1. 
   a. \( \sum_{i=3}^{n} (2i + 1) = (n - 3 + 1)((2n + 1) + (2 \times 3 + 1))/2 = (n - 2)(2n + 8)/2 = n^2 + 2n - 8 = \Theta(n^2). \)
   
   b. \( \sum_{i=0}^{n} (3^i + 2^{i+1}) = \sum_{i=0}^{n} 3^i + 2 \sum_{i=0}^{n} 2^i = (3^{n+1} - 1)/2 + 2(2^{n+1} - 1) = \Theta(3^n + 2^n) = \Theta(3^n). \)
   
   c. \( \sum_{i=1}^{n} (\log(n) + i) = \sum_{i=1}^{n} \log(n) + \sum_{i=1}^{n} i = n \log(n) + n(n + 1)/2 = \Theta(n^2). \)
   
   d. \( \sum_{i=0}^{\lfloor \log(n) \rfloor} 2^i = 2^{\lfloor \log(n) \rfloor + 1} - 1. \)

   Note that \( \log(n) \leq \lfloor \log(n) \rfloor + 1 \leq \log(n) + 1. \)

   Therefore, \( 2^{\lfloor \log(n) \rfloor + 1} - 1 = 2n - 1, \) and \( 2^{\lfloor \log(n) \rfloor + 1} - 1 \geq 2^{\log(n)} - 1 = n - 1. \)

   Hence \( \sum_{i=0}^{\lfloor \log(n) \rfloor} 2^i = \Theta(n). \)

2. 
   • \( f(n) \in O(g(n)) \) implies that \( g(n) \in O(f(n) + g(n)). \)

   Proof:

   As both \( f(n) \) and \( g(n) \) are asymptotically positive, \( g(n) \leq f(n) + g(n) \) for sufficiently large \( n. \)

   Therefore \( g(n) \in O(f(n) + g(n)). \)

   • \( f(n) \in O(g(n)) \) implies that \( g(n) \in \Omega(f(n) + g(n)). \)

   Proof:

   By definition of \( O, \) there exists positive \( c \) and \( n_0 \) such that \( f(n) \leq cg(n), \) or equivalently, \( g(n) \geq \frac{1}{c}f(n), \) for all \( n \geq n_0. \) We can show that:

   \[
   g \geq \frac{1}{c}f \\
   g + \frac{1}{c}g \geq \frac{1}{c}f + \frac{1}{c}g \\
   (1 + \frac{1}{c})g \geq \frac{1}{c}(f + g) \\
   (c + 1)g \geq (f + g) \\
   g \geq \frac{1}{c+1}(f + g)
   \]

   Therefore, if we choose \( d = \frac{1}{c+1}, \) we have shown that \( g(n) \geq d(g(n) + f(n)) \) for all \( n > n_0. \) Hence \( g(n) \in \Omega(f(n) + g(n)). \)

   • \( f(n) \in O(g(n)) \) implies that \( 2^{f(n)} \in O(2^{g(n)}). \)

   This is incorrect. Try \( f(n) = 2n \) and \( g(n) = n. \)
3.

a. 3 1 4 5 2
   1 3 4 5 2
   1 2 4 5 3
   1 2 3 5 4
   1 2 3 4 5

b. Line 5 will be executed the most number of times. Line 5 will be executed the same number of times in the best case as well as in the worst case:

\[ T(n) = \sum_{i=0}^{n-2} (n - i) = n + (n - 1) + \cdots + 2 = (n - 1)(n + 2)/2 = \Theta(n^2) \]

c. After moving line 12 and 13 into the inner for loop, the execution of the swap (line 12 and line 13) will depend on the input, and similarly as line 7, they will be executed \( (n^2 - n)/2 \) times in the worst case, but 0 time in the best case.

d. After the move, the running time of the algorithm stays the same \( \Theta(n^2) \), as line 5 is still the most executed line.

4.

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