Q1

FALSE First, we observe that the equation $T(n) = 3T(\frac{n}{2}) + n$ is of a form which is suitable to use with the master method. We see that $a = 3$, $b = 2$, and $f(n) = n$. Following, $n^{log_b a}$ is equal to $n^{log_2 3}$, which is approximately equal to $n^{1.58}$. From this, we can set ϵ to a constant value of $n^{log_2 3} - 1$. Since $\epsilon > 0$ and $f(n) = n = O(n^{log_b a - \epsilon}) = O(n)$, then the solution is $T(n) = \Theta(n^{log_2 3}) \approx \Theta(n^{1.58})$.

Q2

FALSE This question is tricky, as it is not immediately apparent that the equation $T(n) = T(\sqrt{n}) + 1$ is of a form that is suitable for use with the master method. However, we can employ a change of variables (as seen in the textbook on page 86) that will allow us to process the equation with the master method. Perform the following:

1. $T(n) = T(\sqrt{n}) + 1$

2. $m = lg(n)$

3. $T(2^m) = T(2^{m/2}) + 1$

4. $S(m) = S(m/2) + 1$

5. $a = 1$, $b = 2$, and $g(m) = 1$

6. $n^{log_b a} = n^{log_2 1} = 1 = g(m)$ (CASE 2)

7. $S(m) = \Theta(lg(m))$

8. $T(n) = \Theta(lg(lg(n)))$

Q3

THE MASTER METHOD CANNOT BE USED FOR THIS EQUATION The equation $T(n) = 2T(\frac{n}{3} + 1) + n$ is not of a form suitable for use with the master method. The master method is used for "divide and conquer" style recursion equations, and while this equation does have elements of the "divide and conquer" style, it also incorporates a linear term (the "+ 1") that makes it unsuitable for use with the master method.

Q4

(a) The recursion tree for this equation $T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$ can be viewed on page 91 of the textbook. Considering that the upper bound of the equation can be found by multiplying the max height of the tree (the longest simple path from root to leaf) by the cost at each level, it stands to reason that the lower bound can be estimated by multiplying the shortest simple path from root to leaf by the cost at each level. In this case, the shorted possible path is $log_3 n$, but the

estimate for complexity will still be the same; that is, $\Omega(nlgn)$. We can verify this using the same substitution method used for verifying the upper bound as seen on page 92 of the textbook. In this case, the compartor will be reversed.

$$T(n) \geq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$$
$$\geq d(\frac{n}{3})lg(\frac{n}{3}) + d(\frac{2n}{3})lg(\frac{2n}{3}) + cn$$
$$= dnlgn - dn(lg3 - \frac{2}{3}) + cn$$
$$\geq dnlgn$$

The previous relation stands as long as $d \leq \frac{c}{lg3-2/3}$. Thus, the lower bound is confirmed to be $\Omega(nlgn)$.

(b) The recursion tree for this problem is illustrated here.
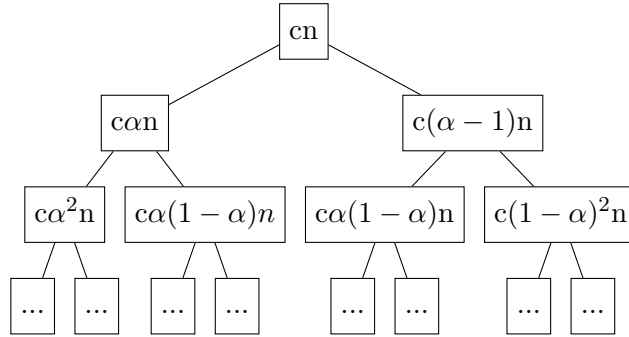
It can be seen that the sum at each level is cn. Assuming $0.5 < \alpha < 1.0$, the longest path (upper bound) is following the leftmost branch of the recursion tree. Conversely, the shortest path is found by following the rightmost path of the recursion tree. The maximum height of the tree is $log_{\frac{1}{\alpha}} n$. The minimum height is $log_{\frac{1}{1-\alpha}} n$. The total cost is given by the height multiplied by the cost per level, which gives an estimate for both the upper and lower bound as approximately nlgn. This can be used as a guess for verification by substitution (as demonstrated on page 92 of the textbook).

$$T(n) \leq T(\alpha n) + T((1-\alpha)n) + cn$$
$$\leq d\alpha n \lg(\alpha n) + d(1-\alpha)n \lg((1-\alpha)n) + cn$$
$$= d\alpha n \lg(\alpha) + d\alpha n \lg(n) + d(1-\alpha)n \lg(1-\alpha) + d(1-\alpha)n \lg(n) + cn$$
$$= dn \lg(n) + d\alpha n \lg(\alpha) + d(1-\alpha)n \lg(1-\alpha) + cn$$
$$\leq dn \lg(n)$$

Holds when $d\alpha n \lg(\alpha) + d(1-\alpha)n \lg(1-\alpha) + cn \leq 0$. Similarly, for the lower bound the relation holds when $d\alpha n \lg(\alpha) + d(1-\alpha)n \lg(1-\alpha) + cn \geq 0$. Thus, $\Theta(nlgn)$.