Q: Explain, in English (no code), how you would search a given, sorted, vector for duplicate values. Imagine if you were asked to determine how many values in the vector were duplicated. What would be your process for determining this and what is the Big-O/Big-Theta?

A:

Given that the vector array is sorted, then the duplicate values must be close to each other. I would use a stack to solve the problem, that I would go through every element of the vector from beginning to end. When it’s reading the first element, push that element in the stack, and comes to the next element. If the next element is duplicate of the previous, pop the last one out, store that value in another list and count plus one, and skip to the next element. If the next element is not the same, push it, continue reading and so on. I would create another array to store the duplicate values and make a counter to sum the numbers of them (only if I have to output those duplicate values or a single counter is enough). The reading process would be linear time and the O(N) = N.