1. (20 points) Analysis of recursive algorithms.

(a) Tracing Alg1(2): Alg1(2) = 2 * Alg1(1) = 2 * (2*Alg1(0)) = 2 * (2 * 1) = 4.

(b) Tracing Alg1(3): Alg1(3) = 2 * Alg1(2) = 2 * 4 = 8

(c) Tracing Alg2(2): Alg2(2) = Alg2(1) + Alg2(1) = (Alg2(0) + Alg2(0)) + (Alg2(0) + Alg2(0))
   = (1 + 1) + (1 + 1) = 4.

(d) Tracing Alg2(3): Alg2(3) = Alg2(2) + Alg2(2) = 4 + 4 = 8

(e) Tracing Alg3(1):
   line 3: m = 0;
   line 4: p = Alg3(0) = 1;
   line 5: p = p * p = 1 * 1 = 1;
   line 7: return 2*p = 2;

(f) Tracing Alg3(2):
   line 3: m = 1;
   line 4: p = Alg3(1) = 2;
   line 5: p = p * p = 2 * 2 = 4;
   line 9: return p = 4;

(g) Tracing Alg3(3):
   line 3: m = 1;
   line 4: p = Alg3(1) = 2;
   line 5: p = p * p = 2 * 2 = 4;
   line 7: return 2*p = 8;

(b) To prove that Alg1 is correct, we can use induction.

Base case: Alg1 is correct when \( n = 0 \), as \( 2^0 = 1 \) and the Alg1 returns 1.

Inductive hypothesis: assume that Alg1 works for \( n = k - 1 \), i.e., Alg1(k-1) correctly computes \( 2^{k-1} \).

Step: If the inductive hypothesis is correct, Alg1(k) will return \( 2 * 2^{k-1} = 2^k \).

Therefore Alg1 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 \( C(n) = \Theta(\log n) \).

2. (30 points) Assume that \( T(1) \in \Theta(1) \). Solve the following recurrences using the recursion tree method.
a. \( T(n) = 4T(n/2) + n^2 \)

b. \( T(n) = 2T(n/2) + n^2 \)

This is similar to (a) but each internal node has only two branches rather than four branches; the sum of each level follows a decreasing geometric series:

The \( i \)-th level has \( 2^i \) nodes and each node has value \( (\frac{n}{2^i})^2 = \frac{n^2}{4^i} \), so the sum of each level is \( 2^i \times \frac{n^2}{4^i} = \frac{n^2}{2^i} \). The sum of the series is \( \sum_{i=0}^{h} \frac{n^2}{2^i} \) and \( h = \log_2(n) \), which is dominated by the first term, \( n^2 \).

Therefore \( T(n) = \Theta(n^2) \).

c. \( T(n) = T(n-2) + n \)

\[ T(n) = n + (n - 2) + (n - 4) + ... + (n - 2i) + ... + 0 = (n + 0)/2 \times (n/2 + 1) \in \Theta(n^2). \]

d. \( T(n) = 4T(n/2) + n \)

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 \( 2^i n \). The sum of the series is dominated by the last term, which is \( 2^h n \), with \( h = \log_2 n \) being the height of three.

Therefore \( T(n) = \Theta(2^{\log_2 n} n) = \Theta(n^2) \).

e. \( T(n) = 2T(n-2) + 1 \)
1. \( h = \frac{n}{2} \)

Total = \( 2^{h+1} - 1 \)

= \( \Theta(2^h) \)

= \( \Theta(2^{n/2}) \)

= \( \Theta(\sqrt{2}^n) \)

f. \( T(n) = T(n/2) + T(n/3) + n \)