8. Data Structures and Algorithms :: Complexity without Proof

Once you become familiar with the concept of orders of growth and Big O notation it becomes easy in many cases to state the asymptotic complexity of an algorithm without rigorously proving it.

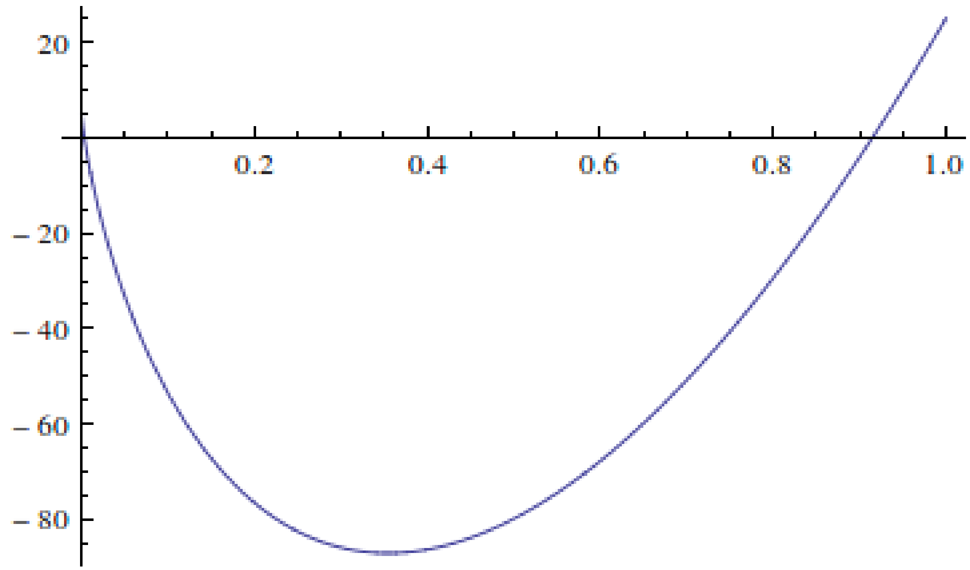
Remember that if $f(n)$ is $O(g(n))$ then $C \cdot g(n)$ serves as an upper bound on $f(n)$ when n exceeds some value n_0. The key to easily recognizing the order of growth of $f(n)$ is to understand that the asymptotic behavior of $f(n)$ **is determined by the fastest growing term**, i.e., the **term with the largest order of growth**.

For example, suppose we analyze algorithm A and determine that the number of times the key operation is performed is $f(n) = 3n + 5 + 17n^3 - 177\ n\ lg\ n$. What is the asymptotic complexity of *A?*

To answer this question we would have to determine $g(n)$, C, and n_0 and then prove that $|f(n)| \le C \cdot g(n)$ for all $n > n_0$. The question then becomes: what $g(n)$ do we choose?

8. Data Structures and Algorithms :: Complexity without Proof (continued)

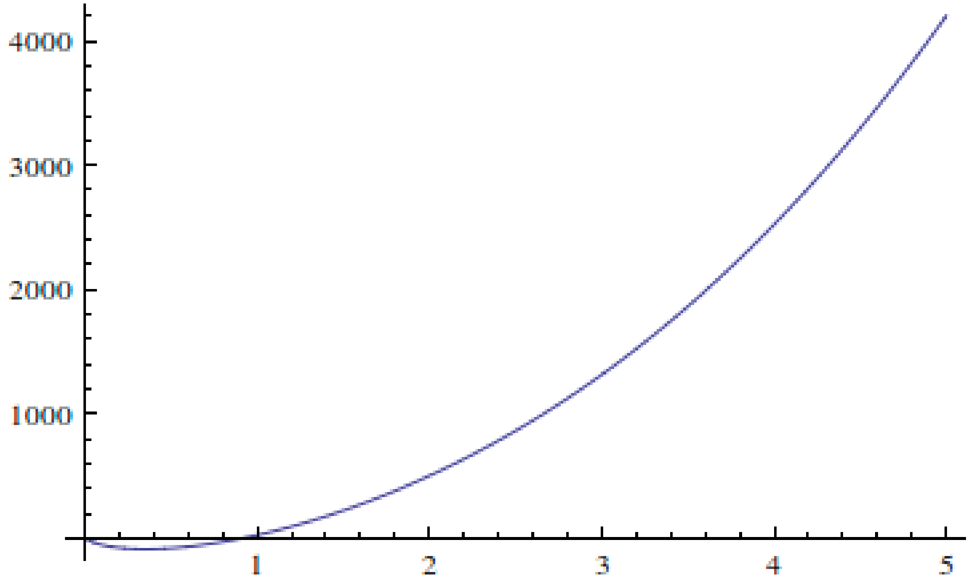
Looking at the plot of $|f(n)|$ may be instructive. For $0 \leq n \leq 1.0$ we have:



That does not seem very helpful. Let's extend our range to $0 \leq n \leq 5$.

8. Data Structures and Algorithms :: Complexity without Proof (continued)

Here is $|f(n)|$ for $0 \leq n \leq 5$:



Now we seem to be getting somewhere. At around $n = 0.9$, $|f(n)|$ becomes positive and exhibits polynomial-like behavior. Recall:

$$|f(n)| = 3n + 5 + 17n^3 + 177\ n\ lg\ n$$

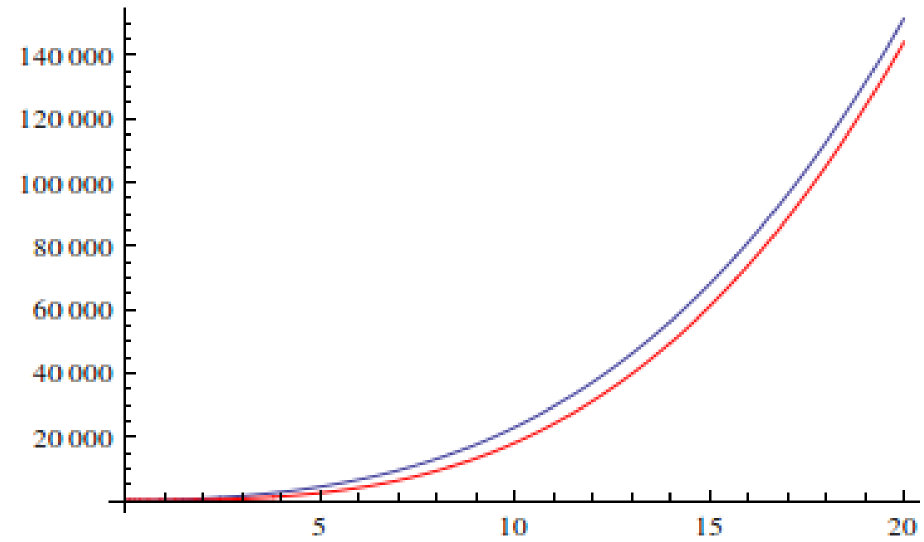
which has four terms:

$3n$
5
$17n^3$
$177n\ lg\ n$

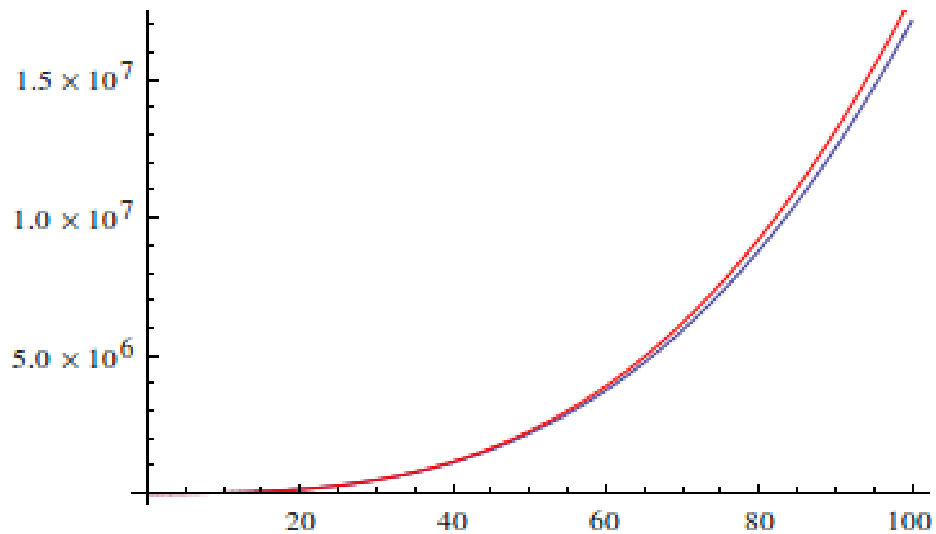
Can you guess which one of these term dominates, i.e., has the largest order of growth?

8. Data Structures and Algorithms :: Complexity without Proof (continued)

That's right: n^3. In fact, below is a plot of $|f(n)|$ and $18n^3$ from $n = 0$ to 20:



Here is the same plot from $n = 0$ to 100:



8. Data Structures and Algorithms :: Complexity without Proof (continued)

It is clear from these two plot that somewhere between $n = 20$ and $n = 80$, $18n^3$ overtakes $f(n)$ and never looks back. Therefore, to rigorously prove that $f(n)$ is $O(n^3)$ we could choose $g(n) = n^3$, $C = 18$, and $n_0 = 80$.

The rigorous proof will be left as an exercise to the viewer, but to conclude, note that we have informally proved that $f(n) = 3n + 5 + 17n^3 - 177\ n\ lg\ n$ is $O(n^3)$ which means the algorithmic time complexity of algorithm A is $O(n^3)$.