CS 310 'Algorithms' Fall 2019 **Quiz-1A** [Total marks=16] [Time: 20 mins]

Name: ______________________ Roll No: _________________________

Q1. Do worst-case analysis of the following my_func() procedure and determine the time each statement takes and the number of times each statement is executed. [8 marks]

	`my_func()`	**Cost x times**
1	`int n = 128;`	c x 1
2	`int i, j, stepso = 0, stepsi = 0;`	c x 1
3	`for (i=1; i<=n; i*=2) {`	c x (log(n) +1 +1)
4	`  stepso++;`	c x (log(n) +1)
5	`  for (j=0; j<n; j++){`	c x (log(n) +1) (n+1)
6	`     stepsi++; } }`	c x (log(n) +1) (n)

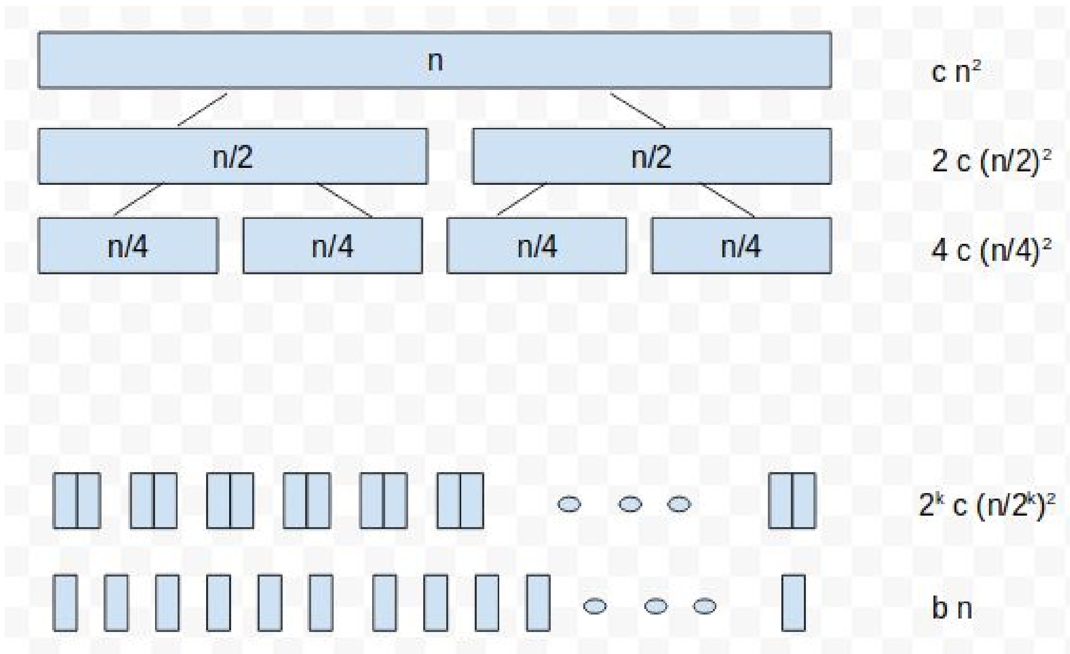
What is the time complexity of the above procedure. For full credit, show your working.
T(n) = c + c + c x (log(n) +2) + c x (log(n) +1) + c x (log(n) +1) (n+1) + c x (log(n) +1) (n)
Taking the dominant term, T(n) is O(n log(n))

Q2. Draw the recurrence tree of the func() procedure and determine the time taken each layer. [8 marks]

```
func(n)

  if (n <= 1)  return n+1;
  else {
    compute(n); // compute() is O(n²)
    return func(n/2)*func(n/2); }
```

Recurrence tree



What is the time complexity of the above procedure. For full credit, show your working.

$T(n) = c\, n^2 + 2\, c\, (n/2)^2 + 4\, c\, (n/4)^2 + \dots + 2^k\, c\, (n/2^k)^2 + b\, n$
$k = \log(n) - 1$
$T(n) = c\, n^2 (1 + ½ + ¼ + ⅛ + \dots + ½^{(\log(n)-1)}) + b\, n$
$T(n) = c\, n^2 (1 - ½^{\log(n)}) / (1 - ½) + b\, n$

Simplifying, $T(n)$ is $O(n^2)$

Common Formulae:

$$a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} = \sum_{k=0}^{n-1} ar^k = a\left(\frac{1-r^n}{1-r}\right)$$

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$
$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$
$$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}.$$

$$\log_a b = \frac{\log_c b}{\log_c a}$$

$$\log_a(xy) = \log_a(x) + \log_a(y)$$

$$\log_a n^b = b \cdot \log_a n$$