**Question 1:**

1. O(1)

2. O(n­­­­­­­2)

3. O(log2n)

4. O(2n)

5. O(log2n)

**Question 2:**

**# NUMBER OF ELEMENTARY OPERATIONS**

**def** my\_function(n):

**if** n < 1: # 2 (1 if, 1 comparison)

**return # 1**

**while** n > 1: # 2 (1 while, 1 comparison)

**if** n % 2 != 0: # 3 (1 if, 2 comparisons)

n = 3 \* n + 1 # 3 (1 assignment, 1 addition, 1 multiplication)

**else**: #

n = n / 2 # 2 (1 assignment, 1 division)

**return** n # 1

**# NUMBER OF TIMES LINE IS RUN**

**def** my\_function(n):

**if** n < 1: # 1

**return # 1**

**while** n > 1: # n - 1

**if** n % 2 != 0: # n - 1

n = 3 \* n + 1 # n - 1

**else**: #

n = n / 2 # n - 1

**return** n # 1

f(n) = 2 + 1 + 2(n - 1) 2 + 3(n - 1) 2 + 3(n - 1) 2 + 2(n- 1) 2 + 1

**Question 3:**

The timing function could not be derived unless n is equal to an integer power of 2 and n > 1, as if n < 1, then the function returns nothing, and the worst-path timing could not be calculated. It would also not work if n were not an integer power of 2, as the function must take all possibilities into account, therefore needing n to be a power integer of 2 to satisfy all the possible conditions.