Assignment 3

Our goal in this assignment is to present an algorithm that determines whether a string w is an interleaving of two strings u and v.  We begin by comparing |u|+|v| to |w|. If they are unequal, then w cannot be an interleaving of u and v. If they are equal, we then consider a two-dimensional array of boolean values that has a size of (|u|+1) x (|v|+1).  Each ordered row corresponds with a character in u and each ordered column corresponds with a character in v. An exception is the first row and column which are set aside to represent cases with empty strings. We fill out this array row by row and column by column as follows:

* At (0,0), the value is always true.
* At (0,m) the value is true if the (0,m-1) entry is true and the m’th character in w is equal to the m’th character in v.
* At (n,0) the value is true if the (n-1,0) entry is true and the n’th character in w is equal to the n’th character in u.
* At (n,m), the value is true if either:
  + Entry (n-1, m) is true and the m+n’th character in w is equal to the nth character in u.

or

* Entry (n, m-1) is true and the m+n’th character in w is equal to the mth character in v.

Where  0 < n ≤ |u| and 0 < m ≤ |v|

    Using this array, we determine that w is an interleaving of u and v if entry (|u|, |v|), the bottom right entry in this two-dimensional array, is true.  To understand how this array works, we think of it like a directed graph. Each edge in the graph travels from a node either to the node below it or to the right of it.  A downward-traveling edge stands for a transition to the next character in u, while a right-traveling edge stands for a transition to the next character in v. Any connected path travelling from the origin point to the point (|u|, |v|) therefore represents some interleaving of u and v.  Each true value in this two-dimensional array stands for a node with an edge leading to it.