Dharsan Ravindran (20219218)

Professor Burton Ma

CISC 121

April 9, 2020

Assignment 4 Non-Coding Written Question Answers

## Question 1:

1. $O(n^6)$
2. $O(n^4)$
3. $O(n^2)$
4. $O(2^n)$
5. $O\left(\frac{n}{\sqrt{n}}\right)$

## Question 2:

**If n is an integer power of 2 ($2^n$) and n>1**

| | # of elementary operations | | # of times run |
|---|---|---|---|
| def my_function(n): | | | |
| if n < 1: | **2** | * | **1** |
| return | **1** | * | **0** |
| | | | |
| while n > 1: | **2** | * | **(n+1)** |
| if n % 2 != 0: | **3** | * | **n** |
| n = 3 * n + 1 | **3** | * | **0** |
| | | | |
| else: | **1** | * | **n** |
| n = n / 2 | **2** | * | **n** |
| | | | |
| return n | **1** | * | **1** |

**f(n) = 2+2(n+1)+3n+n+2n+1**
**= 2n+2+n+3n+2n+1+2**
**= 8n+5**

∴ The timing of the function **f(n)** when n is an integer power of 2 and n>1 is **8n+5,** and the big-O complexity is **O(n)**.

## Question 3:

The timing of the function in question 2 can not be derived when the integer n is not equal to an integer power of 2 and n>1, due to the nature of the if statement in the while loop. If you end up with an odd number of any size n the modulus 2 of that number will not equal zero. Causing the function to execute the line n=3*n+1. This will keep executing until the number is divisible by 2. If the integer n is even, while the modulus will equal 0, you will eventually reach an odd number since the number n is not a power of 2. The if statement will execute increasing the value of the integer n. The if and else blocks will keep executing until a power of 2 is reached. As the only way you can satisfy the while loop and exit is if the number n reaches 1, and the only possible way to do this is continuously dividing by 2 as this is the only operation which reduces the integer n. The only way to reach 1 while continuously dividing by 2 is if the number n is a power of 2. It is impossible to tell how many times the if and else statements will execute until the number n reaches a power of two without the exact number provided. An example is the integer 5 and the integer 9. With the integer 5 the if block executes once until a power of 2 is reached and the else block executes 4 times. With the integer 9 the if block executes 5 times, until a power of 2 is reached and the else block executes 10 times. With such variation in the numbers, it is impossible to tell when the integer n will reach a power of 2 eventually reducing to one making it unable to tell the timing of the function when n is not a power of 2 and n>1 .