[Edit Distance](https://leetcode.com/problems/edit-distance/)

Operations: 1) insert 2) delete 3) replace

**Approach**: we have two strings we have to return minimum number of operations to convert str1 -> str2

=> We will start matching from the n-1, m-1

1. If str1[i] == str2[j] matches; we have to see next for i-1 and j-1
2. If they doesn't match: str1[i] != str2[j]

1) insert j char to str1: j has match to newly inserted char, now try matching j-1 char (i, j-1)

2) delete i char: means move i pointer 1 step back (i-1, j)

3) replace ith char with j char: they will definitely match, next try matching for i-1 & j-1 (i-1, j-1)

**Recursion**:

1. f(i, j)
2. Match- not match
3. Return min number of operations

**if(s[i]==t[j])**

**return f(i-1, j-1, str1, str2)**

**//if its not matching, take min of 3 operations**

**else**

**return 1 + min(f(i, j-1), min(f(i-1, j), f(i-1, j-1)))**

**=> Base case**

* if(i<0) return j+1;
* if(j<0) return i+1;

**Tabulation:**

1. Right shift of index to deal with i<0 and j<0 base cases: n->n+1, m->m+1
2. Use base case to make the base of the table
3. Use recurrence relation to form table i=1 to n, j=1 to m