Assignment 2 Report

Time Analysis:
-The step of construction of the matrix (Tabulation) is done in O(n*m) because of the nested for loop where n is the length of string x and m is the length of string y.
-The tracing back of the matrix is done in O(n*m) in the worst ,it starts from the bottom right of the matrix till the top left
So the overall time complexity is O(n*m).

Code Explanation:
-The function uses tabulation dynamic programming approach where it calculates the scores of the subsequences of x and y and stores them in a matrix ('Matrix') .
-It iterates through the matrix finding the maximum score for each character in the sequence , it compares the score if it matches or if x is gap or if y is gap and gets the maximum of them when added to the previous cell (the score so far) ; for matching the previous cell is the upper right (diagonal),for x gap it is the left cell for y gap it is the upper cell , and then it stores the maximum in the cell.
- The second step is tracing back to get the alignments that result in the highest score. It starts from the bottom right till the top left and compares the value with the three scenarios (matching , gap x , gap y) . If the current value was the result from matching it adds both x and y characters to the resulted x and y respectively, if gap x it adds a gap to the resulted x and the y character to the resulted y and else it adds a gap to the resulted y and the character of x in the resulted x .

For the inputs:

x="TCCCAGTTATGTCAGGGGACACGAGCATGCAGAGAC"
y="AATTGCCGCCGTCGTTTTCAGCAGTTATGTCAGATC"

The result is :

x= "---TCCCAGTTATGTCAGGGGACACG-AG-CATG-CAGAGAC"
y= "AATTGCC-G-C-CGTC-GTTTTCA-GCAGTTATGTCAGAT-C"

Which has a score of 5 while the output given in the assignment description' score was -7.4, so the output of my alignment function is more accurate.