

Programming Question 5b)

Since the vector given is sorted, you can step through the vector number by number and compare each value at each index with the value at the next index. If they are equal, increase the count by one. Then, you can check to see if the new count is greater than the current maximum count. If it is, then replace the value with the value at that index.

If the value at the current index is not equal to the value at the next index, reset the count to 1 and step through again.

Once the maximum count is greater than the remaining number of values you can exit and return the current value as the mode because it's not possible for any other value to be the mode.

This ends up running in $O(n)$ time. With the constant being some fractional value based on the count of the mode and how early it can exit the vector because the mode count had exceeded the count of the remaining values. The best case would be if the first half of the array was the mode so it would exit after the half. The worst case would be $O(n)$ because it would have to traverse the entire vector if all the counts were 1.

The space complexity is only the size of the input vectors and a few constant values for the count and the value. If you don't include the input vector in the space it would be space complexity $O(1)$ constant, but if you do include the vector it would be $O(n)$.

Time Complexity = $O(n)$

Space Complexity = $O(1)$ (or $O(n)$ if the input vector is included)