Dheeraj Satya Sushant Viswanadham
(z5204820)

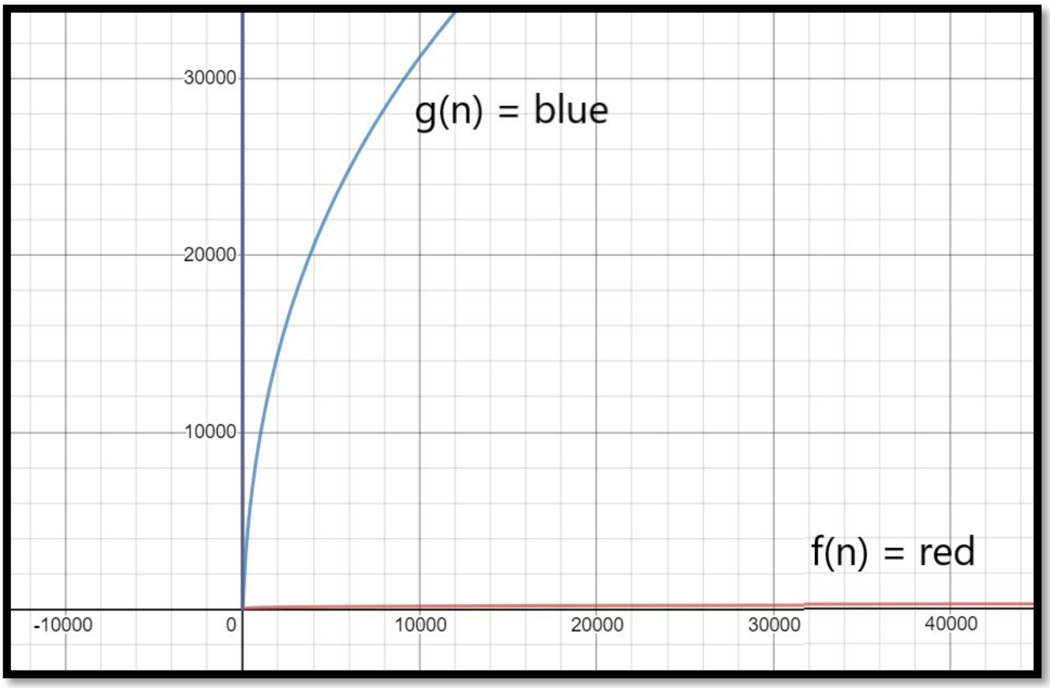
COMP3121 Assignment 1 – Question 5

Visuals of our functions have been included to better explain the points made.

5A) Note – for the purposes of question 5A log will refer to log base 2.

$f(n) = (log_2(n))^2$

$g(n) = log_2\left(n^{log_2 n}\right)^2$



Taking g(n) above, we can apply the following log rule –

log(a)^b = b*log(a)

Where g(n) will now be:

2 * log(n^(logn))

And reapplying the above log rule again to get:

2 * logn * logn

Which can be simplified to:

2 * [logn]^2

Which is essentially:

c * f(n) [as f(n) = [logn]^2] as well] for a positive c = 2 and n is sufficiently large.

Now, if we take the limit of this as n approaches infinity, we can observe that:

$$\lim_{\{n\to\infty\}} \frac{f(n)}{g(n)} = \lim_{\{n\to\infty\}} \frac{f(n)}{2*f(n)} = \frac{1}{2}$$

We have to show that f(n) <= c * g(n) for some constant c and n.

Taking c = 1, it is easy to see that f(n) < g(n) as f(n) = [log(n)]^2 < 2*[log(n)]^2 = g(n) for all n.

∴ **f(n) = O(g(n))** as f(n) will grow slightly slower than g(n) as g(n) is the asymptotic upper bound of f(n).

Next, to prove that f(n) = Ω(g(n)) we need to show that there exists a c > 0 so that c*g(n) <= f(n).

We can take c = ½ which gives us:

$$g(n) = \frac{1}{2} * (2*(\log(n))^2) = (\log(n))^2 \leq (\log(n))^2 = f(n)$$

The above is true and so **f(n) = Ω(g(n))** as well.

∴ We have **f(n) = Θ(g(n))** as the constant is negligible as both functions belong to the same family of complexities and we have shown above that **f(n) = O(g(n))** and **f(n) = Ω(g(n))**.

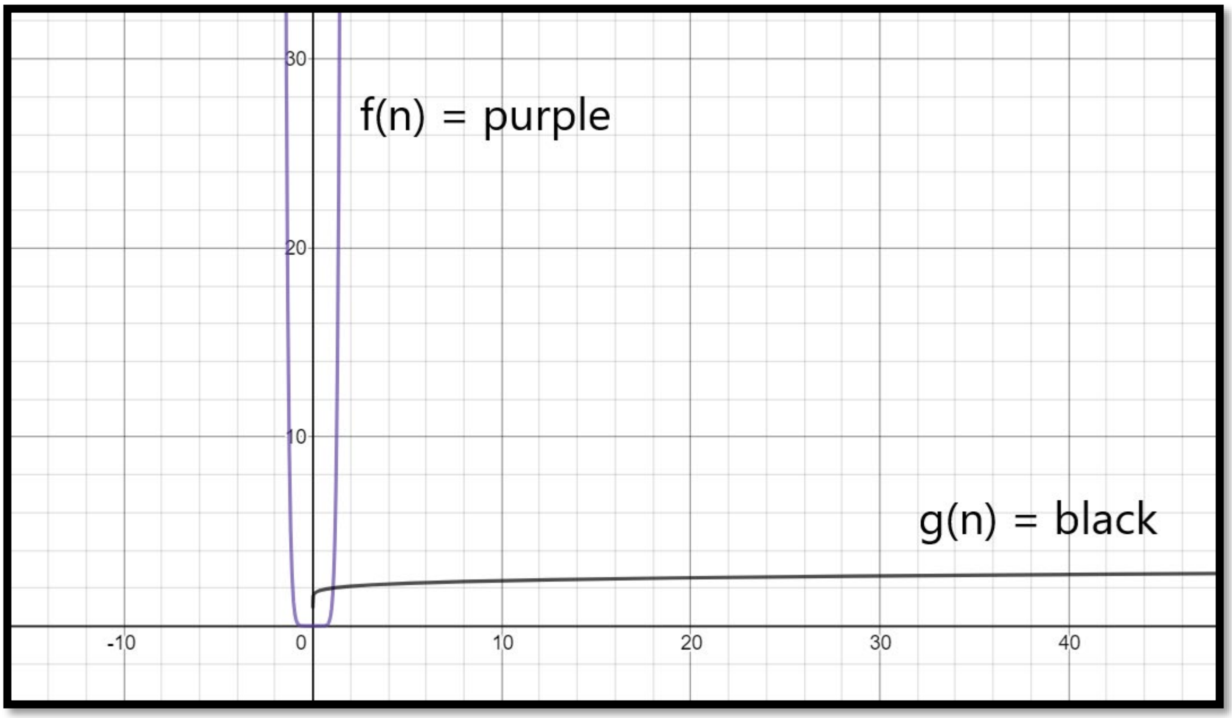
Q5B is on the next page!

Dheeraj Satya Sushant Viswanadham
(z5204820)

5B) Note – for the purposes of question 5B log will refer to log base 2.

$f(n) = n^{10}$

$g(n) = 2^{\sqrt[10]{n}}$ = 2^(n^(1/10))



Please note for the above graph – it does not fully show the real picture as for sufficiently large n [talking n = 10^30 as an example], g(n) will grow faster than f(n). This can be shown here:

n = 10^30: f(n) = $(10^{30})^{10}$ = 1e+300

n = 10^30: g(n) = $2^{(10^{30})^{(1 \div 10)}}$ = 1.071509e+301

Now, we want to show that **f(n) = O(g(n))** which means that we have to show that n^10 < c*2^(n^(1/10)) for some positive c and all sufficiently large n. We know that the log function is monotonically increasing and this will hold just in case:

log(n^10) < log(c) + log2^(n^(1/10))

and applying the same log rule as in 5(a) [i.e. **log(a)^b = b*log(a)**]

10 * log(n) < log(c) + n^(1/10) [as the logarithm of 2 to base 2 = 1]

If we now take c = 1, we can show that [where log(c) = log(1) = 0]:

10* log(n) < n^(1/10)

for all sufficiently large n and we can go further and show that by squaring both sides by 10:

[10*log(n)]^10 < n (for all sufficiently large n).

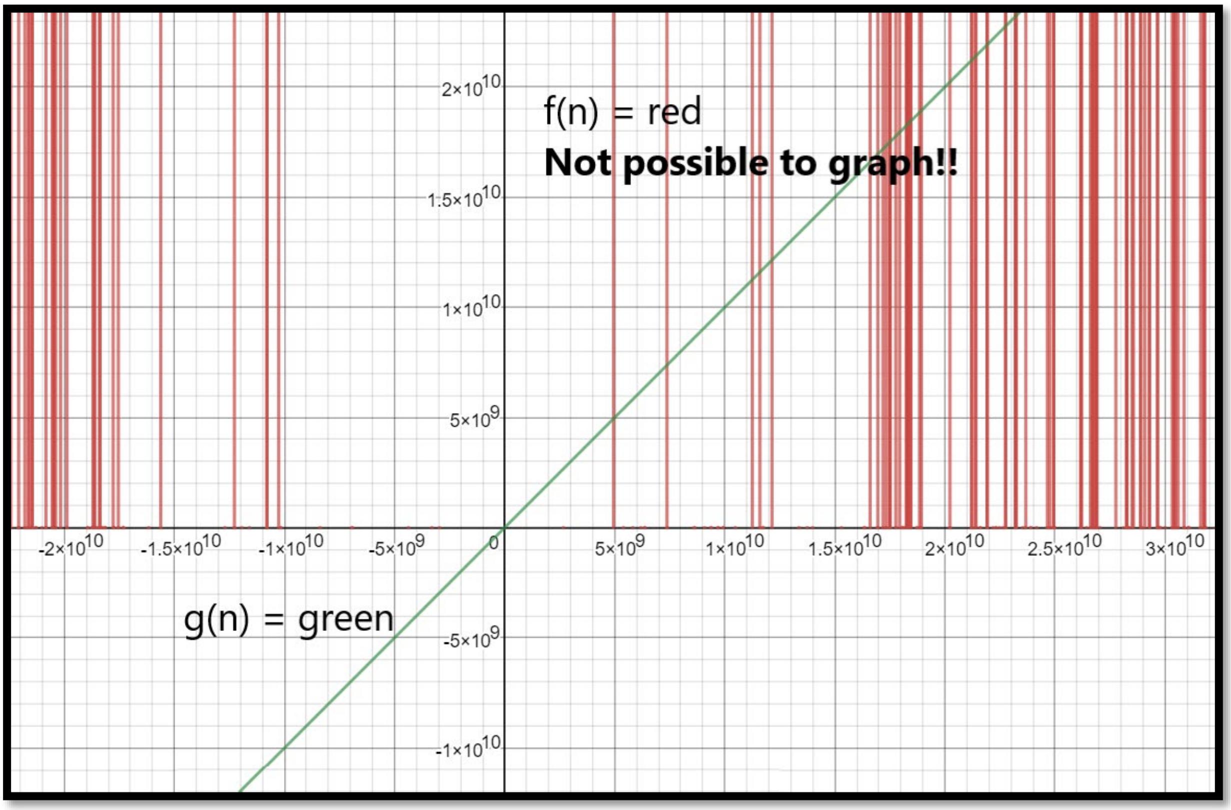
∴ **f(n) = O(g(n))** as f(n) will grow slightly slower than g(n) as g(n) is the asymptotic upper bound of f(n).

Dheeraj Satya Sushant Viswanadham
(z5204820)

5C)

$f(n) = n^{1+(-1)^n}$

$g(n) = n$



Note that f(n) = 1 when n is an odd integer and f(n) = n^2 when n is an even integer.

As seen above, it is not possible to graph this above as it is dependent on the n given.

This is because it is not necessarily true that for every two functions, f(n) and g(n), either f(n) = O(g(n)) or g(n) = O(f(n)).

Putting this another way, let n be a nonnegative, increasing integer -

$$f(n) = \begin{cases} n^2 & \text{if n is even} \\ 1 & \text{if n is odd} \end{cases}$$

Meanwhile, g(n) will remain to be a linear function i.e. g(n) = n is the expression where as n increases, g(n) is also strictly increasing.

Hence, as seen above, there is no **constant** positive multiple of n which, for sufficiently large n, bounds f(n) from below as there will always be the next odd input which is below any constant we can pick.

∴ Neither f(n) = O(g(n)) nor g(n) = O(f(n)).