1. (30 points) Assume that $T(1) \in \Theta(1)$. Solve the following recurrences using the recursion tree method.

a. $T(n) = 4T(n/2) + n^2$

![Recursion Tree for $T(n) = 4T(n/2) + n^2$]

The sum is dominated by the last term, which is $n^2$ with height $h = \log_2 n$.

Total $= n^2 \log_2 n = \Theta(n^2 \log n)$

b. $T(n) = T(n - 2) + n$

$T(n) = n + (n - 2) + (n - 4) + \ldots + (n - 2i) + \ldots + 0 = (n + 0)/2 \ast (n/2 + 1) \in \Theta(n^2)$.

c. $T(n) = 4T(n/2) + n$

This is similar to (a) but the sum of each level follows an increasing geometric series:

The $i$-th level has $4^i$ nodes and each node has value $\frac{n}{2^i}$, so the sum of each level is $2^i \cdot n$. The sum of the series is dominated by the last term, which is $2^h \cdot n$, with $h = \log_2 n$ being the height of three. Therefore $T(n) = \Theta(2^{\log_2 n}) = \Theta(n^2)$.

d. $T(n) = 2T(n - 2) + 1$
2. (40 points) Assume that \( T(1) \in \Theta(1) \). Solve the following recurrence functions using the master method. If the master method cannot be applied, state the reason, and give an upper bound (big-Oh) as tight as you can. Justify your answer.

a. \( T(n) = 8T(n/2) + n^2 \);
\[ f(n) = n^2. \ a = 8, \ b = 2, \ \text{so} \ \ n^\log_b a = n^\log_2 8 = n^3. \ n^\log_b a/f(n) = n \in \Omega(n^\epsilon) \ \text{for} \ \epsilon = 0.5. \ \text{This is case 1.} \ \text{Therefore} \ T(n) \in \Theta(n^3). \]

b. \( T(n) = T(3n/5) + n; \)
\[ f(n) = n. \ a = 1, \ b = 5/3, \ \text{so} \ \ n^\log_b a = 1. \ f(n)/n^\log_b a = n \in \Omega(n^\epsilon) \ \text{for} \ \epsilon = 1. \ \text{This is possibly case 3.} \ \text{This recurrence also satisfies} \ \text{the regularity condition} \ af(n/b) \leq cf(n): \ 3n/5 \leq cn \ \text{for} \ c = 3/5. \ \text{Therefore} \ T(n) \in \Theta(f(n)) = \Theta(n). \]

c. \( T(n) = 9T(n/3) + n^2; \)
\[ f(n) = n^2. \quad a = 9, \quad b = 3, \quad \text{so } n^{\log_b a} = n^2 = \Theta(f(n)). \] This is case 2. Therefore \( T(n) = \Theta(n^2 \log n). \)

d. \( T(n) = 16T(n/4) + n \log n; \)
\[ f(n) = n \log n. \quad a = 16, \quad b = 4, \quad \text{so } n^{\log_b a} = n^2, \quad n^{\log_b a}/f(n) = n^2/n \log n = n/\log n = n^{0.5}n^{0.5}/\log n = n^{0.5}\Omega(1) = \Omega(n^{0.5}). \] This is case 1. Therefore \( T(n) \in \Theta(n^2). \)

e. \( T(n) = 2T(n/4) + \log^2 n; \)
\[ f(n) = \log^2 n. \quad a = 2, \quad b = 4, \quad \text{so } n^{\log_b a} = n^{0.5}, \quad n^{\log_b a}/f(n) = n^{0.5}/\log^2 n = n^{0.25}n^{0.25}/\log^2 n = n^{0.25}\Omega(1) = \Omega(n^{0.25}). \] This is case 1. Therefore \( T(n) \in \Theta(n^{0.5}). \)

f. \( T(n) = 3T(n/3) + \log n; \)
\[ f(n) = \log n. \quad a = 3, \quad b = 3, \quad \text{so } n^{\log_b a} = n. \quad n^{\log_b a}/f(n) = n/\log n = n^{0.5}n^{0.5}/\log n = n^{0.5}\Omega(1) = \Omega(n^{0.5}). \] This is case 1. Therefore \( T(n) \in \Theta(n). \)

g. \( T(n) = 4T(n/4) + n \log n; \)
\[ f(n) = n \log n. \quad a = 4, \quad b = 4, \quad \text{so } n^{\log_b a} = n. \quad f(n)/n^{\log_b a} = n \log n/n = \log n \notin \Omega(n^\epsilon) \text{ for any } \epsilon > 0. \] Therefore the standard master theorem cannot be applied. However we can use the extended case 2 of the master theorem, as \( f(n) = \Theta(n^{\log_b a} \log n). \) The solution is \( T(n) = \Theta(n^{\log_b a} \log^2 n) = \Theta(n \log^2 n). \)

h. \( T(n) = 2T(n/4) + \sqrt{n}; \)
\[ f(n) = \sqrt{n}. \quad a = 2, \quad b = 4, \quad \text{so } n^{\log_b a} = \sqrt{n} = \Theta(f(n)). \] This is case 2. Therefore \( T(n) = \Theta(\sqrt{n} \log n). \)

i. \( T(n) = 3T(n/3) + (n + \log n); \)
\[ f(n) = n + \log n. \quad a = 3, \quad b = 3, \quad \text{so } n^{\log_b a} = n. \quad f(n) = \Theta(n^{\log_b a}). \] This is case 2. Therefore \( T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n \log n). \)

j. \( T(n) = 2T(n/2) + n/\log n; \)
\[ f(n) = n/\log n. \quad a = 4, \quad b = 4, \quad \text{so } n^{\log_b a} = n. \quad n^{\log_b a}/f(n) = n/(n/\log n) = \log n \notin \Omega(n^\epsilon) \text{ for any } \epsilon > 0. \] Therefore the standard master theorem cannot be applied. Note that the extended case 2 of the master theorem cannot be applied here, as \( f(n) = \Theta(n^{\log_b a} \log^k n), \) where \( k = -1. \) The extended case 2 of the master theorem requires that \( k \geq 0. \)

However it is fair to say that \( T(n) = 4T(n/4) + O(n), \) and therefore \( T(n) = O(n \log n). \)

It is also fair to say that \( T(n) = 4T(n/4) + \Omega(1), \) and therefore \( T(n) = \Omega(n). \)

(Just for your information, the real solution is \( T(n) = n \log \log n. \))

3. (10 points)

a. \( T(n) = 3T(n - 2) + n; \)
\[ \text{Let } n = \log_2 m, \text{ we have } T(\log_2 m) = 3T(\log_2 m/4) + \log_2 m. \] Denote \( T(n) = T(\log_2 m) \) as \( S(m), \) we then have \( S(m) = 3S(m/4) + \log_2 m. \) Using the master method case 1 we can obtain \( S(m) = \Theta(m^{\log_2 3}) = \Theta((2^n)^{\log_2 3}) = \Theta(3^{n \log_2 3}) = \Theta(3^{n/2}) = \Theta(\sqrt{3}^n). \)

Therefore, \( T(n) = \Theta(\sqrt{3}^n). \)

b. \( T(n) = T(n - 2) + n^3; \)
\[ \text{Let } n = \log_2 m, \text{ we have } T(\log_2 m) = T(\log_2 m/4) + \log_2^3 m. \] Denote \( T(n) = T(\log_2 m) \) as \( S(m), \) we then have \( S(m) = S(m/4) + \log_2^3 m. \) Using the master method extended case 2 we can obtain \( S(m) = \Theta(\log_2^4 m) = \Theta(\log_2^4 n^3) = \Theta(n^4). \) (Note: \( \log_2^3 m \) means \( (\log_2 m)^3. \))

Therefore, \( T(n) = \Theta(n^4). \)
4. (20 points) Analysis of recursive algorithms.

a. Skipped.

b. To prove that Alg2 is correct, we can use induction.

Base case: Alg2 is correct when \( n = 0 \), as \( 2^0 = 1 \) and the Alg2 returns 1.

Inductive hypothesis: assume that Alg2 works for \( n = k - 1 \), i.e., Alg2(k-1) correctly computes \( 2^{k-1} \).

Step: If the inductive hypothesis is correct, Alg2(k) will return Alg2(k-1) + Alg2(k-1), which is equal to \( 2 \times 2^{k-1} = 2^k \).

Therefore Alg2 is correct.

c. \( A(n) = A(n-1) + 1 \).
\[ B(n) = 2B(n-1) + 1 \]
\[ C(n) = C(n/2) + 1 \]

d. You can solve \( A(n) \) and \( B(n) \) with the recursion tree method and the solution is \( A(n) = \Theta(n) \), and \( B(n) = \Theta(2^n) \).

Using the master method, it is easy to show that \( C(n) = \Theta(\log n) \).

5. (10 points) Assume that \( T(1) \in \Theta(1) \) and \( T(n) = T(3n/4) + T(n/2) + n^2 \). Prove \( T(n) \in \Theta(n^2) \) using the substitution method.

First we prove that \( T(n) \in O(n^2) \). According to the definition of \( O \), we have to show that \( T(n) \leq cn^2 \) for some \( c > 0 \) and all \( n \geq n_0 \). Assume this inequality is true for \( T(3n/4) \) and for \( T(n/2) \), which means \( T(3n/4) \leq c(3n/4)^2 \), and \( T(n/2) \leq c(n/2)^2 \). Given this assumption, we can substitute \( T(3n/4) \) and \( T(n/2) \) in the recurrence relation by the corresponding right hand side of the inequalities. We then have

\[
T(n) = T(3n/4) + T(n/2) + n^2 \\
\leq c(3n/4)^2 + c(n/2)^2 + n^2 \\
\leq 13cn^2/16 + n^2 \\
\leq cn^2 + (n^2 - 3cn^2/16) \\
\leq cn^2, \text{ if } n^2 - 3cn^2/16 \leq 0.
\]

Therefore, if we choose \( c \geq 16/3 \), we have \( T(n) \leq cn^2 \) for all \( n \geq 0 \). Hence, by definition, \( T(n) \in O(n^2) \).

Similarly, we can prove that \( T(n) \in \Omega(n^2) \). According to the definition of \( \Omega \), we need to show that \( T(n) \geq cn^2 \) for some \( c > 0 \) and all \( n \geq n_0 \). Assume this is true for \( T(3n/4) \) and \( T(n/2) \), i.e, \( T(3n/4) \geq c(3n/4)^2 \), and \( T(n/2) \geq c(n/2)^2 \). Substituting \( T(3n/4) \) and \( T(n/2) \) in the recurrence relation by the corresponding right hand side of the inequalities, we have

\[
T(n) = T(3n/4) + T(n/2) + n^2 \\
\geq c(3n/4)^2 + c(n/2)^2 + n^2 \\
\geq 13cn^2/16 + n^2 \\
\geq cn^2 + (n^2 - 3cn^2/16) \\
\geq cn^2, \text{ if } n^2 - 3cn^2/16 \geq 0.
\]

Therefore, if we choose \( c \leq 16/3 \), we have \( T(n) \geq cn^2 \) for all \( n \geq 0 \). Hence, by definition, \( T(n) \in \Omega(n^2) \).

Since \( T(n) \) is in both \( O(n^2) \) and \( \Omega(n^2) \), we conclude that \( T(n) \in \Theta(n^2) \).