Order Analysis

BRUTE FORCE COMPLEXITY:

The time complexity of the brute force method is O(n2).

Reasoning: The implementation of the brute force method uses two for loops – the first for-loop traverses through the array, as the second for-loop checks for a complement value. The space complexity is constant because no values are being stored during this process.

HASH MAP COMPLEXITY:

The time complexity of the hash map complexity is O(n).

Reasoning: The initial for-loop will put all the values of the array in a map. Then another loop (not nested) will once again traverse the list searching for the complement value to check if it matches the target value. Though this is faster in time complexity, it requires more space – O(n) space complexity, for all values in the array need to be mapped.

BINARY SEARCH TREE COMPLEXITY:

The time complexity of the BST is O(n).

Reasoning: The implementation of the binary search tree requires a traversal throughout all the elements in the tree, which is O(n) time. Since all values need to be stored, the space complexity is O(n) time.