Quiz 1

David Trinh

September 28, 2024

- **Question 1**

$$r = \sum_{i=0}^{n} \sum_{j=0}^{n-i-1} 1 \tag{1}$$

$$r = \sum_{i=0}^{n} (n-i) \tag{2}$$

$$r = \sum_{i=0}^{n} n - \sum_{i=0}^{n} i \tag{3}$$

$$r = n(n+1) - \frac{n(n+1)}{2} \tag{4}$$

$$r = \frac{n(n+1)}{2} \tag{5}$$

$$r = \frac{n^2+n}{2} \tag{6}$$

$$r = \frac{1}{2}n^2 + \frac{1}{2}n \tag{7}$$

Big-Θ: $\Theta(n^2)$
Let positive constants c_1 and n_o, we have:

$$\tfrac{1}{2}n^2 + \tfrac{1}{2}n \leq c_1 \cdot n^2 \text{ for all } n \geq n_0$$

$$\frac{1}{2} + \frac{1}{2n} \leq c_1$$

Let n be 1, we have:

$$\frac{1}{2} + \frac{1}{2 \cdot 1} \leq c_1$$

$$1 \leq c_1$$

As $n \to \infty$, the term $\frac{1}{2n}$ tend to 0.
Thus, for all $n \geq 1$, $c_1 \geq 1$.
Therefore, there exist $n_0 = 1$ and $c_1 = 6$ and Big-O: $O(n^2)$.

Let positive constants c_2 and n_o, we have:

$$\tfrac{1}{2}n^2 + \tfrac{1}{2}n \geq c_2 \cdot n^2 \text{ for all } n \geq n_0$$

$$\frac{1}{2} + \frac{1}{2n} \geq c_2$$

Let n be 1, we have:

$$\frac{1}{2} + \frac{1}{2 \cdot 1} \geq c_2$$

$$1 \geq c_2$$

As $n \to \infty$, the term $\frac{1}{2n}$ tend to 0, leaving constant $\frac{1}{2}$.
Thus, for all $n \geq 1$, $c_2 \geq \frac{1}{2}$.
Therefore, there exist $n_0 = 1$ and $c_2 = \frac{1}{2}$ and Big-Ω: $\Omega(n^2)$.

Therefore, $\frac{1}{2}n^2 + \frac{1}{2}n$ has a Big-Θ: $\Theta(n^2)$

- **Question 2**

  1. Converting an image of size nxn from color to grayscale.
     Big-Θ: $\Theta(n^2)$
     Every pixel in the image is visited once to calculate its grayscale using a math formula based on its RGB values in constant time. Therefore, as the width/height of the image grows ($n \to \infty$), the number of pixel grows exponentially (n^2), but the time to calculate the grayscale for each pixel remains the same.

  2. Multiplying two matrices of size nxn
     Big-Θ: $\Theta(n^3)$
     Multiply two nxn matrix would result in a nxn matrix, such that each value in the new matrix need to be calculated individually. In addition, going from n to n + 1 will meant that each value in the new matrix has to perform one more multiplication and addition. Therefor, as $n \to \infty$, the matrix multiplication grows at the rate of n^3.

  3. Searching for a number in an unsorted array of size n.
     Big-Θ: $\Theta(n)$
     The function is traversing the entire list and checking every number against a target, which is in linear time.

  4. Searching for a number in an balanced binary search tree with n nodes.
     Big-Θ: $\Theta(logn)$
     The function only has to perform one operation at every level of the tree. Given that as the tree grow linearly in height, n grow exponentially. Therefore, the higher n is, the lower the rate of change in tree height is. And because this search function correlates linearly with tree height, it is in $\Theta(logn)$.