

## 1. Code

```
EditStringTest
package report4;

import java.util.Scanner;

public class EditStringTest {
    public static void main(String[] args) {
        String X;
        String Y;
        int n, m;
        Scanner sc = new Scanner(System.in);

        System.out.println("두 문자열의 길이를 입력 : ");
        n = sc.nextInt();
        m = sc.nextInt();
        System.out.println("X 문자열 입력 : ");
        X = sc.next();
        System.out.println("Y 문자열 입력 : ");
        Y = sc.next();
        System.out.println();

        EditStringClass ec = new EditStringClass(n, m, X, Y);
    }
}
```

```
EditStringClass
package report4;

import java.util.Stack;

public class EditStringClass {
    private int cost[][]; // 비용을 저장할 2차원 배열
    private char edit[][]; // 연산을 저장할 2차원 배열
    private int Lx, Ly; // 문자열 x와 문자열 y의 길이를 저장할 변수
    private int VI, VD, VC, Vmin; // 각각 연산에 대한 비용을 저장 및
    최소값 저장에 위한 변수

    public EditStringClass(int n, int m, String x, String y) {
        if (x == y) {
            System.out.println("두 문자열이 같습니다."); // 두
            문자열이 같을 경우 종료
            return;
        }

        Lx = n; // x문자열의 길이인 n을 Lx에 저장
        Ly = m; // y문자열의 길이인 m을 Ly에 저장

        cost = new int[Lx + 1][Ly + 1]; // 비용배열 동적생성
        edit = new char[Lx + 1][Ly + 1]; // 연산배열 동적생성

        cost[0][0] = 0; // 첫번째 배열은 연산이 적용되지 않으므로
        비용이 0
        edit[0][0] = '-';
    }
}
```

```

for (int i = 1; i <= Lx; i++) { // 첫 번째 열은 삭제연산만 적용
    cost[i][0] = cost[i - 1][0] + 1;
    edit[i][0] = 'D';
}
for (int j = 1; j <= Ly; j++) { // 첫번째 행은 삽입연산만 적용
    cost[0][j] = cost[0][j - 1] + 1;
    edit[0][j] = 'I';
}

for (int i = 1; i <= Lx; i++) {
    for (int j = 1; j <= Ly; j++) {
        if (x.charAt(i - 1) == y.charAt(j - 1)) { //
비용없이 문자열을 변경할 수 있는지 확인
            cost[i][j] = cost[i - 1][j - 1];
            edit[i][j] = 'C';
        } else { // 최소 비용이 발생하는 경우 찾기
비용 +2
            VC = cost[i - 1][j - 1] + 2; // 교체
비용 +1
            VD = cost[i - 1][j] + 1; // 삭제 비용
            VI = cost[i][j - 1] + 1; // 삽입 비용 +1
            if (VC <= VD && VC <= VI) { // 교체
비용이 가장 작은 경우
                Vmin = VC;
                edit[i][j] = 'C';
            } else if (VD <= VC && VD <= VI) {
// 삭제 비용이 가장 작은 경우
                Vmin = VD;
                edit[i][j] = 'D';
            } else if (VI <= VC && VI <= VD) { //
삽입 비용이 가장 작은 경우
                Vmin = VI;
                edit[i][j] = 'I';
            }
            cost[i][j] = Vmin; // 최소비용을
cost[i][j]에 저장
        }
    }
}

System.out.println("<cost>"); // cost 그래프 출력
System.out.print(" ");
for (int i = 0; i <= Ly; i++) { // Y의 길이 = 열의 개수
    System.out.print(i + " ");
}
System.out.println();

for (int i = 0; i <= Lx; i++) {
    System.out.print(i + " "); // X의 길이 = 행의 개수
    for (int j = 0; j <= Ly; j++) {
        System.out.print(cost[i][j] + " ");
    }
    System.out.println();
}

System.out.println();
System.out.println("<edit>"); // edit 그래프 출력
System.out.print(" ");

```

```

        for (int i = 0; i <= Ly; i++) { // Y의 길이 = 열의 개수
            System.out.print(i + " ");
        }
        System.out.println();
        for (int i = 0; i <= Lx; i++) {
            System.out.print(i + " "); // X의 길이 = 행의 개수
            for (int j = 0; j <= Ly; j++) {
                System.out.print(edit[i][j] + " ");
            }
            System.out.println();
        }
        findpath(edit);
    }

    private void findpath(char edit[][]) { // 경로를 출력할 함수
        int i = Lx;
        int j = Ly;
        Stack st = new Stack(); // 스택 선언
        while (i >= 0 && j >= 0) { //
            st.push("(" + i + "," + j + ");");

            if (edit[i][j] == 'C') { // edit[i][j]가 교체면 i,j값 모두
                i--;
                j--;
            } else if (edit[i][j] == 'D') { // edit[i][j]가 삭제면 i값
                i--;
            } else // edit[i][j]가 삽입이면 i값 감소
                j--;
        }
        System.out.println();
        System.out.print("경로 : " + st.pop());
        while (!st.isEmpty()) { // pop하여 경로 순서대로 출력
            System.out.print(" -> " + st.pop());
        }
    }
}

```

감소

감소

## 2. 실행화면

X = "aabab" / Y = "babb"	
두 문자열의 길이를 입력 :	
5	
4	
X 문자열 입력 :	
aabab	
Y 문자열 입력 :	
babb	
<cost>	
0 1 2 3 4	
0	0 1 2 3 4
1	1 2 1 2 3
2	2 3 2 3 4
3	3 2 3 2 3
4	4 3 2 3 4
5	5 4 3 2 3
<edit>	
0 1 2 3 4	
0	- I I I I
1	D C C I I
2	D C C C C
3	D C D C C
4	D D C D C
5	D C D C C
경로 : (0,0) -> (1,1) -> (2,2) -> (3,3) -> (4,3) -> (5,4)	

X = "abcd" / Y = "efgh"	
두 문자열의 길이를 입력 :	
4	
4	
X 문자열 입력 :	
abcd	
Y 문자열 입력 :	
efgh	
<cost>	
0 1 2 3 4	
0	0 1 2 3 4
1	1 2 3 4 5
2	2 3 4 5 6
3	3 4 5 6 7
4	4 5 6 7 8
<edit>	
0 1 2 3 4	
0	- I I I I
1	D C C C C
2	D C C C C
3	D C C C C
4	D C C C C
경로 : (0,0) -> (1,1) -> (2,2) -> (3,3) -> (4,4)	

X = "bbb" / Y = "bbbaabbb"

두 문자열의 길이를 입력 :

3

8

X 문자열 입력 :

bbb

Y 문자열 입력 :

bbbaabbb

|

<cost>

	0	1	2	3	4	5	6	7	8
0	0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6	7
2	2	2	1	0	1	2	3	4	5
3	3	3	2	1	0	1	2	3	4

<edit>

	0	1	2	3	4	5	6	7	8
0	-	I	I	I	I	I	I	I	I
1	D	C	C	C	I	I	C	C	C
2	D	C	C	C	I	I	C	C	C
3	D	C	C	C	I	I	C	C	C

경로 : (0,0) -> (0,1) -> (0,2) -> (0,3) -> (0,4) -> (0,5) -> (1,6) -> (2,7) -> (3,8)

X = "abcde" / Y = "abcef"

두 문자열의 길이를 입력 :

5

5

X 문자열 입력 :

abcde

Y 문자열 입력 :

abcef

|

<cost>

	0	1	2	3	4	5
0	0	0	1	2	3	4
1	1	0	1	2	3	4
2	2	2	1	0	1	2
3	3	3	2	1	0	1
4	4	4	3	2	1	0
5	5	5	4	3	2	1

<edit>

	0	1	2	3	4	5
0	-	I	I	I	I	I
1	D	C	I	I	I	I
2	D	D	C	I	I	I
3	D	D	D	C	I	I
4	D	D	D	D	C	C
5	D	D	D	D	C	I

경로 : (0,0) -> (1,1) -> (2,2) -> (3,3) -> (4,3) -> (5,4) -> (5,5)

X = "ohohohoh" / Y = "hoho"

두 문자열의 길이를 입력 :

8

4

X 문자열 입력 :

ohohohoh

Y 문자열 입력 :

hoho

|

<cost>

	0	1	2	3	4
0	0	1	2	3	4
1	1	2	1	2	3
2	2	1	2	1	2
3	3	2	1	2	1
4	4	3	2	1	2
5	5	4	3	2	1
6	6	5	4	3	2
7	7	6	5	4	3
8	8	7	6	5	4

<edit>

	0	1	2	3	4
0	-	I	I	I	I
1	D	C	C	I	C
2	D	C	D	C	I
3	D	D	C	D	C
4	D	C	D	C	D
5	D	D	C	D	C
6	D	C	D	C	D
7	D	D	C	D	C
8	D	C	D	C	D

경로 : (0,0) -> (1,0) -> (2,0) -> (3,0) -> (4,1) -> (5,2) -> (6,3) -> (7,4) -> (8,4)