Question 1:

The reason that a simple change of variable names might throw a similarity search algorithm off is based on our algorithm. Through our algorithm, we are only detecting the neighbors or in other words, the similarity between each words. But while we are coding, the similarity is more focused on the framework and algorithm... We no longer focused on variables name or the disorder of each function. So the original way won't be accurate enough to detect coding plagiarism.

Question 2:

In order to use **jaccard distance**, **cosine distance**, **manhatten distance**... to find the differences, we can use another algorithm to find the similarity, but this time, we will not focus on the true differences. Instead, we will put more emphasis on the differences between each two pairs. For example, if similarity between code A and code B is 0.666, and meanwhile similarity between code A and code C, and similarity between code B and code C are approximately equal to 0.666, then this situation may occur when their code are different just due to the variables name, their framework and algorithm is not changed. By applying this method can help us more easily detect coding plagiarism, and with the help of LSH and MinHash, this would work more effectively.