# Assignment 4 - Algorithm Analysis

Vishak Srikanth

September 27, 2021

## 1 Assignment 4 - Solutions to Questions 1-4

1. Arrange the following expressions by growth rate from slowest to fastest.
   $4n^2$, $log_3 n$, $n!$, $3^n$, $20n$, 2, $log_2 n$, $n^{2/3}$
   Use Stirling's approximation in for help in classifying $n!$
   Stirling's approximation states that $n! \approx \sqrt{2\pi n} * (n/e)^n$
   *Solution.*
   $n! \approx \sqrt{2\pi n} * (n/e)^n$, so $n!$ grows at a rate similar to $n^n$. $n^n$ is only multiplied by a factor of 3 rather than a factor of n/e as n increases by 1, so $3^n$ has a slower growth rate than $n!$. $4n^2$ is a quadratic function and grows slower than an exponential function. $20n$ is a linear function and grows slower than a quadratic function. The linear function $20n$ grows faster than $n^{2/3}$ since as n is double $20n$ is doubled, but $n^{2/3}$ does not even double. The logarithmic function $log_2 n$ grows slower than power functions such as $n^{2/3}$. The function $log_2 n$ grows faster than $log_3 n$, since as n is double $log_2 n$ increases by 1, but n must be tripled for $log_3 n$ to be increased by 1. The constant function 2 does not increase with n, so 2 grows the slowest. Therefore for growth rates, $2 < \log_3 n < \log_2 n < n^{2/3} < 20n < 4n^2 < 3^n < n!$.

2. Estimate the number of inputs that could be processed in the following cases:

   (a) Suppose that a particular algorithm has time complexity T(n)=$3 * 2^n$, and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose that we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds?
   *Solution.*
   We could process 6 more inputs in t seconds, since the new machine can process what the old machine could in 64t seconds. The old machine could process n+6 inputs in 64t seconds since T(n+6)=$3 * 2^{n+6}$=64*(3*$2^n$)=64t. Therefore, We could process 6 more inputs(n+6 inputs) in t seconds.

   (b) Suppose that another algorithm has time complexity T(n) = $n^2$, and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose that we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds?
   *Solution.*
   We could process 8 times as many inputs in t seconds, since the new machine can process what the old machine could in 64t seconds. The old machine could process 8n inputs in 64t seconds since T(8n)=$(8n)^2$=64*($n^2$)=64t. Therefore, We could process 8 times as many inputs(8n inputs) in t seconds.

   (c) A third algorithm has time complexity T(n) = 8n. Executing an implementation of the algorithm on a particular machine takes t seconds for n inputs. Given a new machine that is 64 times as fast, how many inputs could we process in t seconds?
   *Solution.*

We could process 64 times as many inputs in t seconds, since the new machine can process what the old machine could in 64t seconds. The old machine could process 64n inputs in 64t seconds since T(64n)=$8 * 64n$=64*($8n$)=64t. Therefore, We could process 64 times as many inputs(64n inputs) in t seconds.

3. A hardware vendor claims that their latest computer will run 100 times faster than that of their competitor. If the competitor's computer can execute a program on input of size $n$ in one hour, what size input can vendor's computer execute in one hour for each algorithm with the following growth rate equations?
$n \quad n^2 \quad n^3 \quad 2n$
*Solution*

   (a) n is the growth rate equation. *Solution.*
   We could process 100 times more inputs in one hour, since the new machine can process what the old machine could in 100 hours. The old machine could process 100n inputs in 100 hours since T(100n)=100$n$=100*(n)=100*1 hour. Therefore, We could process 100 times more inputs(100n inputs) in one hour.

   (b) $n^2$ is the growth rate equation.
   *Solution.*
   We could process 10 times as many inputs in 1 hour, since the new machine can process what the old machine could in 100 hours. The old machine could process 10n inputs in 100 hours since T(10n)=$(10n)^2$=100*($n^2$)=100*1 hour. Therefore, We could process 10 times as many inputs(10n inputs) in t seconds.

   (c) $n^3$ is the growth rate equation.
   *Solution.*
   We could process $100^{\frac{1}{3}}$ times as many inputs in one hour, since the new machine can process what the old machine could in 100 hours. The old machine could process $100^{\frac{1}{3}}$n inputs in 100 hours since $T(100^{\frac{1}{3}}n) = (100^{\frac{1}{3}}n)^3 = 100 * (n^3) = 100 * 1$ hour. Therefore, We could process $100^{\frac{1}{3}} = 4.61$ times as many inputs( $100^{\frac{1}{3}}n$ inputs) in one hour.

   (d) $2^n$ is the growth rate equation.
   *Solution.*
   We could process $\lfloor log_2 100 \rfloor$ more inputs in one hour, since the new machine can process what the old machine could in 100 hours. The old machine could process $n + \lfloor log_2 100 \rfloor$ inputs in 100 hours since $T(n + \lfloor log_2 100 \rfloor) = 2^{\lfloor log_2 100 \rfloor} * 1$ hour. Therefore, We could process $\lfloor log_2 100 \rfloor$ more inputs(n + 6 inputs) in one hour.

4. Using the definition of big-Oh, show that 1 is in $O(1)$ and that 1 is in $O(n)$.
*Solution*

   (a) If $f(n) \in O(g(n)) \implies$ there are positive constants $c$ and $n0 \ni$
   $0 \leq f(n) \leq cg(n) \quad \forall \quad n \geq n0$

Since

$$\begin{aligned}
& 1 \in O(1) \\
\therefore \quad & 1 \leq c(1) \\
\text{Let's assume} \quad & c = 1 \\
\implies & 1 \leq 1(1) \\
\implies & 1 \leq 1
\end{aligned}$$

This is true for any value of n

$\therefore$ We can set $\boxed{n0 = 1}$

$$\therefore 1 = O(1) \quad \text{given } c = 1 \text{ and } n0 = 1$$

(b) If $f(n) \in O(g(n)) \implies$ there are positive constants $c$ and $n0 \ni$
$0 \leq f(n) \leq cg(n) \quad \forall \quad n \geq n0$
Since

$$\begin{aligned}
& 1 \in O(n) \\
\therefore \quad & 1 \leq c(n) \\
\text{Let's assume} \quad & c = 1 \\
\implies & 1 \leq 1(n) \\
\implies & 1 \leq n
\end{aligned}$$

This is true for all $n \geq 1$

$\therefore$ We can set $\boxed{n0 = 1}$

$$\therefore 1 = O(n) \quad \text{given } c = 1 \text{ and } n0 = 1$$

5. For each of the following pairs of functions, either $f(n) \in O(g(n))$ or $f(n) \in \Omega(g(n))$, or $f(n) = \Theta(g(n))$. For each pair, determine which relationship is correct. Justify your answer.
*Solution Approach*
For each case we examine $\lim_{x \to +\infty} \frac{f(x)}{g(x)}$. The asymptotic notations are closely related to the definition of a limit. As a result, $\lim_{x \to +\infty} \frac{f(x)}{g(x)}$ reveals a lot about the asymptotic relationship between $f$ and $g$, provided the limit exists. The following scenarios reveal the relationship between the limit of $\frac{f(x)}{g(x)}$ into facts about the asymptotic relationship between $f$ and $g$:
Case 1:

$$\lim_{x \to +\infty} \frac{f(x)}{g(x)} = c$$

where c is a positive constant $c \neq 0$ and $c \neq +\infty$

$$\implies \quad f(n) \in \Theta(g(n))$$

Case 2:

$$\lim_{x \to +\infty} \frac{f(x)}{g(x)} = 0$$

$$\implies \quad f(n) \in O(g(n))$$

Case 3:

$$\lim_{x \to +\infty} \frac{f(x)}{g(x)} = +\infty$$

$$\implies \quad f(n) \in \Omega(g(n))$$

Answers to Questions 5 and 6 will be submitted in handwritten format as a separate file