REVIEW FOR FINAL EXAM

The content of this course has been roughly:

```
    34 lectures
=   9  on Object Oriented Design and Java (3-6,16-19,30)
  + 17 on data structures and algorithms (7-9,20-29,31-34)
  + 8  mathematical issues such as number representations (1-2)
       recursion and recurrences (10-11), runtime analysis (12-15)
```

The assignments already *extensively* covered OO Design and Java, so I will not examine you on those topics for the final exam. (You can ignore the OOD and Java lecture listed above.) Rather, I will examine you on data structures, algorithms, and mathematical issues such as number representations, recursion and recurrences, and runtime analysis, i.e. all the other lectures.

The final exam will attempt to test your understanding, as opposed to your ability to memorize. However, you still do need to memorize many things (in order to understand them). For example, you need to memorize the definitions of big O. And you need to memorize how different traversals work: pre-,post-,level-, depth first, breadth first.

Today's lecture

Today I will cover a few issues that T.A.s and I feel are giving the most trouble, and that are very important for future CS courses you will take.

Big O

First, you need to make sure you know the definition of what it means to say "f(n) is O(g(n))" and "f(n) is not O(g(n))":

- f(n) is O(g(n)) means that there exists a c and n_0 such that for all $n >= n_0$, $f(n) <= cg(n)$
- f(n) is not O(g(n)) means that for every c and n_0, there exists an $n > n_0$ such that $f(n) > cg(n)$.

Comment 1: where does "f(n) is not O(g(n))" come from ?

"f(n) is O(g(n))" says that there exists a c and n_0 such that some statement `S` is true, namely `S` $\equiv$ "for all $n >= n_0$, $f(n) <= cg(n)$". If we say that "f(n) is NOT O(g(n))" then we saying that the statement "f(n) is O(g(n))" is false. This means we are saying there *does not* exist c and n_0 such that `S` is true or, in other words, for any c and n_0, the statement `S` is false. So, what does it mean to say that the statement `S` is false? It means that it is not true that "for all $n >= n_0$, $f(n) <= cg(n)$" or, in other words, there must exist some n such that $n >= n_0$ and $f(n) > cg(n)$. Note that this is precisely the definition of "f(n) is NOT O(g(n))" !

Comment 2: Taking the limit

You cannot prove "f(n) is O(g(n))" by showing that the limit $f(n)/g(n)$ exists and taking c to be this limit.

For example: if you are asked to prove $5n^2 + 3n + 4$ is $O(n^2)$ then you might be tempted to say:

```
‘‘I want show that 5n^2 + 3n + 4 <= c n^2 for n sufficiently large.  So, dividing
both sides by n^2 and taking the limit as n goes to infinity, I get 5 <= c.
So take c=5 and I am done."
```

This is incorrect reasoning. First, notice that $5n^2 + 3n + 4 <= 5n^2$ is never true for positive n, so $c = 5$ does not do the job. You need a bigger c. Second, when you do choose the bigger c, you need to find an n_0 such that $5n^2 + 3n + 4 <= cn^2$ for all $n \geq n_0$. Sometimes $n_0 = 1$ does the job, but sometimes it doesn't and you will need to choose a larger n_0.

Comment 3: how not to write proofs

When writing proofs, many of you leave out important quantifiers. For example, some of you write "proofs" like this: (I have written line numbers to the right for the discussion below.)

```
Claim:   n+2 is O(n)           (1)
Proof:   n+2 <= cn             (2)
         n+2 <= 2n             (3)
           2 <= n              (4)
```

Someone reading this proof has no idea what you are assuming and what you are trying to prove. Look at what I just wrote more closely.

(1) is fine.

(2) is not fine. It doesn't say whether this is something that is given, or whether this is something you are trying to prove. Even if you say that this is what you are trying to prove, you also need to give quantifiers for c and n. You want prove that this statement holds *for some c and* n_0 (there exists a c, n_0) and *for all* $n >= n_0$.

(3) is not fine. Even if you correct line (2) as I just discussed, line (3) only follows from line (2) if you say explicitly that c=2.

(4) is not fine. What is the relationship between (3) and (4)? Does (4) follow from (3)? Does (3) follow from (4)? Are (3) and (4) equivalent ?

Here is a more complete proof:

```
Claim:   n+2 is O(n)
Proof:  We want to show that there exists c and n_0 such that
         n+2 <= cn  for all n >= n_0.   [SAY WHAT YOU WANT TO DO.]

   Let’s try c = 2.

            n+2 <= 2n
      <=>   2 <= n      [ASIDE:  the symbol  <=> means "if and only if"]
      Thus, we can take n_0 = 2.
```

Here is another example of an incorrect proof (of which we have seen alot).

```
Claim:   n+2 is not O(1)
Proof:   n+2 > c
         which is true for n sufficiently large.  And I'm done...
```

No, you're not done. This proof is not correct, since it doesn't show that the definition of "f(n) is not O(g(n))" holds. Here is a correct proof:

```
Claim:   n+2 is not O(1)
Proof:
   We want to show that, for any c,n0, there exists
   an n >= n0 such that   n+2 >= c.     [This states the definition
   and says what you are trying to prove.]

Take any c, and take any n0.    Choose n = max(c-2,n0),
So, n >= n0 and n >= c-2, as required.
```

Note that when I say "take any c and n0", I don't mean that you should plug numbers in there e.g. c=5 and n0 = 8. Rather it means that *for any* numbers you plug in, you can find an n such that the definition holds.
YOU SHOULD SEE ASSIGNMENT 2 SOLUTIONS FOR MORE CHALLENGING EXAMPLES.

PROOFS BY INDUCTION

Here is another example of a proof which one student emailed me:

```
Claim:   n^n >= n!

Proof  (by induction):
It is obviously true for n = 1 because 1 >= 1.
Assume true for n, we'll prove for n+1:

 (n+1)^(n+1) >= (n+1)!                                   (1)
 ((n+1)^(n)) * (n + 1) >= n! * (n + 1)                   (2)
 (n+1)^n >= n!                                           (3)
 But, (n+1)^n >= n^n >= n! by assumption.                (4)
 Therefore, n^n >= n!                                    (5)
```

I had great difficulty understanding this proof. Why? One problem is that the student is starting out in the first line with the inequality that he is trying to prove. He then proceeds line by line, where each line is true if and only if the previous line is true. *But this "if and only if" relationship is not stated explicitly. I had to figure it out!*

Another problem: look at the relationship between lines line 4 and line 5. Here the relationship is not "if and only if". (But again, I am supposed to figure this out.) Line 4 says that `n^n >= n!` *by assumption*, and line 5 *concludes* that `n^n >= n!` Hmm.. confusing, no?

Compare that proof to the following:

```
Claim:   n^n >= n!  for all n >= 0.          Note the "for all" quantifier.

Proof  (by induction):
It is obviously true for n = 1 because 1 >= 1.
Assume true for some arbitrarily chose n, and prove it for n+1.
```

[ASIDE: normally one substitutes n=k to avoid confusing the "for all n" in the claim for the arbitrarily chosen "n" of the proof. But I will just stick with the n in this example to compare it to the example I discussed above. Anyhow, continuing on with the proof...]

```
                    n^n >= n!   (induction hypothesis)
  ==>           (n+1)^n >  n!     since n+1 > n
  ==>     (n+1)*(n+1)^n  >  (n+1)* n!    i.e. multiplied both sides by (n+1)
  ==>        (n+1)^(n+1) >  (n+1)!   which is what we wanted to prove.
```

Summary

When you do a proof:

- begin with what you know (or what you are explicitly assuming) and derive what you are trying to show *from* what one know/assume. Try to use `p ==> q` or `p <==> q` arguments, not `p <== q` arguments. (If you must use the latter, then make sure you indicate which way the implication goes. Otherwise the reader will get confused.)
- Make sure you quantify your variables: "for all n" is not the same thing as "there exists an n". If you don't include a quantifier, then the reader may not know what you mean.

One final point: do I need to use induction ?

Do you need to prove the above claim using induction? No. Here is another way to prove it.

```
Claim:   n^n >= n!  for all n >= 0.

Proof:

If I want to show the n^n >= n! for all n, then it is sufficient
to show that n! / n^n  <= 1  for all n.

  n! / n^n  =  (n/n) * (n-1)/n * (n-2)/n * .... 3/n * 2/n * 1/n

The right side is the product of n terms, each of which is greater
than 0 and less than or equal to 1.   Such a product has to be between
0 and 1, since the product of any two or more numbers between 0 and 1 is
between 0 and 1.  (You could prove this last statement by induction if
you want to.)
```