CSE 222 Homework #1

Muhammed Bilal Türk - 210104004047 -

Q1.

a) $f(n) = (n^2 - 3n)^2$ and $g(n) = 5n^3 + n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{(n^2-3n)^2}{5n^3+n} = \lim_{n\to\infty} \frac{n^4 - 6n^3 + 9n^2}{5n^3+n} =$$

$$= \lim_{n\to\infty} \frac{n^4\left(1 - \cancel{\frac{6n^3}{n^4}}^{\to 0} + \cancel{\frac{9n^2}{n^4}}^{\to 0}\right)}{n^3\left(5 + \cancel{\frac{n}{n^3}}^{\to 0}\right)} = \lim_{n\to\infty} \frac{n^4}{5n^3} = \lim_{n\to\infty} \frac{n}{5} = \infty$$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$. Therefore, $f(n) \in \Omega(g(n))$.

b) $f(n) = n^3$ and $g(n) = \log_2 n^4$

$$\boxed{(\log_a f(x))' = \frac{f'(x)}{f(x)} \cdot \ln a}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^3}{\log_2 n^4} = \frac{\infty}{\infty}$$

$\xrightarrow{\text{L'hospital}}$ $$\lim_{n\to\infty} \frac{3n^2}{\frac{4n^3}{n^4} \ln_2} = \lim_{n\to\infty} \frac{3n^2 \cdot n^4}{4n^3 \cdot \ln_2} = \lim_{n\to\infty} \frac{3n^3}{4\ln_2} =$$

$$= \frac{1}{4\ln_2} \lim_{n\to\infty} 3n^3 = \infty$$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$. Therefore, $f(n) \in \Omega(g(n))$.

c) $f(n) = 5n \log_2(4n)$ and $g(n) = n \cdot \log_2(5^n)$

$$\boxed{(a^{f(x)})' = a^{f(x)} \cdot f'(x) \cdot \ln a}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{5n \cdot \log_2(4n)}{n \cdot \log_2(5^n)} = \frac{\infty}{\infty}$$

$\xrightarrow{\text{L'Hospital}}$ $$\lim_{n\to\infty} \frac{5 \cdot \frac{\cancel{4}}{\cancel{4}n \cdot \cancel{\ln_2}}}{1 \cdot \frac{\cancel{5^n} \cdot \ln_5}{\cancel{5^n} \cdot \cancel{\ln_2}}} = \lim_{n\to\infty} \frac{\frac{5}{n}}{\ln_5} = \lim_{n\to\infty} \frac{5}{n \cdot \ln_5} = 0$$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Therefore, $f(n) \in O(g(n))$.

d) $f(n) = n^n$ and $g(n) = 10^n$.

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^n}{10^n} = \lim_{n\to\infty} \left(\frac{n}{10}\right)^n = \infty^\infty = \infty$$

$\lim_{n\to\infty} \frac{f(n)}{f(n)} = \infty$. Therefore, $f(n) \in \Omega(g(n))$.

e) $f(n) = 8n.\sqrt[5]{2n}$ and $g(n) = n.\sqrt[3]{n}$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{8n.\sqrt[5]{2n}}{n.\sqrt[3]{n}} = \lim_{n\to\infty} \frac{2^3.n.2^{\frac{1}{5}}.n^{\frac{1}{5}}}{n.n^{\frac{1}{3}}} = \lim_{n\to\infty} \frac{2^{\frac{16}{5}}.n^{\frac{6}{5}}}{n^{\frac{4}{3}}}$$

$$= 2^{\frac{16}{5}} \lim_{n\to\infty} \frac{n^{\frac{6}{5}}}{n^{\frac{4}{3}}} = 2^{\frac{16}{5}}. \lim_{n\to\infty} n^{\frac{6}{5}-\frac{4}{3}} = 2^{\frac{16}{5}}. \lim_{n\to\infty} n^{-\frac{2}{15}}$$

$$= 2^{\frac{16}{5}}. \lim_{n\to\infty} \frac{1}{\sqrt[15]{n^2}} = 2^{\frac{16}{5}}.0 = 0$$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Therefore, $f(n) \in o(g(n))$.

Q2.

a) str_array[i] == ""; has a O(1) complexity. But it is a inside of loop. The complexity of loop is O(n). Therefore, complexity of methodA() is O(n).

b) We know complexity of methodA is $O(n)$. In methodB, methodA is inside of the loop whose complexity is $O(n)$. Therefore, complexity of first for loop is $O(n) \cdot O(n) = O(n^2)$. Second for loop has a print statement whose complexity is $O(1)$. Therefore, complexity of second for loop is $O(n)$. Consequently, the complexity of methodB is $O(n^2)$.

c) We know complexity of methodB is $O(n^2)$. In methodC, there is a inner for loop which has methodB inside of itself. Complexity of outer for loop is n, complexity of inner for loop is n too, complexity of methodB is n^2. Therefore, complexity of methodC is $n \cdot n \cdot n^2 = O(n^4)$.

d) In methodD, there is a loop which increments i by 1. However, i is decrementing by 1 in index operator. Therefore, i will never change except 0 and 1. So this causes infinite loop. Consequently, we can't analyze methodD in terms of time complexity.

e) In methodE, iterator may not reach to the end of the array but we assume it will. Because when analyze the worst-case time complexity, we must consider worst scenerio. Therefore, "if" statement will never implemented. So complexity of methodE is $O(n)$ due to the for loop.

Q3. a) initialize min with first element of array A.
initialize max with last element of array A.

calculate difference by subtracting min from max

print difference

If array is sorted in ascending order, it means first element of array is minimum element and last element of array is maximum element. To be able to calculate max difference, subtract min from max.

Time complexity of this algorithm is O(1) because accessing element by index is a constant time.

b) initialize min with first element of array A.
initialize max with first element of array A.

for i=1 to n which is size of array A.
    if A[i] is less than min
        assign A[i] to min

    if A[i] is greater than max
        assign A[i] to max

calculate difference by subtracting min from max.

print difference

if array is not sorted, to be able to determine min and max all element must be checked. Therefore, initilize min and max with first element of array A. Then iterate each element of array by checking if element that is currently iterated is less than min or greater than max. In less case, assign element that is currently iterated to min; in greater case, assign element that is currently iterated to max. Then calculate difference by subtracting min from max.

Time complexity of this algorithm is O(n) because there is a loop which iterates all elements of array A whose size is n. Inside the loop, if statements and assignments have constant time complexity. Consequently, time complexity of this algorithm is O(n).