問題敘述：

從一個string到另外一個string，如何去定義這兩者之間的距離(edit distance)呢?從Morgan’s operations中，可以定義三個關於距離(edit distance)：

1. 改變字元
2. 從已知的string當中刪掉一個字元
3. 從已知的string當中插入一個字元

從距離(edit distance)的概念中，我們可以解決像是拼字問題、最長共同子序列(LCS)等等常見的問題。最終目標，是藉由上述三個操作，並對string *A*到string *B*操作Morgan’s operations的數量能夠達到最少。

解題敘述：

Step1. 首先，對於string *A* 到string *B* 的最少距離(edit distance)，先定義一個操作數最少的函數δ(*A*,*B*)=min { γ(T) | T is a trace from *A* to *B*}，並且用被線碰處到的字元強調一對一的關係，並且也強調不可讓線交錯，亦即，不允許交錯的區段搬移。

Step2. 為了能夠對δ(*A*,*B*)進行演算法分析，定義*D* ( i , j) =δ(*A*<i>,*B*<j>) = min{*D* (i-1, j-1) +γ(*A*<i>->*B*<j>), *D* (i-1, j) +γ(*A*<i>->empty string), *D* (i, j-1) +γ(empty string->*B*<j>)，min 中第一個是針對交換一個字元的情況下討論，第二個是針對刪除string *A*中的字元討論，第三個是針對插入一個string *B*中的字元討論。

Step3. 因為必須對string *A*的每個字元與string *B*的每個字元作分析，需要兩個 loop來執行，一個是跑string *A*的字元，一個是跑string *B*的字元。γ則針對想做的事情賦予它定義，比方說：做LCS(longest common sequence)，交換字元時，γ= 0，如果字元一樣，γ = 2，如果字元不一樣；刪除字元和增加字元時，γ= 2。

Step4. 在同時執行兩個 loop 執行的情況下，一直找出*D*(i, j) 對於兩個 string 的每個字元的操作數取最小值，即是其解。

時間複雜度分析：

由於要跑完兩個loop，而且兩個loop 分別跑的是string *A*的長度(|*A*|)和string *B*的長度(|*B*|)，在兩個loop 中，有四個敘述分別為交換字元、刪除字元及增加字元的*D* 值及對三個*D* 值取最小值。另外，再加上前面對*D*值的初始化和初始化兩個string 的*D* 值。整個演算法的時間複雜度為 1+ |*A*| + |*B*| + 4\*|*A*|\*|*B*| = *O*(|*A*|\*|*B*|)。

心得：

讀完wanger對於string-to-string correction的分析，再加上課堂上教授對於使用DP分析LCS的講解，在兩個string 中作線性比對及分析有更深的了解，希望日後再看到類似的問題時都能夠迎刃而解。