Here is the Pseudocode for the functions that I decided to implement in Java Script.

Function 1 FizzBuzz

The first function I decided to do was FizzBuzz. That is because this is a question that is common on Hacker Rank assessment for the Java programming language so I thought it might be useful. The idea is that you are given a range of numbers and have to return an array where each entry represents a number in the original range. If the number is a multiple of 3 you represent it with Fizz. If it is a multiple of 5 you represent it with Buzz. If it is a multiple of 3 and 5 you represent it with FizzBuzz. It if is not a multiple of 3 and not a multiple of 5 you represent it with the number itself. I thought this would be a good question as it requires loops and conditionals.

Define array returnArray

First I set up the array I will return

Now we go through the range

For number in range

Now we check the conditions

First I will start with if it is a multiple of both

so that we do not get caught in a less strict condition

If number is a multiple of 3 and if number is a multiple of 5

add FizzBuzz to returnArray

Else if number is a multiple of 3

add Fizz to returnArray

Else If number is a multiple of 5

add Buzz to returnArray

Else

add number to returnArray

Return returnArray

Function 2 isPalindrome

The second function I decided to do was a function to check if a word is a palindrome. I decided to do this as it is a common question people are told to practice for technical interviews. The idea is simple. If you reverse the letters in a word and end up with the same word it is a palindrome.

Most languages usually have a cheat to this question as some languages give string a function called reverse that reverses the order of the characters. Even if the language does not have this it is easy enough to do this ourselves. I decided to not do this fast way though and go with the method of solving where we compare the first character to the last character and so on.

First we set up a loop to go through the first half of the word

For (int I = 0; I < word.length ; I ++)

If word[I] not equal to word[word.length - I - 1]

return false

if we find one instance of them being different we can say it is not a palindrome

Return true

If we compare all complimentary characters and never reach the false condition then we have a palindrome and return true

Function 3 Binary Search

The next method I decided to make was a Binary Search that is because it is a common search strategy. There are 2 ways of implementing it. Iterative and recursive. I decided to do the iterative method as I do not like recursion. The idea of a binary search is to look through a sorted list to find an item. We look at the halfway point and decide if what we are searching for is bigger or smaller than it.

We then look to the right half if we are looking for a larger number and to the left for a smaller number. We keep going until we find the number or see that the number is not in the list.

Start = 0

End = array.lentgh - 1

For an iterative binary serach we keep going as long as the start is less that the End as if they cross over the item does not exits

While start < End

Mid = (End - Start)/2

Find the midway point

If array[Mid] equals to target

return True

Else if array[mid] < target

left half has been eliminated

start = mid +1

Else if array[mid= > target

right half has been eliminated

end = mid - 1

Return False

If we exit the while loop the number does not exist

Function 4 Frequency Counter

For the last I decided to do something that would make use of the dictionary. This is because it is a very useful data structure due to its speed. What I do is I take the characters in a word and place them in a dictionary where the values are the number of times they appear.

freqCounter= new empty dictionary

For letter in word

If letter not in freqcounter

freqCounter[letter] = 1

If the letter is not in the dictionary we add it and set the value to 1

Else

freqCounter[letter] = freqcounter[letter] + 1

if the word is already in the dictionary we increment the value