Julian Garcia
CSE344: Hw3
I certify that every answer in this assignment is the result of my own work; that Ihave neither copied off the Internet nor from any one else's work; and I have not shared my answers or attempts at answers with anyone else.

1. Analysis of the time complexity of a recursive algorithm has resulted in the following recurrence equations.

   $$T(1) = 1$$
   $$T(n) = n^2 + 4T(\tfrac{n}{2}) \quad \text{for } n > 1$$

   (a) Use the top-down method to find a closed-form solution for $T(n)$. (Recap: the top-down method is one in which you use the recurrence equations to rewrite $T(n)$ in terms of $T(x)$ where $x < n$ and repeat the process until you reach the base case.)

   (b) Draw a recursion tree to get a closed-form solution for $T(n)$. Use the notation in the class (show the input size within each node, annotate it with the the non-recursive time, indicate the number of nodes at each level, the non-recursive time for each level near the right margin, and the grand total vertically below). Your tree must include at least three topmost levels including the root and two bottom-most levels including the leaf.

(a) $T(1) = 1$
$T(n) = n^2 + 4T(n/2)$ for $n>1$

$T(n) = n^2 + 4(n/2)^2 + 16T(n/4)$ After 1 substitution

$T(n) = n^2 + 4(n/2)^2 + 16(n/4)^2 + 64T(n/8)$ After 2 substitutions

$T(n) = n^2 + 4(n/2)^2 + \dots + 4^{n-2}(n/2^{n-2})^2 + 4^{n-1}T(n/2^{n-1})$ After n-2 substitutions

$T(n) = n^2 + 4(n/2)^2 + \dots + 4^{n-2}(n/2^{n-2})^2 + 4^{n-1}(n/2^{n-1})^2 + 4^n(n/2^n)^2$ After n substitutions

$T(n) = n^2 + 4(n/2)^2 + \dots + 4^{n-2}(n/2^{n-2})^2 + 4^{n-1}(n/2^{n-1})^2 + 4^n n^2/4^n$ After n substitutions

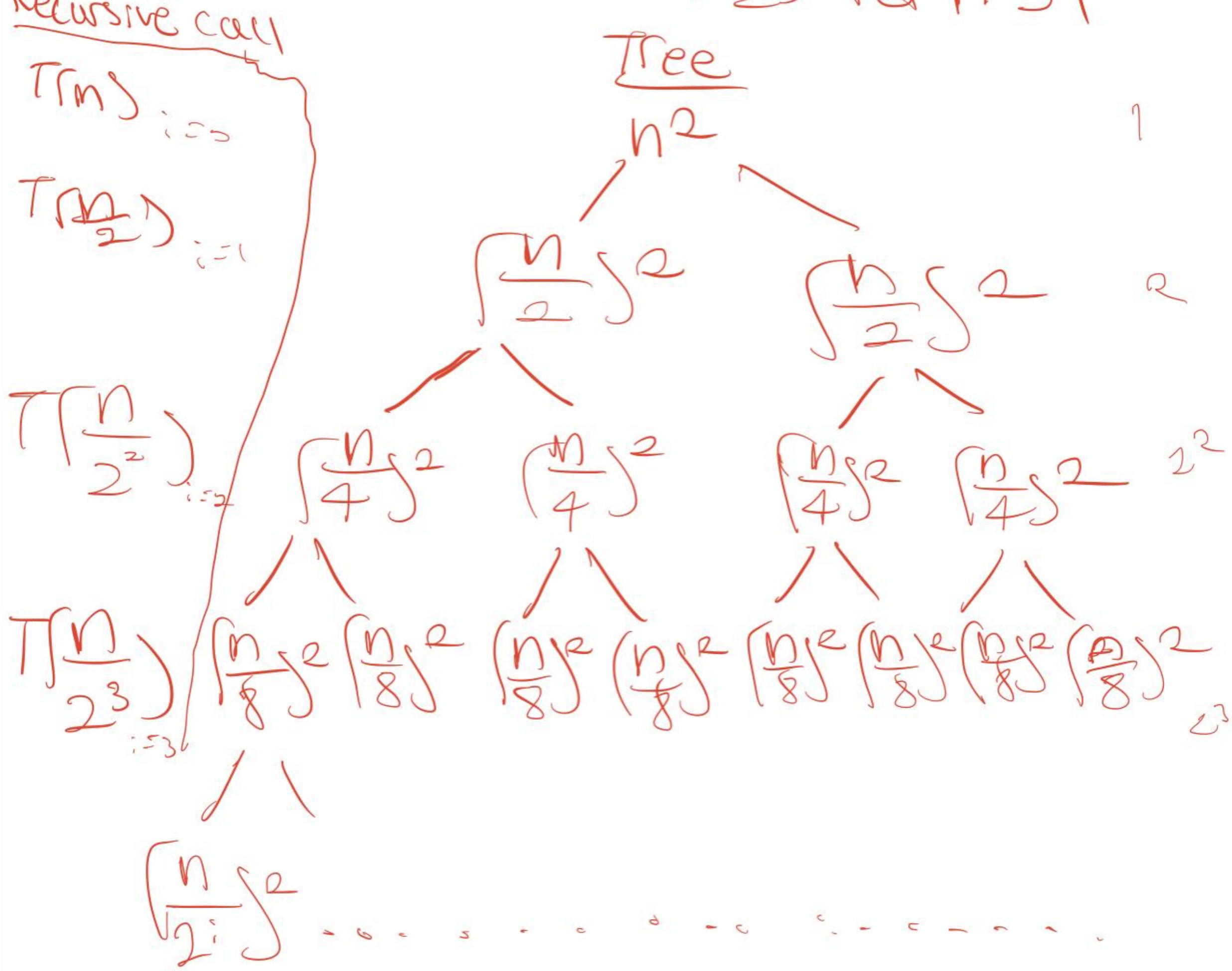
$T(n) = n^2 + n^2 + \dots + n^2$ After n substitutions

$T(n) = O(n^2\log(n))$

(b)

$T(1) = 1$

$T(n) = n^2 + 4T(\frac{n}{2})$ for $n > 1$



Row sum		Level
n^2	—	0
$\frac{n^2}{2}$	—	1
$\frac{n^2}{4}$	—	2
$\frac{n^2}{8}$	—	3
$\frac{n^2}{2^i}$	—	i

$1 = \frac{n}{2^i}$

$\log_2 n = i$

$$\sum_{i=0}^{\log_2 n} \frac{n^2}{2^i} = n^2 \sum_{i=0}^{\log_2 n} \frac{1}{2^i}$$

$$= n^2 \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots + \frac{1}{2^{\log_2 n}}\right)$$

$$\sim \Theta(n^2 \log(n))$$

2. Suppose $T(n)$ satisfies the following recurrence equations.

$$\begin{aligned} T(1) &= 1 \\ T(n) &= n + 4T(\tfrac{n}{4}) \quad \text{for } n > 1 \end{aligned}$$

Use the substitution method (based on the Principle of Induction) to verify that $T(n) = \Omega(n \lg n)$.

Guess: T(n) = Ω(nlgn) ----> i.e T(n) ≥ c(n)(logn) for all n.

Prove By Induction that:

T(n) ≥ c(n)(logn) for all n ≥ n_0 ≥ 1 and some c ≥ 0

Assume T(m) ≥ (c)(m)(logm) for all 2 ≤ m < n.

Proof:

T(n) = 4T(n/4) + n

*≥ 4(c*n/4*log(n/4)) + n*

*= cn*log(n/4) + n*

= cn(log n) − cn + n

≥ cn(log n) if −cn + n ≤ 0 ⇒ c ≤ 1

True if c ≤ 1

So, by choosing c = 1, we can establish our guess that T(n) = Ω(nlgn)