**CSC 140 Advanced Algorithm Design and Analysis**

**CSC 206 Algorithms and Paradigms**

**Spring 2023**

**Assignment 1: Asymptotic Complexity**

**Solutions**

**Question 1 [30 Points]**

For each pair of functions below, first determine which **function grows asymptotically faster**, and then express the relation between the two functions using all the asymptotic notations (**θ, O, Ω, o** and **ω)** that apply. Only give the relations in which **f(n)** appears on the left-hand side.

Prove the relation between the **first two pairs** of functions using the limit definition.

(1) f(n) = n logn g(n) = n+logn

*f(n) grows faster*

*f(n) =* ω*(g(n))*

*f(n) =* Ω*(g(n))*

Proof: 

(2) f(n) = 2n g(n) = 22n

*g(n) grows faster*

*f(n) = o (g(n))*

*f(n) = O (g(n))*

Proof: 

(3) f(n) = 3n g(n) = 3n+2

*Same order*

*f(n) = θ (g(n))*

*f(n) = O (g(n))*

*f(n) = Ω (g(n))*

(4) f(n) = 6n3+7n2+2 g(n) = n3

*Same order*

*f(n) = θ (g(n))*

*f(n) = O (g(n))*

*f(n) = Ω (g(n))*

(5) f(n) = n50 g(n) = n!

*g(n) grows faster*

*f(n) = o (g(n))*

*f(n) = O (g(n))*

**Question 2 [10 points]**

If we know that the running time **T(n)** of some algorithm satisfies the relations **T(n)** **=** **O(n4 logn)** and **T(n) = ω(n2 log2n),** which of the following functions can **T(n)** possibly be? Circle **all** that apply.

The right answers are underlined:

n2 logn **n3** **n2.2** n2 log2n **n4 logn**

**Question 3 [10 points]**

Given an algorithm with asymptotic complexity **θ**(n4), if the running time = 3s when n=100, what is the expected running time when n=200?

*Time = number\_of\_operations / machine\_speed*

*T1 = n14 / speed ….. (1)*

*T2 = n24 / speed ….. (2)*

*Dividing (2) by (1), we get:*

*T2/T1 = (n2/n1)4*

*T2 = T1 (n2/n1)4*

*Plugging in the numerical values, we get:*

*T2 = 3 (200/100)4 = 3 x 24 = 3 x 16 = 48 s*

**Question 4 [10 points]**

Compute the **largest input size** that each of the following algorithms can solve in **one second**, assuming that they are run on a machine that executes **400 million** operations per second. Compute the best possible numerical value using the given information.

Alg X with an asymptotic complexity of **θ(2n)**

Alg Y with an asymptotic complexity of **θ()**

Alg X with an asymptotic complexity of **θ(2n)**

*Time = number\_of\_operations / machine\_speed*

*1 = 2n/(4x108)*

*2n = 4x108*

*n = log2 (4x108)= log2 4 +8 log210 = 2+ 8 x 3.32 = 28.56*

*The answer is 28*

Alg Y with an asymptotic complexity of **θ()**

*Time = number\_of\_operations / machine\_speed*

*1 = n0.5/(4x108)*

*n0.5 = 4x108*

*n = 16x1016*

**Question 5 [40 points]**

Find the asymptotic complexity for each of the following algorithms. Be as accurate as possible. Assume that the input size is ***n*** and that ***op*** is a constant-time operation (or possibly sequence of operations). **Show your analysis for full credit.**

(1)

for (i=1; i<=n; i\*=3) **θ(logn)**

for (j=0; j<n/5; j+=2) **θ(n)**

for (k=n; k>=1; k--) **θ(n)**

op

**T(n) =** **θ(logn) \* θ(n) \* θ(n) = θ(n2logn)**

(2)

for (i=1; i<n; i+=3) **θ(n)**

{

for (j=4; j<10; j++) **θ(1)**

op

for (k=4; k<n/2; k++) **θ(n)**

op

}

**T(n) =** **θ(n) (θ(1) + θ(n)) = θ(n2)**

(3)

for (i=1; i<n; i+=3) **θ(n)**

{

for (j=1; j<n; j\*=2) **θ(logn)**

op

x = n\*n;

while(x > 1) **θ(n2)**

{op; x--;}

}

**T(n) =** **θ(n) ( θ(logn) + θ(n2) ) = θ(n3)**

(4)

for (i=10; i <= 3\*n; i++) { First, develop an **exact formula** for the running time.

for (j=i; j >= 5; j--) Then, give the asymptotic complexity.

op; Remember that we do the analysis for *very large* **n**.

}

i number of inner loop iterations

--- -----------------------------------------

10 10 - 4 = 6

11 11 - 4 = 7

.

.

.

3n 3n – 4