Task 1 Part C

For part A, the size of the input word is N.

Since we can easily retrieve the last character of the input word, the time complexity of retrieving the last character uses O(1) time.

Now, we have to determine the time rest of the characters in the input word, and the size of the rest of the characters is N-1. Since you have to retrieve each letter in the rest of the characters, the time complexity of retrieving each lette r in the rest of the characters uses O(N^2) time because there are N-1 characters, and you have to do that retrieval pr ocess for N-1 times.

Lastly, since all of the mathematical operations in the formula take O(1) time (i.e. multiply by 33, the XOR operatio n, and determining the remainder of 2^M), the final time complexity is O(n^2) because O(n^2) multiply many O(1) i s still the same as O(n^2), and M does not change the time complexity at all.

For part B, since we're looking for the worst case possible, N would be 5, as the question gives the max range for N.

From part A, we know that the time complexity of any size input word N is O(n^2), and part A gives the equivalent hash value of the input word N. For this part, we have to figure out all 5 letter words that have the same hash value a s the given input K.

Since we have to search all possible equivalent hash value of a 5 letter word that matches K, we have to check each l etter in the 5 letter word, and since there are 26 letters in the alphabet, in the worst case, we have to search 26^5 time s.

Also, for each time that we have to check if a 5 letter word has the equivalent hash value as K, in the worst case, part A's algorithm would search 5^2 times (from O(n^2) where n=5 in this case).

Therefore, in total, in the worst case, part B's algorithm would have to search 26^5 * 5^2 = 297034400 times.

Hypothetically, if N didn't have a limit, then the time complexity would be O(26^n) * O(n^2) = O(26^n * n^2).