C241 Fall 2008

Midterm II: Review Problems

(I) Induction: Prove $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

Base Case $n = 0$
$\sum_{i=0}^{0} i^2 = \frac{0(0+1)(20+1)}{6}$
$0^2 = 0$
$0 = 0$

Induction Hypothesis
Assume: $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

Induction Step
Show: $\sum_{i=0}^{n+1} i^2 = \frac{(n+1)((n+1)+1)(2(n+1)+1)}{6}$

$\sum_{i=0}^{n+1} i^2 = \frac{(n+1)((n+2)(2n+3)}{6}$	Simplify Right Hand Side
$\sum_{i=0}^{n} i^2 + (n+1)^2 = \frac{(n+1)((n+2)(2n+3)}{6}$	Pull last term off summation
$\frac{n(n+1)(2n+1)}{6} + (n+1)^2 = \frac{(n+1)((n+2)(2n+3)}{6}$	Use the induction hypothesis
$\frac{n(2n+1)}{6} + (n+1) = \frac{(n+2)(2n+3)}{6}$	Divide both sides by $(n+1)$
$\frac{n(2n+1)}{6} + \frac{6(n+1)}{6} = \frac{(n+2)(2n+3)}{6}$	Get a common denominator
$n(2n+1) + 6(n+1) = (n+2)(2n+3)$	Multiply both sides by 6
$2n^2 + n + 6n + 6 = 2n^2 + 3n + 4n + 6$	Multiply everything out
$2n^2 + 7n + 6 = 2n^2 + 7n + 6$	

(II) Big-O: Prove $10x^2 + 5x + 13 \in O(x^3)$

To prove that $10x^2 + 5x + 13 \in O(x^3)$, we must prove that there are two positive integers C and N such that $10x^2 + 5x + 13 \leq C(x^3)$ for all $x \geq N$.

A good way to prove this is to find an example of a C and an N which will make that statement true, and then show that they do, in fact, make it true.

We have three terms on the left hand side, and one term on the right hand side. If we make C a multiple of 3, it will make it easier to show that our choice of C is valid, since $3x^3 = x^3 + x^3 + x^3$. If C is a multiple of 3, we can expand the right hand side so it has as many terms as the left hand side, and then we can compare the two equations term by term.

But we're not done yet. It's not true that $x^3 > 10x^2$ for all x. We could choose $N = 11$, which would work because the statement is true for all $x \geq 11$. Or, we could choose a bigger C. The largest coefficient on the left hand side is 13, so let's choose $C = 14 \times 3$.

Now our inequality looks like: $10x^2 + 5x + 13 \leq 3 \times 14(x^3)$ for all $x \geq N$

This is true for all $x \geq 1$, so we can choose $N = 1$.

The final part of the proof (don't leave this out!) is to *show* that our choices are correct.

$10x^2 + 5x + 13 \leq 3 \times 14(x^3)$ for all $x \geq 1$

$10x^2 + 5x + 13 \leq 14(x^3) + 14(x^3) + 14(x^3)$ for all $x \geq 1$

Since we know that: $10x^2 \leq 14(x^3)$ for all $x \geq 1$

and $5x \leq 14(x^3)$ for all $x \geq 1$

and $13 \leq 14(x^3)$ for all $x \geq 1$

...this inequality is true, and we're done.

(III) Relations: Add the *minimum* number of edges to this relation to make is transitive, symmetric, and reflexive

$A = \{a, b, c, d\}$, $R : A \times A$, $R = \{(a, b), (a, c)\}$

In order to be reflexive, you must have an edge $(x, x) \in R$ for every element x in the domain A.... in other words, each point in the graph must have a self-loop. This relation doesn't have any edges of the form (x, x), so we need to fix that. Also, to be symmetric, we need to have an edge (x, y) for every edge (y, x). But in this relation we have (a, c) and no (c, a), and (a, b) with no (b, a). So we need to fix that too. Here's the fixed relation: $R = \{(a, a), (b, b), (c, c), (d, d), (a, b), (b, a), (a, c), (c, a)\}$.

Now to deal with transitivity. A relation is transitive if any time we have two edges that fit the pattern $(x, y), (y, z)$ we also have an edge (x, z). Now that we've made our relation symmetric, we have edges $(b, a), (a, c)$, but we don't have an edge (b, c). Similarly, we've got $(c, a)(a, b)$ but no edge (c, b). So let's fix that. $R = \{(a, a), (b, b), (c, c), (d, d), (a, b), (b, a), (a, c), (c, a), (b, c), (c, b)\}$

And now our relation finally has all three properties. You can check that it doesn't violate any property... every point has a self-loop (reflexive), every edge between two different points has an edge going back the other way (symmetry), and we haven't left out any edges that are required by the transitive property. Our final relation is: $\{(a, a), (b, b), (c, c), (d, d), (a, b), (b, a), (a, c), (c, a), (b, c), (c, b)\}$.

(IV) Program Verification: Verify that the following program is correct. (note: $y\ != 0$ is another way of writing $y \neq 0$).

```
{x = A, y = B}
while y != 0: {x + y = A + B}
       begin
       x := x + 1;
       y := y - 1;
       end
{x = A + B}
```

There are three logical statements in {} brackets in this program. The first one is the premise. The premise is a logical fact that we assume is true when the program begins. The second logical statement is the loop invariant. We'll prove that the loop invariant is true when the loop begins, and we'll also prove that if it's true at the beginning of a loop iteration, it's still true after the statements in the loop body have executed. Those two facts will tell us that the loop invariant is still true when the loop finally ends. The last logical statement is the conclusion. We need to prove that the conclusion is true when the program completes in order to "verify that the program is correct". To prove that the conclusion is true at the end of the program, we'll use the fact that the loop invariant is still true after the loop has ended (which we know because we just proved that to be the case), and also the fact that the loop condition ($y \neq 0$) is false (it must be false, or we would not have reached the end of the program).

There are three steps in a loop invariant proof:

(1) Prove that the loop invariant, $\{x + y = A + B\}$, is true at the beginning of the loop.

Since the premise tells us that $x = A, y = B$, we know that $x + y = A + B$. Easy.

(2) Prove that: If the loop invariant $\{x + y = A + B\}$ is true, it will remain true after we execute the two statements in the loop body. (If executing the statements in the loop body doesn't affect the truth of the loop invariant, we can run the loop for as many iterations as we like and the loop invariant will still be true when the loop finishes.)

Assume $\{x + y = A + B\}$. Assign the new values to x and y: $x' = x + 1$, $y' = y - 1$

Is the statement still true with the new values of x and y? Ie is $\{x' + y' = A + B\}$ true?

$\{x' + y' = A + B\}$?

$\{x + 1 + y - 1 = A + B\}$?

$\{x + y + 0 = A + B\}$?

$\{x + y = A + B\}$ This is the statement we assumed to be true, so we're done.

(3) Prove that the conclusion is true at the end of the program, using the fact that the loop invariant is still true after the loop ends, and the loop condition is false (since we've exited the loop).

So we know $\{x + y = A + B\}$ and $y = 0$. We're trying to show $\{x = A + B\}$.

This is not hard.

$x + y = A + B$ and $y = 0$

so $x + 0 = A + B$

so $x = A + B$. We're done.

(V) Performance Estimation: About how many steps does this program take to execute?

```
for i = 1:N
    begin
    x := x + 1;
    y := y - 1;
    for j = 1:M
        begin
        z := z + z;
        end
    for k = 1:P
        begin
        z := z - z;
        end
    end
```

We're not counting "begin" and "end" as computation steps, they're just textual markers to indicate where code blocks begin and end. All the steps we are counting have been assigned a number so we can refer to them more easily (this is a good idea if you want your solution to be clearly understandable).

We have three loops here. The outer loop runs N times, which means that every statement or code block inside of the outer loop will be executed N times. So we know right off that lines 1,2,3 will each be executed a total of N times. So add $N + N + N$ to our total number of computation steps.

Each time we enter the first inner loop, it will run M times. Since it's inside the outer loop, we'll be entering it N times over the course of the program. So every line of code in the first inner loop will run a total of $M \times N$ times. There's two lines of code in this loop: lines 4 and 5. So add $MN + MN$ to our total.

Each time we enter the second inner loop, it will run P times. Since it's inside the outer loop, we'll be entering it N times over the course of the program. So every line of code in the second inner loop will run a total of $P \times N$ times. There's two lines of code in this loop: lines 6 and 7. So we add $PN + PN$ to our total.

So, in total, we estimate this program takes about $N + N + N + MN + MN + PN + PN = 3N + 2MN + 2PN = N(3 + 2M + 2P)$ steps to execute fully. In other words, since $N, M, andP$ are variables, its time complexity is $O(N(M + P))$.

(VI) Countability: If A, B, C are countable sets (possibly infinite), prove that $A \cup B \cup C$ is also countable.

Going back to our definition of countability, if a set is countable that means that we can list out all of its elements in a long line, so there's a first element, a second element, and so on. This is equivalent to pairing each element of the set with a natural number (the first element gets paired with 1, the second gets paired with 2, and so on...). If we can map a set to the natural numbers, then it's countable.

So, since A is countable, we can list out its elements in order; a_1, a_2, a_3.....

Same thing with B: b_1, b_2, b_3...

And C: c_1, c_2, c_3...

The goal is to find an order to list out all the elements of $A \cup B \cup C$.

It's not completely trivial. The order $a_1, a_2, a_3 .... b_1, b_2 ... c_1, c_2$... won't work because if A is an infinite set, we'll never reach an element of B, and we'll certainly never reach C.

The trick here is to make progress on all three sets at once, by alternating elements:

$A \cup B \cup C$: $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3....$ In this way we'll be able to assign every element in $A \cup B \cup C$ a definite in the ordering, so we'll be able to 'count' them all.

What if an element is in both A and B? We don't want to list any element twice. So we'll just make a note that when we're listing out the elements in this order, we'll skip over any elements that have already appeared previously in the list. And that's it.

(VII) Induction: Prove that for all $n \geq 0$, $t_n = a + \frac{n^2+n}{2}$, given the following recursive definition of t_i:

$$t_0 = a$$

$$\vdots$$

$$t_{k+1} = t_k + k + 1$$

Base Case $n = 0$

$t_0 = a + \frac{0^2+0}{2}$	Just prove the statement is true at $n = 0$
$a = a + 0$	Use the function definition to get the value for t_0
$a = a$	

Induction Hypothesis

Assume: $t_n = a + \frac{n^2+n}{2}$	Assume the statement is true at n

Induction Step

Show: $t_{n+1} = a + \frac{(n+1)^2+(n+1)}{2}$	Show it's true at $n+1$: replace n with $(n+1)$
$t_n + n + 1 = a + \frac{(n+1)^2+(n+1)}{2}$	Use the function definition to expand t_{n+1}
$(a + \frac{n^2+n}{2}) + n + 1 = a + \frac{(n+1)^2+(n+1)}{2}$	Use the induction hypothesis to replace t_n
$\frac{n^2+n}{2} + n + 1 = \frac{(n+1)^2+(n+1)}{2}$	Subtract a from both sides
$\frac{n^2+n}{2} + \frac{2(n+1)}{2} = \frac{(n+1)^2+(n+1)}{2}$	Get a common denominator
$\frac{n^2+n+2(n+1)}{2} = \frac{(n+1)^2+(n+1)}{2}$	Combine fractions on left side
$n^2 + n + 2(n+1) = (n+1)^2 + (n+1)$	Multiply by 2 on both sides
$n^2 + 3n + 2 = n^2 + 2n + 1 + n + 1$	Multiply things out
$n^2 + 3n + 2 = n^2 + 3n + 2$	Simplify

(VIII) Big-O: If $f_1 \in O(g_1)$ and $f_2 \in O(g_2)$, prove that $f_1 + f_2 \in O(g_1 + g_2)$

Using the definition of big-O, we know that there exists some C_1, N_1 such that $f_1(x) \leq C_1 g_1(x)$ for all $x \geq N_1$, and also there is some C_2, N_2 such that $f_2(x) \leq C_2 g_2(x)$ for all $x \geq N_2$. This will be useful information.

To show that $f_1 + f_2 \in O(g_1 + g_2)$ we need to show that there is some C_3, N_3 such that $f_1(x) + f_2(x) \leq C_3(g_1(x) + g_2(x))$ for all $x \geq N_3$.

Let's distribute that C_3 to get a better idea what this inequality is asking: $f_1(x) + f_2(x) \leq C_3 g_1(x) + C_3 g_2(x)$

If we choose C_3 that's bigger than both C_1 and C_2, in other words $C_3 \geq max\{C_1, C_2\}$, then we have $f_1(x) \leq C_3 g_1(x)$ and $f_2(x) \leq C_3 g_2(x)$. So the inequality $f_1(x) + f_2(x) \leq C_3 g_1(x) + C_3 g_2(x)$ will be true.

We also need to choose N_3. Since the two inequalities $f_1(x) \leq C_1 g_1(x)$, $f_2(x) \leq C_2 g_2(x)$ are only valid for $x \geq N_1$ and $x \geq N_2$ respectively, we better pick $N_3 \geq max\{N_1, N_2\}$.

We've shown that we can find values for C_3, N_3 that will make the inequality $f_1(x) + f_2(x) \leq C_3 g_1(x) + C_3 g_2(x)$ hold true for all $x \geq N_3$, so we've proven that $f_1 + f_2 \in O(g_1 + g_2)$.

(IX) Big-O: Prove that $x^2 \notin O(2x)$

Ok. Again we need to look at the definition of big-O. If $x^2 \notin O(2x)$, that means that there does exist *any* C, N such that $x^2 \leq C(2x)$ for all $x \geq N$.

This isn't hard to show. Remember that the inequality needs to hold for *all* $x \geq N$. What happens when $x > 3C$? Then, regardless of what C is, our inequality looks like: $(3C)^2 \leq C(6C)$, or simplified: $9C^2 \leq 6C^2$. There's no way this inequality can ever be true... so there is no C such that $x^2 \leq C(2x)$ for *all* $x \geq N$.... when x gets large enough, the C will fail, no matter how large C is. Thus $x^2 \notin O(2x)$.

(X) Induction: Prove that for all $n \geq 1$, $12^n + 2(5^{n-1})$ is evenly divisible by 7.

We can restate this as: $12^n + 2(5^{n-1}) = 7q$ for some integer q.

<u>Base Case</u> $n = 1$	(since the statement holds only for $n \geq 1$)
$12^1 + 2(5^{1-1}) = 7q$	
$12 + 2(5^0) = 7q$	
$12 + 2(1) = 7q$	
$14 = 7q \; q = 2$	we've shown that q that is an integer, so we're done.

<u>Induction Hypothesis (IH)</u>

Assume that for some integer q:

$12^n + 2(5^{n-1}) = 7q$

<u>Induction Step</u>

Show that for some integer q':

$12^{n+1} + 2(5^{(n+1)-1}) = 7q'$	(q' will be a different value than q in our IH.
$12^{n+1} + 2(5^n) = 7q'$	$1 - 1 = 0$
$12(12^n) + 2(5(5^{n-1})) = 7q'$	expand exponents, to get closer to our IH.
$(5+7)(12^n) + 5 \times 2(5^{n-1}) = 7q'$	we have more 12^n than 5^{n-1}, so expand: $12 = 5 + 7$
$5(12^n) + 7(12^n) + 5 \times 2(5^{n-1}) = 7q'$	distribute 12^n. getting closer...
$7(12^n) + 5(12^n) + 5 \times 2(5^{n-1}) = 7q'$	rearrange terms
$7(12^n) + 5((12^n) + 2(5^{n-1})) = 7q'$	factor out the 5. now, we can use our IH
$7(12^n) + 5(7q) = 7q'$	Use the induction hypothesis
$12^n + 5q = q'$	Divide by 7. We've shown that q' is an integer, we're done.

(XI) Induction: Prove that for all integers $n \geq 2$, $\sum_{i=0}^{n-1}(2i)^2 = \frac{(2n)!}{3!(2n-3)!}$

Base Case $n = 2$	(since $n \geq 2$)
$\sum_{i=0}^{2-1}(2i)^2 = \frac{(2\times 2)!}{3!(2\times 2-3)!}$	
$\sum_{i=0}^{1}(2i)^2 = \frac{(4)!}{3!(1)!}$	
$(2\times 0)^2 + (2\times 1)^2 = \frac{(4)!}{3!(1)!}$	
$0 + 4 = \frac{4\times 3\times 2\times 1}{3\times 2\times 1\times 1}$	
$4 = 4$	

Induction Hypothesis

Assume: $\sum_{i=0}^{n-1}(2i)^2 = \frac{(2n)!}{3!(2n-3)!}$

Induction Step

Show: $\sum_{i=0}^{(n+1)-1}(2i)^2 = \frac{(2(n+1))!}{3!(2(n+1)-3)!}$	
$\sum_{i=0}^{n}(2i)^2 = \frac{(2n+2))!}{3!(2n+2-3)!}$	simplify
$\sum_{i=0}^{n-1}(2i)^2 + (2n)^2 = \frac{(2n+2))!}{3!(2n-1)!}$	pull last term off summation, simplify right side
$\frac{(2n)!}{3!(2n-3)!} + (2n)^2 = \frac{(2n+2))!}{3!(2n-1)!}$	use Induction Hypothesis
$\frac{(2n\times(2n-1)\times(2n-2)\times(2n-3)!}{3!(2n-3)!} + (2n)^2 = \frac{(2n+2))!}{3!(2n-1)!}$	pull out first terms of larger factorial, so we can simplify fractions
$\frac{(2n(2n-1)(2n-2)}{3!} + (2n)^2 = \frac{(2n+2))!}{3!(2n-1)!}$	simplify fraction
$\frac{(2n(2n-1)(2n-2)}{3!} = \frac{(2n+2)\times(2n+1)\times(2n)\times(2n-1)!}{3!(2n-1)!}$	expand factoral
$\frac{(2n(2n-1)(2n-2)}{3!} + (2n)^2 = \frac{(2n+2)(2n+1)(2n)}{3!}$	simplify fraction
$\frac{(2n(2n-1)(2n-2)}{6} + (2n)^2 = \frac{(2n+2)(2n+1)(2n)}{6}$	$3! = 6$
$\frac{(2n-1)(2n-2)}{6} + (2n) = \frac{(2n+2)(2n+1)}{6}$	cancel $2n$ from all terms
$\frac{(2n-1)(2n-2)}{6} + \frac{6(2n)}{6} = \frac{(2n+2)(2n+1)}{6}$	get common denominator
$(2n-1)(2n-2) + 6(2n) = (2n+2)(2n+1)$	mulitply by 6
$4n^2 - 4n - 2n + 2 + 12n = 4n^2 + 2n + 4n + 2$	mulitply everything out
$4n^2 + 6n + 2 = 4n^2 + 6n + 2$	simplify, and we're done

(XII) Countability: If A is an infinite countable set, prove that the set of all two-letter words in A^* is also countable.

Remember that if A, B are countable sets, $A \times B$ is countable as well. In other words, the set of all possible pairs of two elements that we can create where the first element comes from an infinite countable set and the second element comes from an infinite countable set, is countable.

We're doing the same thing here, except instead of picking the second element from B, we're drawing it from A (which is still an infinite countable set), and instead of writing ordered pairs (a_i, a_j) we're writing words: $a_i a_j$.

So our proof will be essentially the same as our proof for $A \times B$. We begin by arranging the elements of our set in a reasonable way. With two elements to find, we're going to expand along two dimensions when we try to list out all the elements. Remember that since A is countable we can list out its elements as: $a_1, a_2, a_3 ...$:

$$\begin{array}{lllll}
a_1a_1 & a_1a_2 & a_1a_3 & a_1a_4 & ... \\
a_2a_1 & a_2a_2 & a_2a_3 & a_2a_4 & ... \\
a_3a_1 & a_3a_2 & a_3a_3 & ... & \\
a_4a_1 & a_4a_2 & ... & & \\
a_5a_1 & ... & & & \\
\vdots & & & &
\end{array}$$

Now we order the elements in this square along an expanding, zig-zagging diagonal line which will eventually hit all the elements in the square:
$a_1a_1, a_1a_2, a_2a_1, a_2a_2, a_1a_3, a_1a_4, a_2a_3, a_3a_2, a_4a_1, a_5a_1, a_4a_2, a_3a_3, a_2a_4 ....$

Since we've found a way to order out all of the elements in our set, we've proved that it is countable.