

str1 : a b f g h e
 str2 : c b g h e
 str3 : b f h e

LCS :

a	b	f	g	h	e
c	b	g	h	e	
b	f	h	e		

b	h	c
---	---	---

str1 \ str2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c							
b							
g							
h							
e							

재귀상, 0일 때 0일 : 0 으로 시작
 DP 의 크기 : $(str1 + 1) \times (str2 + 1)$
 $f_LCS = ""$

$(6+1) \times (5+1)$ 의 2차원 배열

str1 \ str2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
b							
g							
h							
e							

일치 하는 문자 X : 각 칸의 왼쪽 칸과 위쪽 칸중, 더 큰 값 선택

$f_LCS = ""$

str1 \ str2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
b	0	0	1	1	1	1	1
g							
h							
e							

언제 하는 문제 0 : 가장 큰 매칭 확인 시 ,

해당 칸의 왼쪽 위 (대각선) + 1

대입

f_LCS : "b"

str1 \ str2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
b	0	0	1	1	1	1	1
g	0	0	1	1	2	2	2
h							
e							

f_LCS : "b g"

str1 \ str2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
b	0	0	1	1	1	1	1
g	0	0	1	1	2	2	2
h	0	0	1	1	2	3	3
e							

f_LCS : "b g h"

ser1 \ ser2	0	a	b	f	g	h	e
0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0
b	0	0	1	1	1	1	1
g	0	0	1	1	2	2	2
h	0	0	1	1	2	3	3
e	0	0	1	1	2	3	4

f-LCS : "b g h e"

ser1 과 ser2 사 이 의 LCS 길이

f-LCS er ser3 의 LCS 길이 구하면 됨.

f-LCS \ ser3	0	b	g	h	e
0	0	0	0	0	0
b	0	1	1	1	1
f	0	1	1	1	1
h	0	1	1	2	2
e	0	1	1	2	3

result

	0	a	a	b	b	b	b
0	0	0	0	0	0	0	0
b	0	0	0	1	1	1	1
a	0	1	1	1	1	1	1
a	0	1	2	2	2	2	2
b	0	1	2	3	3	3	3
b	0	1	2	3	4	4	4

LCS: 4

bbaa → abbb