C241 Fall 2009
Practice Exam

Important Note: This test is good practice for the final exam, but it is not, itself, the final exam. Topics we've covered in class (which might be on the real final but are not on this one) include, for example: formal logic proofs, set operations including the empty-set, proofs about properties of functions, equivalence relations, numerical induction on things other than summations, and tree traversals.

Answers will be posted later this week, but solutions to similar problems (or sometimes, the *same* problems) can be found in the solutions for previous assignments.

Logic (15 points)

1) Answer each of the following questions with "yes", "no", or "maybe". The predicates involved are: $C(x) =$"x has chocolate chips", $N(x) =$"x has walnuts", $E(z,x) =$"z ate x". The domain of x is cookies, and the domain of z is C241 students.

If it's true that $\forall x : C(x)$

This means that for every cookie C(x) is true. In other words, every cookie is chocolate chip.

a) Is there any cookie which is not chocolate chip? No

b) Does any cookie have nuts? Maybe (if it had nuts and chocolate chips)

If it's true that $\forall x, [\exists z : E(z,x)]$

This means that for each cookie, it's true that there exists a person who ate that cookie. So for every cookie there's someone who ate it.

c) Were all of the cookies eaten? Yes

d) Did one person eat all the cookies? Maybe (it might've been the same person who ate all of them. or it might've been one cookie per person. we don't know from the statement above)

If it's true that $\exists z, [\forall x : E(z,x)]$ This means that there is a person who, for every cookie, they ate that cookie. In other words, there exists a person who ate every cookie.

e) Did one person eat all the cookies? Yes. That's exactly what this says.

If it's true that $\forall x : [C(x) \leftrightarrow (\exists z : E(z,x))]$ This means that for every cookie, the cookie is chocolate chip if and only if there is someone who ate it. In other words, if a cookie was chocolate chip then there exists a person who ate it, *and*, if there exists a person who ate the cookie, then the cookie must have been chocolate chip.

f) Were all of the chocolate chip cookies eaten? Yes (for every chocolate chip cookie, there's someone who ate it. So they all got eaten)

g) If a cookie didn't have chocolate chips, did anyone eat it? No (if someone ate a cookie, then it was chocolate chip. So the only cookies that got eaten were chocolate chip).

2) Simplify, if possible, the following logical statement. (A list of the laws of logic is attached to the back of this exam):

$(w \wedge z \wedge x) \vee (w \wedge z \wedge \neg x) \vee (w \wedge \neg z \wedge x)$

$(w \wedge z \wedge x) \vee (w \wedge z \wedge \neg x) \vee (w \wedge \neg z \wedge x) = w \wedge ((z \wedge x) \vee (z \wedge \neg x) \vee (\neg z \wedge x)) = w \wedge ((z \wedge (x \vee \neg x)) \vee (\neg z \wedge x)) = w \wedge ((z \wedge (T)) \vee (\neg z \wedge x)) = w \wedge (z \vee (\neg z \wedge x)) = w \wedge ((z \vee \neg z) \wedge (z \vee x)) = w \wedge ((T) \wedge (z \vee x)) = w \wedge (z \vee x)$

Combinatorics (10 points)

2) As I'm creating an undirected graph with the vertices: $\{a, b, c, d, e, f\}$, I can pick any two different vertices and draw an edge between them (since it's an undirected graph, it won't matter which vertex I pick first). How many total different edges can I make in this graph? Explain briefly how you get your answer.

So the question is: how many ways can I pick two things from a set of six things (the six vertices a,b,c,d,e,f)? This is just 6 choose 2, or $\frac{6!}{4!2!}$. The total answer is $\frac{6!}{4!2!} = \frac{6\times5\times4!}{4!2} = \frac{6\times5}{2} = 15$

3) Idiot Sort is a very inefficient sorting algorithm. When given a set of n numbers, it just checks every possible arrangement of numbers until it encounters the arrangement that's in sorted order. In the worst case, how long will Idiot Sort take to sort the list?

The question here is how many different arrangements of the items might we have to go through before finding the sorted ordering of the list? Well, at worst, we'll have to go through all of them. So how many ways are there to arrange n items in order? $n!$

Sets, Functions, and Relations (15 points)

4) Say that P is the set of all prime integers, L is the set of integers less than 10, and T is the multiples of 3. Then list the elements of the set:

$(\mathbb{N} \cap L) \cap \overline{(P \cup T)}$.

So what we want here is the numbers which are: Natural numbers and less than 10 (so they're in $(\mathbb{N} \cap L)$) and *not* prime or multiples of 3 (so they're not in $(P \cup T)$). The natural numbers less than 10 are: 0,1,2,3,4,5,6,7,8,9. Of those 2,3,4,5,7 are prime, and 3,6,9 are multiples of 3. So our set should be: $\{0, 1, 4, 8\}$.

5) If $A = \{a, b, c\}$ and $B = \{1, 2, 3\}$, give an example of a function $F : A \to B$ which is neither injective nor surjective.

So remember that in order to be a function, each element of A needs to be paired with one element of B, and in order to be injective each element of B needs to have been paired with a different element of A (so we'll want to avoid that by mapping two elements of A to the same thing in B), and in order to be surjective each element of B must be mapped with some element of A (since we only have three things in A, if we pair two of them with the same thing in B, we'll end up leaving one thing in B unpaired, so we won't be surjective either). So, here's our function: $F = \{(a, 1), (b, 2), (c, 2)\}$.

6) If $A = \{a, b, c, d\}$, give an example of a relation $R \subseteq A \times A$ which is a partial ordering. Is your relation also a total ordering (justify your answer)?.

Remember that a partial order is reflexive, *anti-symmetric* and transitive. So an example of a partial ordering for this set would be:$\{(a, a), (b, b), (c, c), (d, d), (a, b), (b, c), (a, c)\}$. This example is not a total order, because it doesn't tell us how d relates to a, b, c.

Induction (20 points)

7) Use numerical induction to prove: $\sum_{i=1}^{n}(2i-1) = n^2$

Base Case (n = 1):

$$\sum_{i=1}^{1}(2i-1) =^? 1^2$$

$$2(1) - 1 =^? 1$$

$$1 = 1$$

Induction Hypothesis: Assume $\sum_{i=1}^{n}(2i-1) = n^2$

Induction Step: Show that this implies $\sum_{i=1}^{(n+1)}(2i-1) = (n+1)^2$.

$$\sum_{i=1}^{(n+1)}(2i-1) =^? (n+1)^2$$

$$\sum_{i=1}^{n}(2i-1) + (2(n+1)-1) =^? (n+1)^2 \text{ (expand summation)}$$

$$n^2 + (2(n+1)-1) =^? (n+1)^2 \text{ (Induction Hypothesis)}$$

$$n^2 + (2n+1) =^? (n+1)^2 \text{ (basic algebra)}$$

$$n^2 + 2n + 1 = n^2 + 2n + 1 \text{ (multiply out the right side)}$$

8) Use induction to prove that for all $w \in L$, $C(w)$ is an even number.

	L
1.	$\bullet \in L$
2a.	$u \in L \Rightarrow \lhd u \rhd \in L$
2b.	$u, v \in L \Rightarrow \lhd u \bowtie v \rhd \in L$

	$C : L \to \mathbb{N}$
1.	$C(\bullet) = 0$
2a.	$C(\lhd u \rhd) = 2 + C(u)$
2b.	$C(\lhd u \bowtie v \rhd) = 2 + C(u) + C(v)$

Note that the function C is just returning the number of triangles in the string.

Base Case: Show that $C(\bullet)$ is even. $C(\bullet) = 0$, and zero is even, so this is true.

Induction Hypothesis: Assume that $C(u)$ and $C(v)$ are both even.

Induction Step (2a): Show that $C(\lhd u \rhd)$ is even. Well, $C(\lhd u \rhd) = 2 + C(u)$, and $C(u)$ is even by our induction hypothesis, and an even number plus 2 is an even number. So yes, $C(\lhd u \rhd)$ is even.

Induction Step (2b): Show that $C(\lhd u \bowtie v \rhd)$ is even. Well, $C(\lhd u \bowtie v \rhd) = 2 + C(u) + C(v)$. And, $C(u)$ and $C(v)$ are both even by our induction hypothesis. So, since an even number plus an even number plus 2 is an even number... it's true that $2 + C(u) + C(v)$ will be even.

Time Complexity (20 points)

9) Find the asymptotic complexity (the closest fitting Big-O: $O(1)$, $O(log_2(n))$, $O(n\ log_2(n))$, $O(n^2)$**, etc...) for the following (you don't need to justify your answer):**

a) $n^3 + 3n^2$ This is $O(n^3)$

b) $2^n + n + 6$ Since 2^n grows much faster than n (or even n^k for any value of k), this is $O(2^n)$

c) $lg(n) + \frac{1}{n}$ Since the largest $\frac{1}{n}$ will be is 1, this will be $O(lg(n))$

d) This executes the arithmetic operation in the loop body (x = x+1) n times, so it's $O(n)$

```
for(i : 1 to n)
   x = x + 1
```

e) This one is kinda neat. On the first iteration of the outer loop $k = 1$, and the inner loop will execute only once. On the second iteration of the outer loop $k = 2$, and the inner loop will execute twice. This pattern continues until $k = n$ and the outer loop stops running. So, in total the inner loop will execute: $1 + 2 + 3 + ... + n$ times. This should look familiar! (if it doesn't yet, trust me it will before you're done with your degree). $1+2+3...+n = n(n+1)/2 = (1/2)(n^2+n)$. This loop will run in time $O(n^2)$.

```
for(k : 1 to n)
   for (j:1 to i)
      x = x + 1
```

10)Use the formal definition of Big-O to prove that: $5n^2 + n \in O(n^2)$

The formal definition of Big-O states that $f(n) \in O(g(n))$ if and only if $\exists N, C \in \mathbb{N}$ such that $f(n) \leq C \times g(n)$, $\forall n \geq N$. So if we want to show that $5n^2 + n \in O(n^2)$, we need to show that there are integers N, C such that $5n^2 + n \leq C(n^2)$ for all $n \geq N$. A good trick is to add up the coefficients on the left hand side and set C equal to that (so in this case we'll pick $C = 5+1 = 6$, and we'll pick $N = 1$. Now we prove that our choices are valid:

$5n^2 + n \leq^? 6(n^2)$

$5n^2 + n \leq^? 5(n^2) + n^2$

And we know that this inequality is true because it's clear that $5n^2 \leq 5n^2$, and that $n \leq n^2$, $\forall n \geq 1$.

Since we've found a valid choice for N and C, we satisfied the definition of Big-O, and we know that $f(n) \in O(g(n))$.

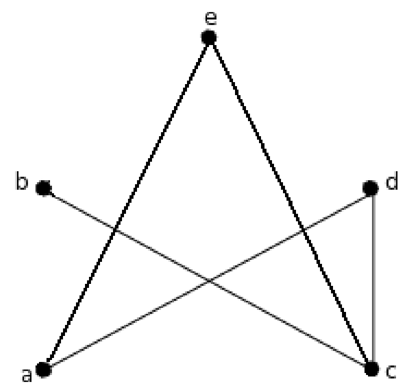
10)Use the formal definition of Big-O (and a proof by contradiction) to prove that: $n^2 \notin O(2n)$

To do this, we must prove that it is *not possible* to find *any* value of $C, N \in \mathbb{N}$ such that $n^2 \leq C(2n)$, $\forall n \geq N$. We'll do this by contradiction. Let's assume that $n^2 \in O(2n)$ and therefore assume that there is C, N such that $n^2 \leq C(2n)$, $\forall n \geq N$.

We can divide out an n and simplify this inequality to: $n \leq 2C$, $\forall n \geq N$.

And what's wrong with this statement? We've just claimed that there is a value of C such that n will always be less than $2C$, *for all values of* n *greater than* N. But C is just some constant number. So what about the values of n that are greater than $2C$? Then our inequality won't be true, obviously. So a valid value for C can't exist, we've found our contradiction, and it must be the case that $n^2 \notin O(2n)$.

Graphs (20 points)



11) Given a graph: $G = \{V, E\}$, an 'Independent Set' is a subset of vertices $V_I \subseteq V$ such that $[v_1, v_2 \in V_I] \Rightarrow \neg[\exists(v_1, v_2) \in E]$. In other words, V_I is a set of vertices which are *not* neighbors in G.

a) List all the vertices that are neighbors of e in the graph above.

a,c

b) Give an example of an independent set of size 2 in the graph above. Give an example of an independent set of size 3.

The rule for an independent set is $[v_1, v_2 \in V_I] \Rightarrow \neg[\exists(v_1, v_2) \in E]$. So if v_1, v_2 are two vertices in V_I, then there's can't exist an edge (v_1, v_2) between them in the graph's edge set E. So an independent set of size two could be $\{b, e\}$ (since there's no edge between them in the graph). And an independent set of size three could be $\{b, e, d\}$, since there's no edges connecting any of them in the graph.

c) Draw a graph with 4 vertices that has an independent set of size 4. Draw a graph with 4 vertices whose largest independent set has size 1.

If we've only got 4 vertices and we want them all to be in an independent set, then we can't have any edges between *any* of them in the graph (or else the two points which the edge connected couldn't be in the same independent set). So the first graph is just four separate vertices, with no edges.

If we want our largest independent set to be of size 1, then we can't have any independent sets of size 2 (or greater). Any time two vertices aren't directly connected by an edge in the graph, they can make an independent set of size 2. So in order to avoid that, all of our vertices must be directly connected to each other by edges. In other words, we must have a complete graph. See the solutions for HW 11 for more on complete graphs.

d) Give an example of a spanning tree for the graph above.

A spanning tree is a connected subgraph which has no cycles, but includes all the vertices of the graph (in other words, it's a subgraph with the same vertices as G, but with edges removed in order to eliminate cycles... but you can't remove edges that would break the graph into multiple pieces). If you remove the edge (d, c) in the graph above, the result is a spanning tree.

Bonus) The inverse of a graph $G = (V, E)$ is the graph $G' = (V, E')$, where $E' = (V \times V) \cap \overline{E}$. Draw the inverse for the graph above. Prove that if V_I is an independent set in G, then V_I will be a clique in G.

12) On the next page is the pseudo-code for Depth First Search on graphs, as well as an example graph. Use them to answer the following questions.

a)List out the vertices in the order they would be reached in a depth first search: a, b, c, d, e, f

b)Explain precisely in clear, detailed english how you could adapt the Depth First Search algorithm to return the number of vertices that are *reachable* from vertex a.

Ignore the Outer Loop. Just call Inner_DFS(a), and then when it's finished, count the number of vertices whose visited array value is 1.

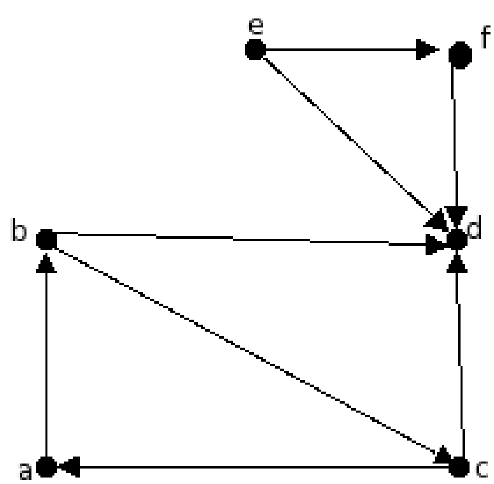
```
Outer_DFS(v)
{
    initialize_visited();    //the visited array keeps track of the
                             //vertices we've visited

    for each v in V;         //V is the set of all vertices in the graph
   {
          if not( visited[v] == 1 )
          { Inner_DFS(v); }
    }
}



Inner_DFS(v)
{
    visited[v] = 1;

    for each n in neighbors(v);//the neighbors of v are the vertices that
    {                          //have an edge leading to them from v
         if not(visited[n] == 1)
         { Inner_DFS(n);}
     }
 }
```



A Chart of the Algebraic Laws of Logic

(An extension of the chart on page 75 in your textbook)

Note: The symbols P,Q,R,K below can represent complex logical statements.

Double Negation	$\neg(\neg P)$	$\equiv$	P		
Identity Laws	$(P \vee F)$	$\equiv$	P		
	$(P \wedge T)$	$\equiv$	P		
	$(P \vee T)$	$\equiv$	T		
	$(P \wedge F)$	$\equiv$	F		
Complement Laws	$(P \vee \neg P)$	$\equiv$	T		
	$(P \wedge \neg P)$	$\equiv$	F		
Idempotence Laws	$(P \vee P)$	$\equiv$	P		
	$(P \wedge P)$	$\equiv$	P		
Commutative Laws	$(P \vee Q)$	$\equiv$	$(Q \vee P)$		
	$(P \wedge Q)$	$\equiv$	$(Q \wedge P)$		
Associative Laws	$(P \vee (Q \vee R))$	$\equiv$	$((P \vee Q) \vee R)$	$\equiv$	$(P \vee Q \vee R)$
	$(P \wedge (Q \wedge R))$	$\equiv$	$((P \wedge Q) \wedge R)$	$\equiv$	$(P \wedge Q \wedge R)$
DeMorgan's Laws	$\neg(P \vee Q)$	$\equiv$	$(\neg P \wedge \neg Q)$		
	$\neg(P \wedge Q)$	$\equiv$	$(\neg P \vee \neg Q)$		
Distributive Laws	$P \vee (K \wedge Q)$	$\equiv$	$(P \vee K) \wedge (P \vee Q)$		
	$P \wedge (K \vee Q)$	$\equiv$	$(P \wedge K) \vee (P \wedge Q)$		
Subsumption Laws	$P \vee (P \wedge Q)$	$\equiv$	P		
	$P \wedge (P \vee Q)$	$\equiv$	P		
Definition of Implication	$(P \rightarrow Q)$	$\equiv$	$(\neg P \vee Q)$		