PA 1 Reflection Essay

The biggest issue I had with this assignment was figuring out what my time complexity was on my functions. I wasn’t sure if I was supposed to include the time complexity in my calculations of called functions in an outer function, and or include C++ library functions in my calculations. For instance, if I had a helper function, do I include the helper functions time complexity in my outer function’s calculation of time complexity. I decided to be specific as possible, and have you tell me whether I was too specific or comment on where I went wrong when I was that specific. I also had issues with how to do the time-complexity of my recursive helper function (huffmanEncodingMapFromTreeHelper). The way I tried to do it was by using the T(N)= # recursive calls \* (N/ how much you’re lessening the variable by) + amount of work per call. I figured my function was 2(N/2) + 1. I had two recursive calls each iteration, and was roughly halving the amount of work each time. And the amount of work was O(1) because I simply insert into a hash table when I do work in the function call. So I figured this function was O(Log(N)) due to roughly halving the variable each recursive call. I’m not that confident in most of my time-complexities and welcome any comments you can give to correct me.

At first look at this assignment I was concerned that its difficulty was above my current ability. But once I was able to conceptualize what it was we were trying to do, it became significantly easier. I think this and taking time to go through your starter code to understand the setup is the best advice I can give to a future student. Also I would recommend a student to create their own test cases. I found that at the beginning of the assignment I wasn’t that confident in the first two functions I created, but there wasn’t a test case setup to test those specific functions. So I created a test function to make sure they worked. And this allowed me to be more confident and get rid of errors in my code fast instead of them compounding and making debugging harder.