CSE 551
Assignment 1

Arun Kumar Kumarasamy

September 21, 2020

Question 1

Prove or disprove the following assertions:

(i) If $f(n) = O(g(n))$ then $\log_2 f(n) = O(\log_2 g(n))$
(ii) If $f(n) = O(g(n))$ then $3^{f(n)} = O(3^{g(n)})$
(iii) If $f(n) = O(g(n))$ then $f(n)^3 = O(g(n)^3)$

Solution:

(i) **False**. Disprove by counterexample.

Since $f(n) = O(g(n))$, we know that,

$$0 \leq f(n) \leq C * O(g(n))$$

Suppose $f(n) = 2$ and $g(n) = 1$, Then for all n, $2 \leq C * 1$ for any constant $C \geq 2$. However, $\log f(n)$ is not in $O(\log g(n))$ in the case that $f(n) = 2$ and $g(n) = 1$ since $\log 2 > C * \log(1) = 1 > C * 0$ for any n and any constant C.

(ii) **False**. Disprove by counterexample.

Since $f(n) = O(g(n))$, we know that,

$$0 \leq f(n) \leq C * O(g(n))$$

Suppose $f(n) = 3n$ and $g(n) = n$. Then for all n, $3n \leq C * n$ for any constant $C \geq 3$. However, 3^{3n} is not in $O(3^n)$ in the case if $f(n) = 3n$ and $g(n) = n$ because $3^{3n} >> C * 3n$ implies that, $1 > C * 0$ for any $n > 0$ and any constant C. Thus $3^{f(n)} > O(3^{g(n)})$

(iii) True

Since $f(n) = O(g(n))$, we know that,

$$0 \leq f(n) \leq C * O(g(n))$$

Hence there exists some n_0 for All $n > n_0$, such that $f(n) \leq C * g(n)$ for some C. Hence for all $n > 0$, $C > 0$, $f(n)^3 \leq C^3 * g(n)^3$

Question 2

Algorithm A_1 takes $10^{-4} \times 2^n$ seconds to solve a problem instance of size n and Algorithm A_2 takes $10^{-2} \times n^3$ seconds to do the same on a particular machine.
(i) What is the size of the largest problem instance A2 will be able solve in one year ?
(ii) What is the size of the largest problem instance A2 will be able solve in one year on a machine one hundred times as fast ?
(iii) Which algorithm will produce results faster, in case we are trying to solve problem instances of size less than 20 ?

Solution:

(i) 1466.

A_2 takes $10^{-2} \times n^3$ seconds to solve a problem instance of size n
Taking 1 year = 365 days = 31536000 seconds,
size for one year is given by

$$10^{-2} \times n^3 = 31536000$$

hence $n = \sqrt[3]{31536000}$
n = 1466

(ii) 6806

Again considering 1 year = 365 days = 31536000 seconds,
let N' be the size of problem instance solved by 100x machine.
Time taken by 100x machine to solve N' sized problem is given by,

$$10^{-2} \times N^{'3}$$

which is 100 times the time taken by regular machine. Hence,

$$10^{-2} \times N^{'3} = 100 \times 31536000$$

Solving for N' gives us, $\boldsymbol{N' = 6806}$

(iii) $\boldsymbol{A_1}$ produces result **faster than** $\boldsymbol{A_2}$ when $\boldsymbol{n < 20}$

By substitution different values less than 20,
Ex: for n = 10, A_1 takes 0.1024 seconds and A_2 takes 10 seconds
Ex: for n = 19, A_1 takes 52.4288 seconds and A_2 takes 68.59 seconds

Question 3

Prove or disprove the following with valid arguments:

(i) $3n^2 + 1000 = O(n)$.
(ii) $2n^3 \log(n) = \Theta(n^3)$.
(iii) $3^n n^4 + 8 * 4^n n^3 = O(3^n n^4)$.

Solution:

(i) **False**.Disprove by contradiction.

Let $f(n) = 3n^2 + 1000$ and $g(n) = n$
if $f(n) = O(g(n))$ then $f(n) \leq C * g(n)$ such that $n > n_0$ for a $C > 0$
But since for any c:(such that $n > c$)

$$3n^2 > cn$$

Hence, $\boldsymbol{3n^2 + 1000 \neq O(n)}$

(ii) False. Disprove by contradiction.
Let $f(n) = 2n^3 \log(n)$
Let $g(n) = n^3$
if $2n^3 \log(n) = \Theta(n^3)$ then, there should exist $c_1, c_2 > 0$ and positive $n > n_0$ such that,

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

i.e., $f(n)$ is $O(g(n))$ and $f(n)$ is $\Omega(g(n))$
Assuming, $f(n)$ is $O(g(n))$, then

$$2n^3 \log(n) \leq C * n^3$$

$$2 \log(n) \leq C$$

$$\log(n) \leq C'$$

The above can not be true since c must be a constant but $\log(n)$ is unbounded. This is a contradiction with the assumption that we can find such a constant c. Therefore, $2n^3 \log(n)$ is not $O(n^3)$ and hence, $\boldsymbol{2n^3 \log(n) \neq \Theta(n^3)}$

(iii) False. Disprove by contradiction.
Let $f(n) = 3^n n^4 + 8 * 4^n n^3$ and $g(n) = 3^n n^4$
if $f(n) = O(g(n))$ then $f(n) \leq C * g(n)$ such that $n > n_0$ for a $C > 0$
Assuming, $f(n)$ is $O(g(n))$, then

$$3^n n^4 + 8 * 4^n n^3 \leq C * 3^n n^4$$

$$8 * 4^n n^3 \leq 3^n n^4 (C - 1)$$

$$\frac{8 * 4^n}{3^n n} \leq C - 1$$

$$\frac{8 * 4^n}{3^n n} + 1 \leq C$$

The above can not be true since c must be a constant but $\frac{8*4^n}{3^n n}$ is unbounded. This is a contradiction with the assumption that we can find such a constant c. Therefore, $\boldsymbol{3^n n^4 + 8 * 4^n n^3 \neq O(3^n n^4)}$

Question 4

Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

(i)$f_1(n) = n^{4.2}$.
(ii)$f_2(n) = (2n)^{1.2}$.
(iii)$f_3(n) = n^{4.1} + 87$.
(iv)$f_4(n) = 60^n$.
(v)$f_5(n) = 180^n$.

Solution:

$\boldsymbol{f_2(n) < f_3(n) < f_1(n) < f_4(n) < f_5(n)}$
Since f_4 and f_5 are exponential and will grow the fastest we'll put them at the end of the list. $f_4 < f_5$ because $60 < 180$. Other three functions are polynomial and will grower slower then exponential. We can represent f_1, f_2 and f_3 as:

$$f_1(n) = n^{4.2} = n^4 \times n^{0.2}$$

$$f_2(n) = (2n)^{1.2} = 2n \times (2n)^{0.2}$$

$$f_3(n) = n^{4.1} + 87 = n^4 \times n^{0.1} + 87$$

The polynomials with higher degree grows faster. hence, $f_1, f_3 > f_2$. Since $n \to \infty$, 87 in f_3 is negligible. Hence, $f_1 > f_3$