CS 70 Discrete Mathematics and Probability Theory
Spring 2015 Vazirani Midterm #1

PRINT your name: ____________________, ____________________
(last) (first)

SIGN your name: ________________________________

PRINT your student ID: ____________________

CIRCLE your exam room: 306 Soda 141 McCone 1 Pimentel

Name of the person sitting to your left: ____________________

Name of the person sitting to your right: ____________________

Please write your name and student ID on every page.

Please write your answers in the spaces provided in the test. We will not grade anything on the back of an exam page or outside the space provided for a question unless we are clearly told on the front of the page in the space provided for the question to look there.

You may consult one single-sided sheet of handwritten notes. Apart from that, you may not look at books, notes, etc. Calculators and computers are not permitted.

You have 120 minutes. There are 6 questions worth 20 points each, for a total of 120. Use the number of points as a rough guide for the amount of time to allocate to that question. Note that the last question is harder than the rest, so you should attempt it only after you take a good shot at the rest. Note also that many of the points are for proofs and justifications for your answers. Please make sure you spend the time to write clear, correct and concise justifications. Good luck!

Do not turn this page until your instructor tells you to do so.

Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Total/120

(Version C)

Name: ______________________________ SID: ______________

1. **Propositions and Quantifiers**

For each of these assertions say whether it is necessarily True or whether it could be False. Give a brief (couple of lines) justification for your answer, if true the main idea of the proof and if false then a counterexample. 4 points for each part — 1 for the correct answer and 3 for justification.

(a) $P \Rightarrow (\neg P \Rightarrow Q)$

True **False**

(b) $P \wedge Q$ and $\neg P \vee \neg Q$ cannot both be False.

True **False**

(c) $(\forall x\, A(x) \wedge \forall x B(x)) \implies \forall x\, (A(x) \wedge B(x))$

True **False**

(d) $(\forall x\, A(x) \vee \forall x\, B(x)) \implies \forall x\, (A(x) \vee B(x))$

True **False**

(e) $\forall x\, (A(x) \vee B(x)) \implies (\forall x\, A(x) \vee \forall x\, B(x))$

True **False**

2. **Induction Logic** For each of the following assertions below, circle the correct alternative: (A) it must always hold, or (N) it can never hold or (C) it can hold but need not always.[1] Give a short (couple of lines) justification for your answer. The domain of all quantifiers is the natural numbers. 2 points for each part — 1 point for correct answer and 1 point for justification.

   For the following parts assume that $P(n)$, $Q(n)$, and $R(n)$ are predicates on the natural numbers, and suppose $\forall k \in \mathbb{N}\ P(k) \Rightarrow Q(k+1)$, $\forall k \in \mathbb{N}\ Q(k) \Rightarrow R(k+1)$, and $\forall k \in \mathbb{N}\ R(k) \Rightarrow P(k+1)$.

   (a) If $(P(0) \wedge Q(0) \wedge R(0))$ is true then $\forall n (P(n) \wedge Q(n) \wedge R(n))$ is true.

   A **N** **C**

   (b) If $(P(0) \vee Q(0))$ is true then $\forall n (P(n) \vee Q(n))$ is true.

   A **N** **C**

   (c) If $(P(0) \vee Q(0) \vee R(0))$ is true then $\forall n\ P(n)$ is true.

   A **N** **C**

   (d) If $(P(101) \vee Q(101) \vee R(101))$ is true then $(\neg P(0))$ is true.

   A **N** **C**

[1] This is just as in HW2. Each part is of the form if P is true then Q is true. Assuming P is true, you must choose (A) if Q is necessarily true, (N) if Q is necessarily false and (C) otherwise.

(e) If $(\neg P(100) \vee \neg P(101) \vee \neg P(102))$ is true then $(\neg P(0) \vee \neg Q(0) \vee \neg R(0))$ is true.

A **N** **C**

(f) If $(\neg P(100) \wedge \neg P(101) \wedge \neg R(101))$ is true then $(\neg P(0) \wedge \neg Q(0) \wedge \neg R(0))$ is true.

A **N** **C**

For the remaining parts, assume we were trying to prove $P(n)$ is true for all $n \in \mathbb{N}$ by induction on n. Instead, we succeeded in proving $\forall k \in \mathbb{N}$ if $P(k)$ is true then $P(2k)$ is true.

(g) If $P(0)$ is true then $\forall n\ P(n)$ is true.

A **N** **C**

(h) If $P(1)$ is true then $\forall n\ P(2^n)$ is true.

A **N** **C**

(i) If $P(0)$ and $P(1)$ are true then $\forall n\ P(n)$ is true.

A **N** **C**

(j) If $P(5)$ is true then $(\forall n > 1\ P(5 \times 2^n))$ is true.

A **N** **C**

Name: ______________________________ SID: ______________

3. **Hypercubes**

   Prove that any cycle in an n-dimensional hypercube must have even length.

   Recall that a cycle is a closed (simple) path and its length is the number of vertices (edges) in it. The n dimensional hypercube is a graph whose vertex set is the set of n-bit strings, with an edge between vertices u, v iff they differ in exactly one bit (Hamming distance $= 1$).

Name: ______________________________ SID: ________________

4. **Stable Marriage**

Consider the traditional propose and reject stable marriage algorithm. On day j, let $P_j(M)$ be the rank of the woman that M proposes to (where the first woman on his list has rank 1 and the last has rank n). Also, let $R_j(W)$ be the total number of men that woman W has rejected up through day $j-1$ (i.e. not including the proposals on day j).

(a) Prove that $\sum_M P_j(M) - \sum_W R_j(W)$ is independent of j. What is its value?

(b) Conclude that one of the men or women must be matched to someone who is ranked in the top half of their preference list. You may assume that n is even.

5. **Counterfeit coins**

A bank has a pile of 3^n coins, one of which is counterfeit. The counterfeit coin looks identical to a regular coin in every way except that it is lighter. The bank has a pan balance, but not much time. Give a recursive algorithm that uses just n weighings to identify the counterfeit coin. Ideally your algorithm will consist of 3-4 lines of pseudocode. Each weighing involves placing some coins on each pan of the balance, resulting in one of three outcomes: left side heavier, or right side heavier, or two sides are the same weight. Prove by induction that your algorithm is correct and terminates in the claimed number of weighings.

Note that a majority of the points for this question are for proving the correctness of your algorithm and the bound on the number of weighings. However, you must give a recursive algorithm and failure to do so will result in 0 points for the entire question.

6. **Trees** (This question is harder than the rest. Attempt it only after you take a good shot at the rest.)

Show that the edges of a complete graph on n vertices for n even can be partitioned into $\frac{n}{2}$ edge disjoint spanning trees.

Recall that a complete graph is an undirected graph with an edge between every pair of vertices. The complete graph has $\frac{n(n-1)}{2}$ edges. A spanning tree is a tree on all n vertices — so it has $n-1$ edges. So the complete graph has enough edges (for n even) to create exactly $\frac{n}{2}$ edge disjoint spanning trees (i.e. each edge participates in exactly one spanning tree). You have to show that this is always possible.