrices is given below.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix} 1g$$

Recall that $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $\ldots$.

Proof. The proof is by induction. Let $P(n)$ be the proposition that

$$M^n = \begin{bmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{bmatrix}.$$

In the base case, $P(1)$ is true because

$$\begin{aligned} M^1 &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} F_0 & F_1 \\ F_1 & F_2 \end{bmatrix}. \end{aligned}$$

■

In the inductive step, for $n \geq 1$ assume $P(n)$ to prove $P(n+1)$. The following equalities hold.

$$\begin{aligned} M^{n+1} &= M^n \cdot M \\ &= \begin{bmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} F_n & F_{n-1} + F_n \\ F_{n+1} & F_n + F_{n+1} \end{bmatrix} \\ &= \begin{bmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{bmatrix} \end{aligned}$$

The first step uses the definition of M^n. The second step uses the induction hypothesis and the definition of the matrix M. We carry out a matrix multiplication in the third step and use the Fibonacci recurrence in the last step. This shows that $P(n)$ implies $P(n+1)$, and the claim is proved by induction.

Problem 3 [10 points] **Structural Induction**

Consider the sets $S_1, S_2, \ldots$ defined inductively as

- $S_0 = \{0, 1\}$
- For all $n \geq 1$, $S_n = \left\{\frac{x+y}{2} \mid x, y \in S_{n-1}\right\}$

i.e., where S_n contains the average values of each pair of numbers in S_{n-1}. Note we allow $x = y$ in the definition of S_n.

(a) [1 point] Write out the elements of each of the following sets: S_1, S_2, and S_3.

$$\begin{array}{ll} S_1 & \{0, 1/2, 1\} \\ S_2 & \{0, 1/4, 1/2, 3/4, 1\} \\ S_3 & \{0, 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8, 1\} \end{array}$$

(b) [2 points] Generalize (in your own words or in symbols) to give the elements of S_n. S_n contains $2^n + 1$ elements (including 0 and 1) evenly spaced along the unit interval; i.e.,

$S_n = \{i/2^n | 0 \leq i \leq 2^n\}$.

(c) [7 points] Prove by induction that for all $n \in \mathbb{N}$, $1/3 \notin S_n$. You will probably want to strengthen the induction hypothesis. You may not assume your generalization from the previous part is true; you must prove that statement if you wish to use it.

Strengthen the theorem to

$$P(n) = \text{“For all } a/b \in S_n \text{ with } a, b \in \mathbb{N} \text{ where } a/b \text{ is in reduced form, } b = 2^k \text{ for some } k\text{”}$$

Clearly, if $\forall n\ P(n)$, then $\forall n\ 1/3 \notin S_n$. Thus, we need only prove $\forall n\ P(n)$.

Base cases. $0 = 0/2^0$, $1 = 1/2^0$.

Induction. Assume $P(n)$. By the construction of S_{n+1}, all elements of S_{n+1} are of the form $(x + y)/2$ for some $x, y \in S_n$. Fix any two such elements $x, y \in S_n$, where (by the induction hypothesis) $x = a/2^k$ and $y = c/2^\ell$ with $a, b, k, \ell \in \mathbb{N}$.

Assume WLOG $k \geq \ell$. Then,

$$\begin{aligned} (x+y)/2 &= (a/2^k + c/2^\ell)/2 \\ &= (a/2^k + c2^{k-\ell}/2^k)/2 \\ &= (a + c2^{k-\ell})/2^{k+1} \end{aligned}$$

While this may not be in reduced form, reduction cannot introduce any non-2 factors into the denominator, so this fits the form specified by the theorem. Therefore, $P(n+1)$.

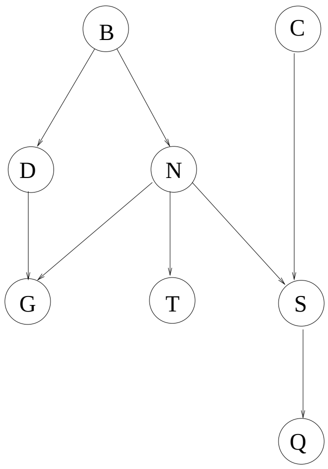
By induction, $\forall n\ P(n)$. Thus, $\forall n\ 1/3 \notin S_n$.

Problem 4 [10 points] **Precedence Relations**

Death is planning the end of the world. This involves a number of tasks each of which takes one minute to complete. The prerequisites associated with these tasks are listed below.

Abbrv.	Task	Prerequisites
C	Compose a requiem	
N	Notify the UN	B
D	Signal the daemons	B
B	Blow the trumpet	
T	Sell T-shirts: "All I got was this lousy t-shirt."	N
Q	Grade the 6.042 quiz	S
G	Open the gates	D,N
S	Put out the sun	C,N

(a) [5 points] Represent the tasks and their prerequisites as a directed graph.



(b) [5 points] Death is omnipotent and can therefore work on as many tasks at a time as he wishes. What is the minimum amount of time required for him to end the world? Why? (An explanation without proof is fine.)

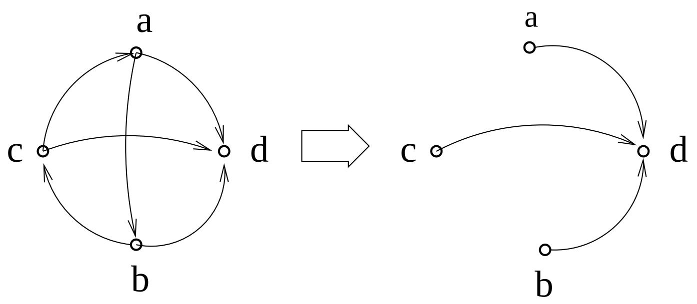
The minimum required time is 4 (length of the critical path B-N-S-Q).

Problem 5 [10 points] **Graphs**

The vertices of a directed graph can be partitioned into *strongly-connected components*. Two vertices u and v belong to the same strongly connected component if there is a path from u to v and a path from v to u. A strongly connected component may consist of a single vertex.

Math Moose has noticed a strange phenomenon. He starts with a directed graph and partitions the vertices into strongly-connected components. He then removes all edges that connect two vertices in the same strongly-connected component. The resulting graph always seems to be acyclic.

An example is worked out in the figure below. The original directed graph is shown on the left. The vertices a, b, and c form one strongly-connected component. The single vertex d forms a second strongly-connected component. Math Moose removes the edges $a \to b$, $b \to c$, and $c \to a$, since these edges connect vertices in the same strongly-connected component. The resulting graph is shown on the right. Note that this graph is acyclic.



Use proof by contradiction to show that for every directed graph, application of Math Moose's procedure gives a directed acyclic graph.

Proof. Let $G = (V, E)$ be an arbitrary directed graph. Let $G' = (V, E')$ be the directed graph that remains after we remove all edges that connect two vertices in the same strongly-connected component.

The proof is by contradiction. Suppose that G' is not acyclic. Then it must have a cycle. Then let u and v be consecutive nodes in a directed cycle. There is a path from u to v consisting of a single edge. There also is a path from v to u, because the nodes are in a cycle. Since every edge in G' is also in G, these same two paths exist in the original graph G. Therefore, u and v are in the same strongly-connected component of G. But then the edges on the path from u to v should have been removed during the construction of G'. Since these edges were not removed, we have a contradiction. Therefore, G' is acyclic. $\blacksquare$

Problem 6 [5 points] **Graphs**

Let $G = (V, E)$ be a simple, undirected graph, and let C be a set of colors. Define a *partial coloring of* G *using* C to be a function that assigns to each node in V either a color in C or no color. (That is, it colors some, none, or all of the nodes using colors in C.) Define a *partly colored graph* to be a simple, undirected graph $G = (V, E)$ together with a partial coloring of G.

Prove the following fact about partly colored graphs:

> In a partly colored graph, any walk connecting a colored node to an uncolored node must include a colored node adjacent to an uncolored node.

Proof. Consider the path $p = v_1 v_2 \cdots v_k$, of which v_1 is colored and v_k is uncolored. Assume there is no edge (v_i, v_{i+1}) such that v_i is colored and v_{i+1} is uncolored. Then, by induction starting from v_1, v_i are colored for all i, so v_k is colored; this is a contradiction. ■

Problem 7 [15 points] **Algorithms**

Let $G = (V, E)$ be a simple, undirected graph. The following algorithm, *RedBlue*, manipulates partial colorings of G using $C = \{\text{red}, \text{blue}\}$. (Partial colorings are defined in the previous problem.)

> Initially, one vertex r of G is colored red, and all the other vertices are uncolored. At each step, one of two events occurs:
>
> a) Some uncolored vertex v that is adjacent to a red vertex becomes blue.
>
> b) Some uncolored vertex v that is adjacent to a blue vertex becomes red.
>
> If neither rule can be applied, the algorithm terminates.

(a) [3 points] Formalize this algorithm as a state machine; that is, define Q, Q_0, and δ.

A state q consists of a (total) function f from V to $\{red, blue, uncolored\}$. A start state is any state in which f maps exactly one $v \in V$ to red, and all others to uncolored. The transitions are all of the form:

$$
\begin{aligned}
(q, q') \in \delta \quad \iff \quad & \exists v \in V\ q.f(v) = \text{uncolored} \\
& \wedge \big[\, (q'.f(v) = \text{red} \wedge \exists w\ f(w) = \text{blue} \wedge (v, w) \in E) \\
& \quad \vee (q'.f(v) = \text{blue} \wedge \exists w\ q.f(w) = \text{red} \wedge (v, w) \in E) \,\big] \\
& \wedge \forall w \neq v\ q.f(w) = q'.f(w)
\end{aligned}
$$

(b) [3 points] Prove that the algorithm eventually terminates.

We describe a termination function. Define the value of the termination function to be the number d of uncolored vertices. Each step decreases d, so we can appeal to the Termination Theorem and conclude that the algorithm always terminates. Alternatively, we can expand the proof a bit more: since d starts at $n - 1$ and is always ≥ 0, it must eventually reach some minimum value (by well-ordering), whereupon the algorithm terminates.

(c) [3 points] Using the fact stated in Problem 6, prove that if G is connected, then all vertices are colored when the algorithm terminates.

Say the algorithm terminates with some vertex uncolored. Since the graph is connected, there is a path from v to some colored vertex w. By the given fact, there must be an edge along this path for which one endpoint x is colored and the other y is uncolored. We can apply one of the two steps above, assigning to y the opposite of x's color. Thus, the algorithm should not have terminated yet.

(d) [3 points] Consider applying this algorithm to *bipartite* graphs, i.e., graphs in which the vertices can be partitioned into two sets A and B such that there is no edge between any pair of vertices in A, and similarly for B. Suppose the node r that is colored red in the initial state is in set A. Using the Invariant Theorem, prove that in any reachable state, only vertices of A are colored red and only vertices of B are colored blue.

The Invariant Theorem requires that we show that the condition is true in all the start states, and that it is preserved by every step.

We need to show that the invariant is preserved from one step to the next, which we do by induction on the number of colored vertices.

$P(n) =$"After n vertices are colored, all red vertices are in A and all blue vertices are in B."

Base case. In any start state, there is exactly one red vertex in R—by construction—and no blue vertices. Therefore, $P(1)$.

Induction. Assume $P(n)$. Then, consider an execution in which $n+1$ vertices are colored. Consider the vertex i which was colored last. If we uncolor it, we can apply the induction hypothesis and conclude that all red vertices were in R and all blue vertices were in B.

Now we must prove that vertex i is colored correctly. Assume WLOG that vertex i is in B. For i to have been colored, it must have been the case that i was adjacent to a colored vertex; but all vertices adjacent to i are in R (by the definition of bipartite), so i must have been colored blue. This satisfies the invariant theorem, so $P(n+1)$.

Thus, $\forall n\ P(n)$, so all red vertices are in R and all blue vertices are in B.

(e) [3 points] Combine the previous parts to prove that when run on a connected bipartite graph, the above algorithm terminates and outputs a valid 2-coloring.

We proved the algorithm terminates on connected graphs with all vertices colored. By the invariant above, this means all vertices in R are red and all vertices in B are blue. By the definition of bipartite, there exists no pair of like-colored vertices connected by an edge, so this is a valid 2-coloring.