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

Name: ______________________ Roll No: ________________________

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

	`compute()`	**Cost x times**
1	`int n = 256;`	c x 1
2	`int i, j, in = 0, out = 0;`	c x 1
3	`for (i=0; i<n; i++) {`	c x (n+1)
4	`out++;`	c x (n)
5	`for (j=1; j<=n; j*=2){`	c x (n) (log(n) +1 +1)
6	`in++; } }`	c x (n) (log(n) +1)

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

T(n) = c + c + c (n+1) + c (n) + c x (n) (log(n) +2) + c x (n) (log(n) +1)

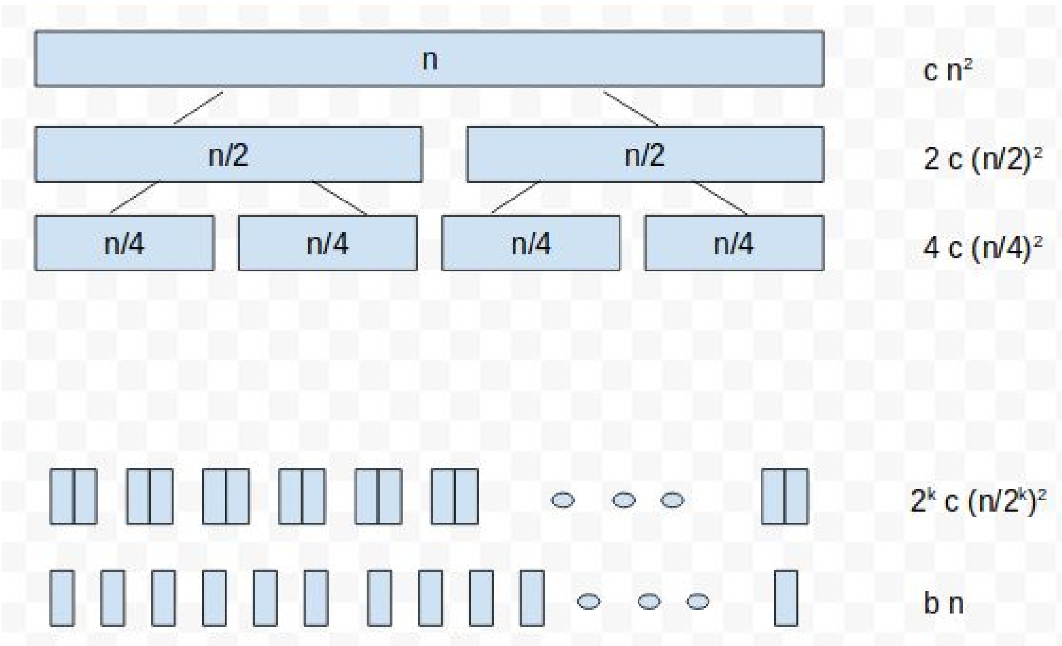
Taking the dominant term, T(n) is O(n log(n))

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

```
split(A, start, end) // A is an n element array

   if (n <= 1)  return 2*(n+1);
   else {
     mid = (start+end)/2;
     func(n); // func() is O(n²)
     return split(A,start,mid)*split(A,mid+1,end); }
```

Recurrence tree



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

$T(n) = c\,n^2 + 2\,c\,(n/2)^2 + 4\,c\,(n/4)^2 + \ldots + 2^k\,c\,(n/2^k)^2 + b\,n$
$k = \log(n)-1$
$T(n) = c\,n^2\,(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots + \frac{1}{2}^{(\log(n)-1)}) + b\,n$
$T(n) = c\,n^2\,(1 - \frac{1}{2}^{\log(n)}) / (1 - \frac{1}{2}) + b\,n$

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

Common Formulae:

$a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} = \sum_{k=0}^{n-1} ar^k = a\left(\frac{1-r^n}{1-r}\right)$
$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$ $\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}.$
$\log_a b = \frac{\log_c b}{\log_c a}$
$\log_a(xy) = \log_a(x) + \log_a(y)$
$\log_a n^b = b \cdot \log_a n$