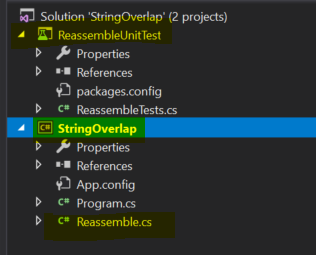
**Aderant Coding Assignment**

1. **GitHub Link**

<https://github.com/ShahUjval/Aderant>

StringOverlap – Project Contains the Main Logic

ReassembleUnitTest – Contains the Unit Tests.



1. **StringOverlap – Reassemble.cs**

Main logic contains within these two methods.

**FindLongestOverlap()** –

* Main Loop for the Greedy Match and Merge.
* We will Scan all the string fragments to find the Longest Overlap between any two fragments.
* Once found - we Merge them, and we repeat this process until we have only one fragment left with us.
* these loops run in **O(n^2)**

**MergeTheOverLapFragments()**

* Helper function to Merge the Overlap Fragments found after each pass.
* Each pass our collection of fragments will be reduced by one

1. **Algorithm and cases covered**

Step 1: get the position of “start char” of the second-string fragments (s2) in first-fragment string (s1)

S1 – “Go Bucks”

S2 – “Bucks -- Beat”

Step 2: find the overlap length (assuming s2 overlaps s1)

* Match to see of it overlaps
* If (yes)
  + Retain the overlap length and s1 , s2 indexes
* If (no)
  + Check to see if you have multiple start char in first string fragment.

Example:

S1 – “Go Boston Bucks”

S2 - “Bucks – Beat”

* + Advance to next start char and check the overlap once again.
  + Continue step 2
* If Failed to find the overlap length
  + Check if (s1 overlaps s2)
  + Continue step 2

|  |  |  |  |
| --- | --- | --- | --- |
| Cases | S1 | S2 | Merge string |
| S1 Contains S2 | “Go Bucks” | “Bucks” | S1 |
| S2 Contains S1 | “Bucks” | “Go Bucks” | S2 |
| S1 overlaps S2 | “**Bucks** -- Beat” | “Go **Bucks**” | “Go Bucks -- Beat” |
| S2 overlaps S1 | “Go **Bucks**” | “**Bucks** -- Beat” | “Go Bucks -- Beat” |
| Special case | “Go **B**oston **Bucks**” | “**Bucks** -- Beat” | “Go Boston Bucks -- Beat” |
| Final two (or no overlap found) fragments with no overlap | “Go Bucks” | “Boston” | S1 + S2 (if we are left with no overlap in all the string fragments) |

1. **Unit Tests**

I have added very basic tests which includes four set of test and one null test

