A few more observations about Big O

Last lecture, we introduced the formal definition of big O and gave a number of examples. Here I will elaborate on some of the ideas from last lecture about big O, and then we will move on to two other asymptotic bounds, namely big Omega and big Theta.

Suppose we have two functions $t(n)$ and $g(n)$ such that $t(n)$ is $O(g(n))$, in particular, $t(n) \leq cg(n)$ for all $n \geq n_0$. Now suppose that $g(n) \leq h(n)$ for all $n \geq n_0$. Then obviously, $t(n) \leq h(n)$ for the same range of n, and so $t(n)$ is also $O(h(n))$.

In particular, for sufficiently big n, each of the following strict inequalities hold:

$$1 < \log_2 n < n < n \log_2 n < n^2 < n^3 < ... < 2^n < n! < n^n$$

which I will leave for you to verify. Note that the n_0 may be different from each of the inequalities. It follows from this claim that if $t(n)$ is $O(g(n))$ for one of the functions listed above, then it will also be $O(h(n))$ for any of the asymptotically larger valued functions.

Generally when we talk about $O()$ of some function $t(n)$, we use the "tightest" (smallest) upper bound we can. For example, if we observe that a function is $O(\log_2 n)$, then generally we would not say that that the function is $O(n)$, even though technicall (according to the above argument) the function *would* be $O(n)$, as well as $O(n^2)$, etc.

A related point is that we do not use constants when discussing the big O properties of a function. One does not writes $O(3n)$ or $O(\frac{n(n-1)}{2})$. Rather, the $g(n)$ that you put inside the brackets is always as simple as possible, e.g. typically one of the functions listed in the inequalities above.

Another related point is that, for a given simple $g(n)$ such as listed in the inequalities above, there are many functions that $O(g(n))$. We have been saying that some function $t(n)$ "is" $O(g(n))$. Sometimes people also say that $t(n)$ "is a member of the set of functions that are" $O(g(n))$, or more simply people say that $t(n)$ "belongs to" $O(g(n))$. In set notation, one writes "$t(n) \in O(g(n))$" where $\in$ is notation for set membership. (If you have not seen set notation before, please email me. I am assuming everyone has seen it.) With this notation in mind, and thinking of various $O(g(n))$ as sets of functions, the discussion in the paragraphs above implies that we have strict containment relations on these sets:

$$O(1) \subset O(\log n) \subset O(n) \subset O(n \log n) \subset O(n^2) \ \cdots \subset O(2^n) \subset O(n!) \ldots$$

I will occasionally use this notation in the course, although usually I'll just say "$t(n)$ is $O(g(n))$" rather than "$t(n) \in O(g(n))$."

Big Omega (asymptotic lower bound)

With big O, we defined an asymptotic upper bound. There is a similar definition for an asymptotic lower bound called "big Omega". We begin by defining $t(n)$ to be asymptotically bounded *below* by $g(n)$ if there exists an n_0 such that, $t(n) \geq g(n)$ for all $n \geq n_0$. As with big O, we would like our $g(n)$ to be a simple as possible, though, and so we tweak the definition slightly. This allows the expression for $g(n)$ not to have any constant factor, and instead a constant factor c becomes part of the defintion, just like what we did last lecture with big O.

Definition (big Omega): We say that $t(n)$ is $\Omega(g(n))$ if there exists positive constants n_0 and c such that, for all $n \geq n_0$,

$$t(n) \ \geq \ c \ g(n).$$

The idea is that $t(n)$ grows at least as fast as $g(n)$ times some constant, for sufficiently large n. Note that the only difference between the definition of $O()$ and $\Omega()$ is the $\leq$ vs. $\geq$ inequality.

Example

Consider $t(n) = \frac{n(n-1)}{2}$. This function is less than $\frac{n^2}{2}$ for all n, so since we want a *lower* bound we need to choose a smaller c than $\frac{1}{2}$. Let's try something smaller, namely $c = \frac{1}{4}$.

$$\begin{array}{rrcl} & \frac{n(n-1)}{2} & \geq & \frac{n^2}{4} \\ \iff & 2n(n-1) & \geq & n^2 \\ \iff & n^2 & \geq & 2n \\ \iff & n & \geq & 2 \end{array}$$

So, we can use $c = \frac{1}{4}$ and $n_0 = 2$. Notice the $\iff$ symbols that connect statements in the different lines. These are "if and only if" symbols, meaning that the statements being compared have the same truth value: for a given n, either all the statements are true or all are false. In particular, if $n \geq 2$ (last statment) then the first inequality must be true also, which is what we wanted to show.

Are these the only constants we can use? No. Let's try $c = \frac{1}{3}$.

$$\begin{array}{rrcl} & \frac{n(n-1)}{2} & \geq & \frac{n^2}{3} \\ \iff & \frac{3}{2}n(n-1) & \geq & n^2 \\ \iff & \frac{1}{2}n^2 & \geq & \frac{3}{2}n \\ \iff & n & \geq & 3 \end{array}$$

So, we can use $c = \frac{1}{3}$ and $n_0 = 3$.

Earlier in the lecture I mentioned that it is common to let $O(g(n))$ stand for a set of functions. We can do the same with $\Omega(g(n))$, of course. Notice, however, that when we talk about lower bounds, the relationship between the sets changes. If $t(n) \in \Omega(g(n))$, and if $g(n) \in O(h(n))$ then $t(n) \in \Omega(h(n))$ also. This gives us the following strict subset relationships:

$$\cdots \subset \Omega(n^2) \subset \Omega(n \log n) \subset \Omega(n) \subset \Omega(\sqrt{n}) \subset \Omega(\log n) \subset \Omega(1)$$

That is, $\Omega(1)$ is the largest of the sets shown. It is the set of functions that are asymptotically bounded below by a positive constant.

(For those of you who have done relatively little math beyond high school, I realize that you may be getting lost in the mathematical abstraction here. Please don't fret. Once you get through MATH 240 (or MATH 235 or MATH 363), you should be more comfortable with this sort of material. The idea here really is quite simple. Basically, if I am taller than someone who is 4 feet tall, then I must be taller than someone who is 3 feet tall also, since a 4 foot tall person is taller than the 3 foot tall person. The rest is just "bells and whistles" having to do with asymptotic behavior.)

Big Theta

It often happens that a function $t(n)$ is both $O(g(n))$ and $\Omega(g(n))$. For example, $t(n) = \frac{n(n+1)}{2}$ is both $O(n^2)$ and $\Omega(n^2)$. In this case, we say that $t(n)$ is $\Theta(g(n))$.

Definition (big theta): We say that $t(n)$ is $\Theta(g(n))$ if there exist three positive constants n_0 and c_1 and c_2 such that, for all $n \geq n_0$,

$$c_1 \; g(n) \;\leq\; t(n) \;\leq\; c_2 \; g(n).$$

Obviously, we would need $c_1 \leq c_2$ for this to be possible.

When we discuss the performance of algorithms in this, the functions $t(n)$ will also have the same Ø( ) and $\Omega($ $)$ behavior, and so we could characterize this behavior as $\Theta($ $)$. Why don't we do so? The answer is that typically when we are discussing the asymptotic bounds, we are interested in upper bounds because we are concerned with algorithms taking too long. Binary search is better than linear search because binary search is $O(\log_2 n)$ whereas linear search is $O(n)$. It is less clear why we would want to make the point that binary search is $\Omega(\log_2 n)$.

There will be times in future courses where you'll see that the performance of some algorithms has a particular $\Omega($ $)$ bound. But not in COMP 250. Nonethesis, while we are learning about big O, it is good to be aware of big Omega and big Theta too.

Finally, at the end of the lecture I discussed a few other aspects of big Theta. I will try to add these into the Exercises.