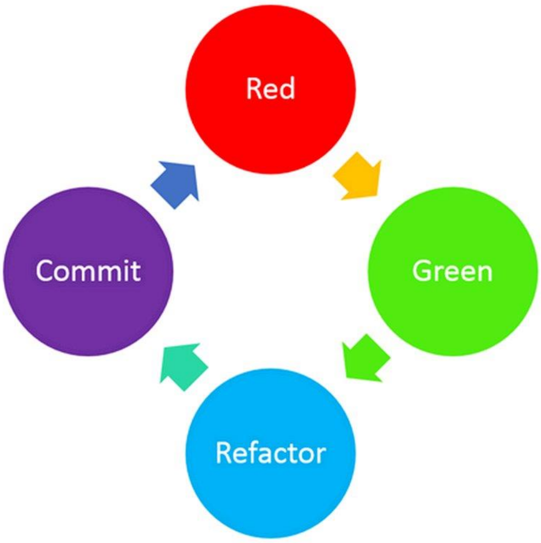
Design Spec

Introduction

As the behavior of this requirement (input and output) is very clear, it is fit for Test Driven Design - TDD. So test cases will be designed and created first, then implement these methods, followed by refactor the code. Additionally, there is no complex class or other component that need to be instantiated, mock is not necessary here. If there are further requirements coming, we should follow an iterative workflow as: Create Test Cases(Red) -> Implement features and make all test cases pass(Green) -> Refactor the code(Blue) -> Commit to git(Purple) -> Create Test Cases with new features.



Limitation

1. It is very useful to implement the feature that generates the string list based on an original string. However, it will out of scope for this release
2. As verifying the result using super long string and tens of or hundreds of strings in string list will be very hard to do manually, only short string and a couple of strings in the list will be used in the test cases to verify the correctness of the app.
3. If this is a normal project, I will do one commit per test case for code quality purpose, but as this is a test, to save time, I did batch commit.

Assumption

1. Assume file extension is .txt, each line only has one string
2. There is no duplicated string in the list
3. If two strings have no overlap, it will be concatenated as the order of their positions of input parameter. For instance, ConcatTwoString(s1,s2, overlap) will get s1 + s2 instead of s2 + s1.

Structure

This feature can break down into 4 steps underneath

1. Read file to string list.
2. Loop in the string list, compare every 2 strings, find the maximum overlapped substring, concatenated string and 2 original strings.
3. Insert concatenated string into the string list and remove the 2 original strings.
4. If the Count of string list is 1, output the last string, otherwise, go back to Step 2.

2 strings have mainly 3 cases:

1. String 1’s beginning has overlap with string 2’s end, or reverse, just concate using overlap
2. Two strings do not have overlap, concate string 1 with string 2 end to end
3. One string contains another, concatenated string is the longer one