```
int penaltyGap = 2; int penaltySub = 1; int penaltyMatch = 0;
// Initiate the table
int[][] table = new int[x.length() + 1][y.length() + 1];

 /*CONSTRUCT the table*/
// The first row - base case when there is no string 2
for (int i = 1; i <= x.length(); i++)
    table[i][0] = table[i - 1][0] + penaltyGap;
// The first column - base case when there is no string 1
for (int j = 1; j <= y.length(); j++)
    table[0][j] = table[0][j - 1] + penaltyGap;

// Calculate the value from Top, Left, and Diagonal
// then take the Min of them
for (int i = 1; i <= x.length(); i++) {
    for (int j = 1; j <= y.length(); j++) {
        int penaltyDiagonal =
            (x.charAt(i-1) == y.charAt(j-1)) ? penaltyMatch : penaltySub;
        int fromDiagonalBox = table[i-1][j-1] + penaltyDiagonal;
        int fromLeftBox = table[i][j-1] + penaltyGap;
        int fromTopBox = table[i-1][j] + penaltyGap;
        table[i][j] =
            Math.min(Math.min(fromDiagonalBox, fromLeftBox), fromTopBox);
    }
}

/*TRACEBACK the solution and put it on a stack*/
Stack<String> solutionStack = new Stack<String>();
//Trace back 2 strings from the end of table
int indexX = x.length();
int indexY = y.length();
while( indexX > 0 && indexY > 0) {
    //The MinEditedDistance was from diagonal, case match
    if (x.charAt(indexX-1) == y.charAt(indexY-1)) {
        solutionStack.push(
            x.charAt(indexX-1) + " " + y.charAt(indexY-1) + " " + "0");
        indexY--;
        indexX--;
    }
    //The MinEditedDistance was from diagonal, case substitute
    else if (table[indexX][indexY] == table[indexX-1][indexY-1] + 1) {
        solutionStack.push(
            x.charAt(indexX-1) + " " + y.charAt(indexY-1) + " " + "1");
        indexX--;
        indexY--;
    }
    //The MinEditedDistance was from the above box, case gap
    else if (table[indexX][indexY] == table[indexX][indexY-1] + 2) {
        solutionStack.push("-" + " " + y.charAt(indexY-1) + " " + "2");
        indexY--;
    }
    //The MinEditedDistance was from the left box, case gap
    else if (table[indexX][indexY] == table[indexX-1][indexY] + 2) {
        solutionStack.push(x.charAt(indexX-1) + " " + "-" + " " + "2");
        indexX--;
    }
}
//Leftover from the first string, there will be all spaces in 2nd string
while ( indexX > 0) {
    solutionStack.push(x.charAt(indexX-1) + " " + "-" + " " + "2");
    indexX--;
}
//Leftover from the 2nd string, there will be all spaces in 1st string
while (indexY > 0) {
    solutionStack.push("-" + " " + y.charAt(indexY-1) + " " + "2");
    indexY--;
}
```