Technical Question 1:

In this problem, we are given a string text and a pattern and asked to see if the pattern is any anagram of the text. An anagram is a word or phrase made by transposing the letters of another word or phrase.

To solve this, we first sort the text string. We then loop through the length of the pattern.

We begin by selecting the middle letter of the string text and compare it to the value of the first letter in the pattern. If this letter matches we increment j by 1 (which stores the total amount of letters matched) and break out of the while clause, thus returning to the for loop.

However, if these letters do not match, we compare the values of the string text letter and the pattern. If the pattern letter is greater, we move the position of the string text one forward. If the pattern letter is lower, we move the position of the string text one backward. We then find the middle character of this range and again compare the letters via the loop.

Finally, we check the length of the matching anagram (variable j) and compare it to the length of the original pattern. If they are of equal length, we return true because all of the letters of the exist in the string text. Otherwise, we return false.

I have approached this problem in a similar way to conducting a binary search since the string is sorted in value. At worst, we will need to loop through each letter of the string which is a time efficiency of O(s) where s is the length of the original string. As for space efficiency, since we only record matched letters, this will be at most O(p) with p being the length of the pattern.