1). isEntwining(x, y, z) = isEntwining(x.substring(0, x.length()-1), y.substring(0, y.length()-1), z) || isEntwining(x.substring(0, x.length()-1), y, z.substring(0, z.length()-1));

2). The memory I will use is a 3-dimensional array (X.length() x y.length() x z.length()).

3). mem = (([0] x x.length()) x y.length()) x z.length().

isEntwining(x, y, z).

if mem[x][y][z] != 0
    return mem[x][y][z]

if x.length == 1 and (x == y or x == z)
    return True

if x[x.length()] is in y and y.substring(0, y.length()-1) is in x.substring(0, x.length()-1) and z is in x.substring(0, x.length()-1),
    isEntwining(x.substring(0, x.length()-1), y.substring(0, y.length()-1), z)

if x[x.length()] is in z and y is in x.substring(0, x.length()-1) and z.substring(0, z.length()-1) is in x.substring(0, x.length()-1),
    isEntwining(x.substring(0, x.length()-1), y, z.substring(0, z.length()-1).

Once the subproblems are solved, the recursion will pass the returned result to the first function call to obtain the result.

4). The running time is $O(x.length \times y.length \times z.length)$.