Asymptotic Complexity Examples

September 7, 2018

1 Example 1: Prove that $f(n) = 3n^3 + 2n$ is $O(n^3)$.

To prove that $f(n) = 3n^3 + 2n$ is $O(n^3)$, we need to show that $\exists c > 0$ and $n_0 > 0$, such that $0 < f(n) = 3n^3 + 2n \leq cn^3, \forall n \geq n_0$. (By the definition of O).

We know that $0 < 3n^3 + 2n \leq 3n^3 + 2n^3 = 5n^3, \forall n \geq 1$. Therefore, if $c = 5$ and $n_0 = 1$, it is true that $f(n) = 3n^3 + 2n \leq cn^3$, which completes the proof.

2 Example 2: Prove that $f(n) = n + lg(n)$ is $\Theta(n)$.

To prove that $f(n) = n + lg(n)$ is $\Theta(n)$, we need to show that:

1. $f(n) = n + lg(n)$ is $O(n)$ and
2. $f(n) = n + lg(n)$ is $\Omega(n)$

Let's first show that $f(n) = n + lg(n)$ is $O(n)$, using the definition of O. We know that $0 < n + lg(n) \leq n + n = 2n, \forall n > 2$. So if $c = 2$, and $n_0 = 2$, $0 < f(n) = n + lg(n) \leq cn, \forall n \geq n_0$, which shows that $f(n) = O(n)$.

Now let's show that $f(n) = n + lg(n)$ is $\Omega(n)$, using the definition of Ω. We know that $0 < n \leq n + lg(n), \forall n > 2$. So if $c = 1$ and $n_0 = 2$, $0 < cf(n) = c(n + lg(n)) \leq n, \forall n \geq n_0$, which shows that $f(n) = O(n)$.

3 Example 3: Prove that the running time of the following algorithm is $O(n)$

Algorithm 1: For-loop with 3 iterations

for $i = 1; i \leq 3; i = i + 1$ **do**
| Some operations with total run time $f(n) = O(n)$
end

Let $p(n)$ be the function defining the total number of operations inside the for-loop. To prove that the running time of this algorithm is $O(n)$, we need to show that $\exists c > 0$ and $n_0 > 0$, such that $0 < p(n) \leq cn, \forall n \geq n_0$. Since one iteration of the loop takes $f(n)$ operations and the loop iterates 3 times, we know that $0 < p(n) = 3f(n)$. The algorithm states, that $f(n) = O(n)$, which means $\exists c_1 > 0$ and $n_0 > 0$, such that $0 < f(n) \leq c_1 n, \forall n \geq n_0$. Let $c = 3c_1$. Then $p(n) = 3f(n) \leq 3c_1 n = cn, \forall n \geq n_0$, which shows that $p(n)$ is $O(n)$, by definition.