Critical Analysis

The aspects I found easy was understanding how to implement the selectionsort function as I found that relatively simple to understand after a while by adding comments next to each line to better understand how the actual code works as I found the concept of how it works very easy, but the actual code itself I found a little trickier. After understanding how the code worked, I found it easier to alter it to work for a Generic type and a string. The aspects I found a little trickier was the BigO notation, I understood that a line that does not repeat has a frequency of one but with the for loop for example was a little trickier for me but with time and re-watching the BigO notation recordings and asking matteo.c on Discord, I managed to understand it fully.

My general understanding of the BigO notation is that the first for loop in the program has a frequency of n+1 , this is because the stopping condition is I < n which means if I is less then n. This means it will run on how big n is and it is an addition + 1 due to the nature of the for loop that it will run an additional 1 time even though it has hit the stopping condition.. The for loop inside of the for loop which is the nested for loop would have the same frequency of n + 1 but since it is nested it will run how many times the original for loop will run so it is n\*(n+1).

My general understanding of how the selectionsort works is that the first for loop in the function is going through the array to see what position it needs to swap with. Then we initialise smallest as I and smallest represents the position we are going to sort. Then the for loop below it is aiming to find the smallest element by stating if j < a.Length to keep it in bounds of the array and then the if statement is checking with a[j] with j going up one every time the for loop is called to go through the array of a because of j = I which equals smallest. Then in this if statement we have the line < a[smallest] which is calling whatever number in the array has been declared as the smallest. So it is saying if the number in the array j is on right now, is less then what we have initialised smallest to be, smallest = j which makes smallest equal to j so we have the smallest. Then after that we use the swap function to swap a[i] with a[smallest] to organise it with whatever number of the array we are on with the smallest.