homework 10

a. list analysis algorithm: algorithm uses nested loops to check all possible pairs of numbers in the list. for each number, it checks every number after to see if their sum equals the target. brute-force approach doesnt use other data structures

b. dictionary algorithm: algorithm uses a dictionary to store numbers seen already as we iterate through the list. for each number it checks if the complement (target - current number) exists in the dictionary. if found, return True. this only uses a single pass through the list

c. separate list algorithm: uses a list where index represents numbers and the value represents their presence. like the dictionary approach but uses a list instead. efficient for non-negative ints and reasonable max values

d. big-oh for list analysis: O(n^2) - for each n element, we check up to n-1 other elements, results in n*(n-1) comparisons

e. big-oh for dictionary: O(n) - make a single pass through the list, and dictionary operations (insert and lookup) are O(1)* (on average)

f. big-oh for separate list: O(n) - make a single pass through list, and list indexing is O(1). however, space complexity is O(k) where k is the maximum number in the list

g. timing observation: for list_size = 10,000, the list analysis will be slow (sec), while dictionary and separate list will be fast (ms). for list_size = 100,000, the list analysis will be really slow (min), while the other two will stay fast. this shows the quadratic vs linear time complexity difference.

h. dictionary preference: the dictionary approach is generally preferred because

    a. it works for integers (not just non-negative)

    b. its more space-efficient when numbers are sparse or large (separate list needs space proportional to the max number)

    c. it doesnt require knowing the max value before