Home Assignment 1 - Theoretical part

Ahmed Nouralla
Group: 02

February 17, 2020

3.1 Big Oh notation

Prove or disprove the following statements using the definition of big O notation:

1. $$f(n) = \frac{n^2}{3} - 3n,\ g(n) = n^2$$

We know that:

$$n^2 \leq n^2 \Rightarrow \frac{n^2}{3} \leq \frac{n^2}{3} \qquad \forall n \geq 0 \quad (1)$$

We also know that:

$$-n \leq n^2 \Rightarrow -3n \leq 3n^2 \qquad \forall n \geq 0 \quad (2)$$

Adding (1), (2) we get:

$$\frac{n^2}{3} - 3n \leq \frac{10}{3}n^2 \qquad \forall n \geq 0$$

Therefore:

$$\frac{n^2}{3} - 3n = O(n^2)$$

by definition of Big-Oh, with

$$n_0 = 0,\ c = \frac{10}{3}$$

2.

$$f(n) = k_1.n^2 + k_2.n + k_3,\ g(n) = n^2$$

We know that:

$$k_3 \leq n^2 \quad \forall n \geq \sqrt{|k_3|}\ (1)$$

We also know that:

$$k_2.n \leq n^2 \quad \forall n \geq k_2\ (2)$$

We also know that:

$$k_1.n^2 \leq k_1.n^2 \quad \forall n \geq 0\ (3)$$

Adding (1), (2), (3) we get:

$$k_1.n^2 + k_2.n + k_3 \leq (2 + k_1).n^2 \quad \forall n \geq max(\sqrt{|k_3|}, k_2, 0)$$

Therefore:

$$k_1.n^2 + k_2.n + k_3 = O(n^2)$$

by definition of Big-Oh, with

$$n_0 = max(\sqrt{|k_3|}, k_2, 0), c = (2 + k_1)$$

3. Suppose $3^n = O(2^n)$ Then there exists constants $n_0, c > 0$ such that

$$3^n \leq c.2^n \quad \forall n \geq n_0$$

Taking log for both sides we get

$$n.log(3) \leq log(c) + n.log(2) \Rightarrow n \leq \frac{log(c)}{log(3) - log(2)}$$

It means that for any constant c that we choose, we can construct another constant

$$n_0 = \frac{log(c)}{log(3) - log(2)}$$

such that

$$3^n \geq c.2^n \quad \forall n \geq n_0$$

Therefore, such constants c, n_0 don't exist and $3^n \neq O(2^n)$

4.
$$f(n) = 0.001 * n.\log_b(n) - 2000n + 6,\ g(n) = n.\log_b(n)$$

We know that

$$0.001 \leq 1 \Rightarrow 0.001 * n.\log_b(n) \leq n.\log_b(n) \quad \forall n \geq 1\ (1)$$

We also know that

$$-1 \leq \log_b(n) \Rightarrow -n \leq n.\log_b(n) \Rightarrow -2000n \leq 2000n.\log_b(n) \quad \forall n \geq 1\ (2)$$

We also know that

$$1 \leq \log_b(n) \Rightarrow 6 \leq 6.\log_b(n) \leq 6.n.\log_b(n) \quad \forall n \geq b > 1\ (3)$$

Adding (1), (2), (3) we get

$$0.001 * n.\log_b(n) - 2000n + 6 \leq 2007.n.\log_b(n) \quad \forall n \geq b$$

Therefore:
$$0.001 * n.\log_b(n) - 2000n + 6 = O(n.\log_b(n))$$

by definition of Big-Oh, with

$$n_0 = b, c = 2007$$

3.2 Hashing

Consider a hash table of size 7 with hash function $h(k) = k \textbf{ mod } 7$.
Draw the table that results after inserting, in the given order, the following values:
19, 26, 13, 48, 17 for each of the scenarios below:

1. When collisions are handled by **separate chaining**.

Index	0	1	2	3	4	5	6
Value	null	null	null	[17]	null	[19, 26]	[13, 48]

* [e1, e2, …] denotes a linked list with elements e1, e2, …

2. When collisions are handled by **linear probing**.

Index	0	1	2	3	4	5	6
Value	13	48	null	17	null	19	26

3. When collisions are handled by **double hashing** using the secondary hash function $h'(k) = 5 - (k \textbf{ mod } 5)$.

Index	0	1	2	3	4	5	6
Value	null	null	48	17	26	19	13