CS 58000-01 Quiz 02 Print your Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

September 6, 2022

1. Analyze the time efficiency for the function multiply(x, y).

function multiply(x, y)

Input: Two integers x and y, where y ≥ 0

Output: Their product

if y = 0 then return 0;

z := multiply (x, └ y/2 ┘);

if y is even then return 2z

else return x + 2z;

1. What is the input size of any integer y?

* \_\_ n =└ log2 ┘ + 1 bits; e.g., let y < . Then n is 8 bits long. \_\_\_

1. What is the total number of recursive calls “multiply (x, └ y/2 ┘” executed?

\_\_\_At most n = └ log2 ┘ + 1 number of recursive calls \_\_\_\_\_\_\_\_\_\_\_\_\_

1. Upon each return from the recursive call “multiply (x, └ y/2 ┘)”, assign the product to z, and then execute the if-then-else statement. What is the time efficiency for executing this if-then-else statement?

\_\_\_\_\_\_\_\_\_O(n) justifying by the following problems d (O(n)), e(O(1)), and f (O(n)). \_\_\_\_\_\_\_\_\_\_\_

1. What is time efficiency for determining whether y is even?

\_\_\_\_\_\_constant time O(1) for every testing the rightmost bit whether is a zero. The other way is to determine whether (y – (└ y/2 ┘)\*2) is zero, where └ y/2 ┘)\*2 can be done by shift-right y by one bit and then shift-left by one bit. Then add the negative result to y. If it is zero, then y is even. This require O(n) + O(1) + O(1) = O(n)\_\_\_\_\_\_\_

1. What is time efficiency for computing 2z?

\_\_\_\_\_\_ constant time O(1) for each left shift-left z by one bit and then append a zero at the rightmost bit, for any given z.\_\_\_\_\_\_

1. What is time efficiency for computing x + 2z?

\_\_\_\_\_\_\_In addition to O(1) for computing 2z, each addition requires O(n), assuming x and y are n-bits long. The time efficiency is O(n)\_\_

1. What is the time efficiency of this algorithm, function multiply(x, y)?

\_\_\_\_\_\_\_n \* O(n) = n\* (c0 + c n) = O(n2).\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Express the function multiple(x, y) in terms of an equation system.

2(x \*└ y/2 ┘), if y is even

x \* y =

\_\_\_\_\_\_ x + 2(x \* └ y/2 ┘), if y is odd\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Is the function multiple(x, y) correct?

It is transparently correct; It also handles the base case (y = 0).­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Does the function multiple(x, y) halt?

\_ The function multiple(x, y) will eventually halt. The reason is: For each recursive call multiply (x, └ y/2 ┘), the continuation of “shift-right y by one bit” for a given n =└ log2 ┘ + 1 bits will eventually yield zero for the value of y. The recursive call will be ended if y = 0 then return 0;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_