CSE 222 - Homework 2

Mert Emir Şeker
2021040040085

Q1)

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

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

Applying l'hopital rule

$$\lim_{n\to\infty} \frac{4n^3 - 18n^2 + 18n}{15n^2+1} \to \text{l'hopital} \to \lim_{n\to\infty} \frac{12n^2 - 36n}{30n}$$

Applying l'hopital rule

$$\lim_{n\to\infty} \frac{24n - 36}{30} = \infty$$

So,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \longrightarrow f(n) = \Omega(g(n))$$

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

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

Applying l'hopital rule

$$\lim_{n\to\infty} \frac{3n^2}{\frac{4}{n \ln 2}} = \lim_{n\to\infty} \frac{3n^3 \cdot \ln 2}{4} = \infty$$

So,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \longrightarrow f(n) = \Omega(g(n))$$

CSE 222 - Homework 2

Mert Emir Şeker
2021040205

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

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{5n\log_2(4n)}{n\log_2(5^n)} = \lim_{n\to\infty} \frac{5n(2+\log_2 n)}{n^2 \log_2 5}$$

$$\lim_{n\to\infty} \frac{5(2+\log_2 n)}{n\log_2 5} \xrightarrow{\text{l'hopital}} \lim_{n\to\infty} \frac{\frac{5}{n\ln 2}}{\log_2 5}$$

$$\lim_{n\to\infty} \frac{5}{n \ln 2 \log_2 5} = 0$$

So,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \longrightarrow f(n) = O(g(n))$$

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

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

$$\lim_{n\to\infty} e^{n\ln(\frac{n}{10})} = \infty$$

So,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \longrightarrow f(n) = \Omega(g(n))$$

CSE 222 Homework 2

Mert Emin Şeker
200104004085

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

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

$$\lim_{n\to\infty} \frac{2^3 n^1 2^{\frac{1}{5}} n^{\frac{1}{5}}}{(n)^{\frac{4}{3}}} = \lim_{n\to\infty} \frac{(2)^{\frac{16}{5}} (n)^{\frac{6}{5}}}{(n)^{\frac{4}{3}}}$$

$$\lim_{n\to\infty} (2)^{\frac{16}{5}} (n)^{\frac{-2}{15}} = \lim_{n\to\infty} \frac{(2)^{\frac{16}{5}}}{(n)^{\frac{2}{15}}} = 0$$

So,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \longrightarrow f(n) = O(g(n))$$

Q2)

a) static void methodA(String str_array[]){

```
    for(int i=0; i<str_array.length; i++)
        str_array[i] = " ";
}
```

→ MethodA utilize a single for loop over the array elements.

→ Each iteration performs a constant time assignment operation. ($O(1)$)

→ The loop runs a total n times for each element of array. ($O(n)$)

So, since the methodA method performs a constant time operation n times, its time complexity is;

$$O(n \cdot 1) = O(n)$$

CSE 222 - Homework 2

Mert Emir Seker
20210604085

b) Static void methodB(String str_array[]){
for(int i=0; i<str_array.length; i++)
methodA(str_array);
for(int j=0; j<str_array.length; j++)
System.out.println(str_array[j]);
}

→ MethodB contains two seperate for loops First one invokes methodA and second one calls println for each element in array.

→ First for loop calls methodA n times and since methodA has a time complexity O(n) this loop's complexity is:

$$O(n \cdot n) = O(n^2)$$

→ Second for loop runs n times to print each element of array and each print operation takes constant time(O(1)) so this loop's complexity is:

$$O(n \cdot 1) = O(n)$$

So the total time complexity of methodB is:

$$O(n^2 + n) = O(n^2)$$

c) static void methodC(String str_array[]){
for(int i=0; i<str_array.length; i++)
for(int j=0; j<str_array.length; j++)
methodB(str_array);
}

→ MethodC contains a nested loop in which array iterated over n times both inner and outer loops.

→ Both loops run n times methodB is invoked $n \cdot n = n^2$ times. So each invokation methodB is $O(n^2)$ time complexity.

→ Therefore when methodB is called n^2 times the methodC's total time complexity becomes:

$$O(n^2 \cdot n^2) = O(n^4)$$

CSE 222 - Homework 2

Mert Emir Şeker
2021060040

d)
```
static void methodD(String str_array[]){
    for(int i=0; i< str_array.length; i++){
        System.out.println(str_array[i]);
        str_array[i--] = "";
    }
}
```

→ In methodD there is a single loop that iterates over the elements of str_array.

→ Inside the loop it prints current element which is an O(1) time complexity.

→ After printing it sets the current element to an empty string and decrements the loop variable i--. This decrementation causes an infinite loop. Because we are incrementing and decrementing i inside the for loop and i never reaches str_array.length.

→ Therefore time complexity of methodD cannot be defined.

e)
```
static void methodE(String str_array[]){
    for(int i=0; i<str_array.length; i++)
        if(str_array[i] == "")
            break;
}
```

→ In methodE there is a single loop that iterates through str_array. If an element equals the empty string loops breaks.

→ In the worst case scenario, where no element is an empty string and loop is capable of going through all elements the time complexity is O(n).

→ So, time complexity of methodE is O(n).

CSE 222 - Homework 2

Mert Emir Şeker
2021040054085

Q3)

a) Assuming the array is sorted in ascending order.

```
function Funct1(Array):
    n = length(Array)
    if n < 2
        return "Array must contain at least two elements."
    maxDifference = Array[n-1] - Array[0]
    return maxDifference
```

Time complexity of funct1 is O(1).

b) Assuming the array is not sorted.

```
function Funct2(Array):
    n = length(Array)
    if n < 2
        return "Array must contain at least two elements"
    min = Array[0]
    max = Array[1]
    for i from 1 to n-1:
        if Array[i] < min
            min = Array[i]
        if Array[i] > max
            max = Array[i]
    return max - min
```

Time complexity of Funct2 is O(n).