



#5582. 공통 부분 문자열

<https://www.acmicpc.net/problem/5582>

25.01.13



Problem 시간: 50분

#5582 공통 부분 문자열

공통 부분 문자열

성공 다국어

☆

한국어 ▾

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	21335	8743	6854	42.474%

문제

두 문자열이 주어졌을 때, 두 문자열에 모두 포함된 가장 긴 공통 부분 문자열을 찾는 프로그램을 작성하시오.

어떤 문자열 s 의 부분 문자열 t 란, s 에 t 가 연속으로 나타나는 것을 말한다. 예를 들어, 문자열 ABRACADABRA의 부분 문자열은 ABRA, RAC, D, ACADABRA, ABRACADABRA, 빈 문자열 등이다. 하지만, ABRC, RAA, BA, K는 부분 문자열이 아니다.

두 문자열 ABRACADABRA와 ECADADABRBCRDARA의 공통 부분 문자열은 CA, CADA, ADABR, 빈 문자열 등이 있다. 이 중에서 가장 긴 공통 부분 문자열은 ADABR이며, 길이는 5이다. 또, 두 문자열이 UPWJCIRUCAXIIRGL와 SBQNYBSBZDFNEV인 경우에는 가장 긴 공통 부분 문자열은 빈 문자열이다.

입력

첫째 줄과 둘째 줄에 문자열이 주어진다. 문자열은 대문자로 구성되어 있으며, 길이는 1 이상 4000 이하이다.

출력

첫째 줄에 두 문자열에 모두 포함된 부분 문자열 중 가장 긴 것의 길이를 출력한다.

예제 입력 1 복사

예제 출력 1 복사

ABRACADABRA
ECADADABRBCRDARA

5



Problem

1. 문제 요약

두 문자열에서 가장 긴 공통 부분 문자열을 찾아라. (Longest common substring)

- **문제 유형:** DP, 문자열(LCS)
- **목표:**
 - 두 문자열이 주어졌을 때, 두 문자열에 모두 포함된 가장 긴 공통 부분 문자열을 찾는 프로그램을 작성하시오.
- **입력:**
 - 문자열 2개
- **출력:**
 - 가장 긴 공통 부분 문자열의 길이



Hint

1. DP를 사용한다.

- 분할 정복 기법을 사용하여 DP 테이블에 **공통 문자열의 길이**를 메모이제이션 해놓자.
- 각 칸이 의미하는 바는, **해당 행과 열까지의 문자만을 사용했을 때의 공통 부분 문자열의 갯수** 이다.

[illegible]



Hint

	A	B	R	A	C	A	D	A	B	R	A
E											
C											
A											
D											
A											
D											
A											
B											
R											
B											
C											
R											
D											
A											
R											
A											

	A	B	R
E	1	2	3
C	4	5	6

- 1번 : A 와 E 에서 공통 문자열의 갯수
- 2번 : AB 와 E에서 공통 문자열의 갯수
- 3번 : ABR 과 E에서 공통 문자열의 갯수
- 4번 : A 와 EC에서 공통 문자열의 갯수
- 5번 : AB 와 EC에서 공통 문자열의 갯수
- 6번 : ABR 과 EC에서 공통 문자열의 갯수

이런 식으로 모든 칸을 채워서 마지막 파란색 영역
(ABRACADABRA와 ECADADABRVCRDARA의 공통 부분 문자열의
길이)
의 값을 구할 수 있지만,



Hint



2. Longest common substring과 Longest common subsequence는 다르다.

- 반드시 공통 부분 문자열이 서로 붙어 있어야 한다. 부분 문자열 사이에 다른 문자가 들어오면 안 된다.

	A	B	C	D
B		1		
C			2	
D				3
E				

그러므로 공통 문자열의 갯수는 행, 열에 동일한 문자가 오는 경우 대각선값을 기준으로 하나씩 늘어난다.
따라서 dp 테이블에 값을 채울 때 행과 열이 서로 다른 문자면 0, 같은 문자면 현재 위치가 (r, c) 일 때, $(r - 1, c - 1)$ 의 값 + 1로 초기화하면 된다



Hint

Longest common substring의 특성을 고려하여 진행한 DP 테이블 설정.

	A	B	R	A	C	A	D	A	B	R	A
E											
C					1						
A	1			1	2		1			1	
D						3					
A	1			1	1		4			1	
D						2					
A	1			1	1		3			1	
B		2							4		
R			3							5	
B		1							1		
C					1						
R			1							1	
D						1					
A	1			1	1		2			1	
R			1							1	
A	1			2							2

		A	B	R	A	C	A	D	A	B	R	A
		0	0	0	0	0	0	0	0	0	0	0
E	0											
C	0				1							
A	0	1			1	2		1			1	
D	0					3						
A	0	1			1	1		4			1	
D	0					2						
A	0	1			1	1		3			1	
B	0		2						4			
R	0			3						5		
B	0		1						1			
C	0				1							
R	0			1							1	
D	0					1						
A	0	1			1	1		2			1	
R	0			1							1	
A	0	1			2							2

대각선 계산을 위한 여분 공간

실제 계산 영역

이때, 행이 0이거나 열이 0인 경우, 두 문자열의 문자가 같을 때 $(r - 1, c - 1)$ 의 값 + 1 을 사용할 수 없으므로,

대각선 계산을 위한 여분 공간을 DP테이블에 추가하거나,

조건문으로 i와 j가 0일때를 분기하자. if $i == 0$ or $j == 0$: $dp[i][j] = 1$

🌟🌟 Solution

```
string1 = input()
string2 = input()

dp = [[0] * len(string1) for _ in
range(len(string2))]

for i in range(len(string2)):
    for j in range(len(string1)):
        # 두 문자열의 i번째 문자와 j번째 문자가 같을때
        if string2[i] == string1[j]:
            # i와 j가 0인 경우 분리
            if i > 0 and j > 0:
                dp[i][j] = dp[i-1][j-1] + 1
            else:
                dp[i][j] = 1
        else:
            dp[i][j] = 0
    # dp테이블에서 최댓값 찾기
print(max(max(row) for row in dp))
```

시간 복잡도

- DP테이블 설정: $O(N * N)$ – N은 최대 4000이므로 16,000,000
- 최댓값 찾기: $O(N * N)$ – 각 row당 $O(N)$ 이므로 16,000,000
- 최대 시간 복잡도: 32,000,000이므로 시간 제한 2초(2억번)안에 가능



추가 내용

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
88593511	jsy4407	5582	맞았습니다!!	236628 KB	424 ms	PyPy3 / 수정	392 B	1시간 전

Hint의 방식으로 진행할 경우, 256mb 중 236mb만 사용하여 메모리 제한을 통과하는데, 다른 LCS문제에서는 메모리 제한을 적게 걸어놓는 경우가 있음. 아래는 DP테이블을 1차원 리스트로 만드는 경우.

```
def get_longest_common_substring(x, y):
    x = " " + x
    y = " " + y
    n = len(x)
    m = len(y)
```

```
max_length = 0
dp = [0] * n # 1차원 리스트
```

```
for r in range(1, m):
    saved = [0] * n # 현재 행의 정보를 담을 리스트
    for c in range(1, n):
        if x[c] == y[r]:
            saved[c] = dp[c - 1] + 1
            max_length = max(max_length, saved[c])
    dp = saved[:] # 현재 행의 정보를 이전 행으로써 저장
return max_length
```

```
str1 = input()
str2 = input()
```

```
print(get_longest_common_substring(str1, str2))
```

출처:

<https://thisismi.tistory.com/entry/백준-5582번-공통-부분-문자열-파이썬-풀이>



Assignment

백준 #20542. 받아 쓰기 (골드3)

문자열 + DP

문자열과 DP를 다루는 문제.

점화식을 작성할 때, 삭제, 추가, 변환 각 경우마다 현재 좌표(r, c)에서 어디를 참조해야 할지 생각해보자.