

Report for ALGORITHM

# Longest Common Subsequence (LCS) problem

NOVEMBER 2020

Department of Computer Science  
Chungbuk National University

# Contents

Contents	ii
Problem definition	iii
Recurrence relation with proof	iv
Pseudocode	v
Time Complexity	vi
Example	vii
Additional Part	viii

# Problem Definition

## Original Prob.

Given two strings  $x = x_1x_2 \cdots x_n$  and  $y = y_1y_2 \cdots y_m$ , we wish to find the length of their *longest common subsequence*, that is, the largest  $k$  for which there are indices  $i_1 < i_2 < \cdots < i_k$  and  $j_1 < j_2 < \cdots < j_k$  with  $x_{i_1}x_{i_2} \cdots x_{i_k} = y_{j_1}y_{j_2} \cdots y_{j_k}$ . Show how to do this in time  $O(mn)$ .

## Translate.

$x = x_1x_2 \cdots x_n$ 와  $y = y_1y_2 \cdots y_m$ 로 두 문자열이 주어진다. 이때,  $i_1 < i_2 < \cdots < i_k$ 이고  $j_1 < j_2 < \cdots < j_k$ 인  $i$ 와  $j$ 에 대하여  $x_{i_1}x_{i_2} \cdots x_{i_k} = y_{j_1}y_{j_2} \cdots y_{j_k}$ 를 만족하는 가장 큰 수  $k$ 를 구하는 알고리즘을 시간복잡도  $O(mn)$ 로 구현하시오.

## Example.

$x = \underline{a}bcdefgh$

$y = aefw\underline{a}sbcd$

Common sequences :  $\epsilon$  ( $k = 0$ ), "a" ( $k = 1$ ), "e" ( $k = 1$ ), "f" ( $k = 1$ ), "b" ( $k = 1$ ), "c" ( $k = 1$ ), "d" ( $k = 1$ ), "bc" ( $k = 2$ ), "bd" ( $k = 2$ ), "cd" ( $k = 2$ ), "ef" ( $k = 2$ ), "aef" ( $k = 3$ ), "bcd" ( $k = 3$ ), "abcd" ( $k = 4$ )

LCS = "abcd" ( $k = 4$ )

$x = \underline{a}bc$

$y = \underline{a}bc$

Common sequences :  $\epsilon$  ( $k = 0$ ), "a" ( $k = 1$ ), "b" ( $k = 1$ ), "c" ( $k = 1$ ), "ab" ( $k = 2$ ), "ac" ( $k = 2$ ), "bc" ( $k = 2$ ), "abc" ( $k = 3$ )

LCS = "abc" ( $k = 3$ )

$x = \underline{a}bcd$

$y = \underline{a}cb\underline{d}$

Common sequences :  $\epsilon$  ( $k = 0$ ), "a" ( $k = 1$ ), "b" ( $k = 1$ ), "c" ( $k = 1$ ), "d" ( $k = 1$ ), "ab" ( $k = 2$ ), "ac" ( $k = 2$ ), "ad" ( $k = 2$ ), "bd" ( $k = 2$ ), "cd" ( $k = 2$ ), "abd" ( $k = 3$ ), "acd" ( $k = 3$ )

LCS = "acd" or "abd" ( $k = 3$ )

$x = abcd$

$y = efgh$

Common sequences :  $\epsilon$  ( $k = 0$ )

LCS =  $\epsilon$  ( $k = 0$ )

# Recurrence Relation with Proof

## Recurrence Relation.

임의의 두 인덱스번호  $i(0 \leq i \leq n)$ 와  $j(0 \leq j \leq m)$ 에 대하여 LCS 문자열 자체와 크기에 대해 판단해야 하기 때문에, 2차원 list를 사용하도록 한다. 이를 바탕으로  $LCS(x[i], y[j])$ 를  $x[0 \sim i]$ ,  $y[0 \sim j]$ 까지의 LCS,  $len[i][j]$ 를 그 지점에서 가능한 LCS의 길이라고 하자. (인덱스 0은  $\epsilon$ 을 의미한다.)

Base Case,  $i = 0$  혹은  $j = 0$

어떤 문자열을 비교하더라도,  $\epsilon$ 과 비교하게 되므로  $len[i][j] =$  전부 0이 되고,  $LCS(x[i], x[j])$ 는  $\epsilon$ 이 들어간다.

Inductive Step,  $i, j > 0$ 이며, 2개의 케이스로 나눈다.

Case 1.  $x[i] = y[j]$

이 경우엔,  $x[i]$ 와  $y[j]$ 를 둘다 포함하지 않는 가장 큰 LCS의 길이인  $len[i-1][j-1]$ 에 1을 더해  $len[i][j]$ 에 대입하고,  $LCS(x[i], x[j])$ 에  $LCS(x[i-1], y[j-1]) + x[i]$ 를 추가한다.

proof)  $x[i]$ 와  $y[j]$ 를 둘다 포함하지 않는 LCS 중 가장 큰 수인  $len[i-1][j-1]$ 이 아닌 어떤  $len[i-k][j-l]$ 이 존재한다면, substring인  $LCS(x[1 \sim i-1], y[1 \sim j-1])$ 의 크기보다  $LCS(x[1 \sim i-k], y[1 \sim j-l])$ 의 크기가 더 커야 하는데,  $k > 1, l > 1$ 이므로  $i-1 > i-k, j-1 > j-l$ 이 되어  $LCS(x[1 \sim i-k], y[1 \sim j-l])$ 이  $LCS(x[1 \sim i-1], y[1 \sim j-1])$ 의 substring이 된다는 것을 알 수 있다. 이때 LCS의 정의에 의해  $LCS(x[1 \sim i-k], y[1 \sim j-l])$ 이 크기가 더 크다면, 모순이 된다는 사실을 확인할 수 있다.

Case 2.  $x[i] \neq y[j]$

이 경우엔,  $x[i]$  또는  $y[j]$ 를 포함하지 않는 LCS 중 가장 큰 두 LCS의 길이인  $len[i-1][j]$ ,  $len[i][j-1]$  두개 중 더 큰 수를  $len[i][j]$ 에 대입하고,  $LCS(x[i], x[j])$ 에 마찬가지로 대입한다.

proof)  $x[i] \neq y[j]$ 는,  $x[i]$ 와  $y[j]$ 를 추가해도 그 이전의 LCS들에는 변동이 일어나지 않는다는 것을 암시하므로, 고를 수 있는 가장 큰 LCS 중 한 개를 고르면 된다. 이때, 그 이전의 LCS중 가장 큰것은 둘 다 미포함한 LCS 중 가장 큰 LCS인  $LCS(x[i-1], y[i-1])$ 를 substring으로 갖는  $LCS(x[i], y[j-1])$ 과  $LCS(x[i-1], y[j])$ 이므로, 그 둘을 비교해 더 큰 값을 대입하면 된다.

$len[i][j]$ 와  $LCS(x[i], y[j])$ 를 구하는데 각각  $[i-1][j-1], [i-1][j], [i][j-1]$ 번째의 값이 필요하므로,  $i = 0, j = 0$ 부터 row major order 순서대로 list를 채워나간다. 이후  $LCS(x[n], y[m])$ 까지 반복했다면 이를 반환하여 결과를 출력한다.

# Pseudocode

By python-like grammer.

```
def LCS(x, y):
    len = [[0 for temp_1 in 0 to m] for temp_2 in 0 to n]
    lcs = [[] for temp_1 in 0 to m] for temp_2 in 0 to n]
    for i in 1 to n:
        for j in 1 to m:
            if x[i] == y[i]:
                len[i][j] = len[i-1][j-1] + 1
                lcs[i][j] = lcs[i-1][j-1]
                lcs[i][j].push(x[i])
            else:
                len[i][j] = len[i][j-1] if len[i][j-1] > len[i-1][j] else len[i-1][j]
                lcs[i][j] = lcs[i][j-1] if len[i][j-1] > len[i-1][j] else lcs[i-1][j]
    return (len[n][m], lcs[i][j])
```

# Time complexity

In function.

$n * m$  크기를 가진 두 list 생성 =  $O(n * m) * 2$

$n$ 번의 반복 안에서  $m$ 번의 반복 =  $\sum_1^n \sum_1^m$

두 값의 비교와 변수 대입 =  $O(1)$

$\therefore 2O(n * m) + \sum_1^n \sum_1^m O(1) = 2O(n * m) + O(n * m) = O(nm)$

# Example

## Used IDE & Compiler.

PyCharm professional 2020.2.2

Python 3.8

#PyCharm 내에서, Run 기능을 이용해 Compile 후 실행

## Used Code

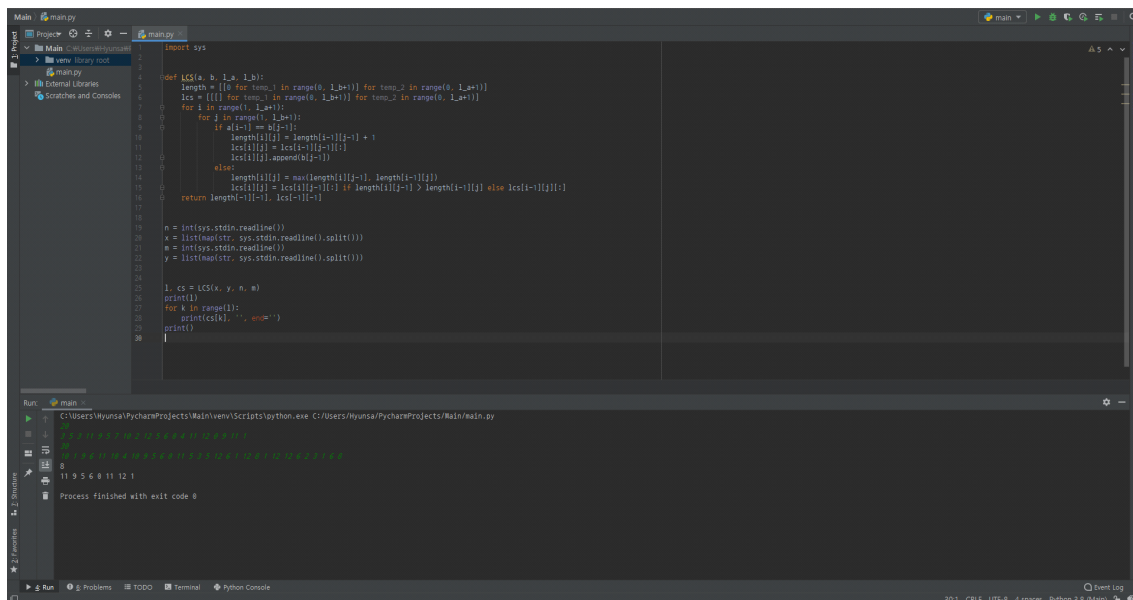
```
import sys

def LCS(a, b, l_a, l_b):
    length = [[0 for temp_1 in range(0, l_b+1)] for temp_2 in range(0, l_a+1)]
    lcs = [[] for temp_1 in range(0, l_b+1) for temp_2 in range(0, l_a+1)]
    for i in range(1, l_a+1):
        for j in range(1, l_b+1):
            if a[i-1] == b[j-1]:
                length[i][j] = length[i-1][j-1] + 1
                lcs[i][j] = lcs[i-1][j-1]
                lcs[i][j].append(a[i-1])
            else:
                length[i][j] = max(length[i][j-1], length[i-1][j])
                lcs[i][j] = lcs[i][j-1] if length[i][j-1] > length[i-1][j] else lcs[i-1][j]
    return length[-1][-1], lcs[-1][-1]

n = int(sys.stdin.readline())
x = list(map(str, sys.stdin.readline().split()))
m = int(sys.stdin.readline())
y = list(map(str, sys.stdin.readline().split()))

l, cs = LCS(x, y, n, m)
print(l)
for i in range(l):
    print(cs[i], '', end = '')
print()
```

## Screenshot.



# Additional Part

## Another example by using random.

해당 코드의 정확성을 판단하기 위하여 random함수를 통해 난수 조합을 뽑아 시도해보았다. 난수를 선정한 방법은 가독성을 위해 n,m = 1~100 을 가지도록 하였다. 이후 결과에서 LCS 부분을 굵기 효과를 주었다.  
사용된 환경은 위의 예제와 동일하며, 변경된 코드는 다음과 같다.

## Modified Code

```
import random

def LCS(a, b, l_a, l_b):
    length = [[0 for temp_1 in range(0, l_b+1)] for temp_2 in range(0, l_a+1)]
    lcs = [[[[] for temp_1 in range(0, l_b+1)] for temp_2 in range(0, l_a+1)]
    for i in range(1, l_a+1):
        for j in range(1, l_b+1):
            if a[i-1] == b[j-1]:
                length[i][j] = length[i-1][j-1] + 1
                lcs[i][j] = lcs[i-1][j-1][:]
                lcs[i][j].append(b[j-1])
            else:
                length[i][j] = max(length[i][j-1], length[i-1][j])
                lcs[i][j] = lcs[i][j-1][:] if length[i][j-1] > length[i-1][j] else lcs[i-1][j][:]
    return length[-1][-1], lcs[-1][-1]

n = random.randrange(1, 101)
m = random.randrange(1, 101)
x = []
y = []
for i in range(n):
    x.append(random.randrange(1, n))
for i in range(m):
    y.append(random.randrange(1, m))

l, cs = LCS(x, y, n, m)
print(n, x)
print(m, y)
print(l)
for k in range(l):
    print(cs[k], '', end='')
print()
```

## Modified Example

```
7 [3, 5, 4, 3, 4, 6, 4]
12 [10, 1, 8, 1, 1, 6, 1, 6, 6, 5, 6, 9]
2
5 6
11 [1, 6, 9, 3, 7, 1, 2, 10, 1, 10, 10]
26 [18, 14, 4, 22, 25, 4, 20, 13, 24, 17, 25, 22, 8, 10, 4, 19, 2, 15, 13, 2, 16, 5, 15, 9, 9, 18]
1
9
77 [75, 51, 24, 21, 11, 9, 24, 10, 18, 46, 18, 12, 44, 63, 21, 66, 38, 58, 33, 61, 32, 21, 18, 36, 24,
14, 23, 17, 61, 4, 5, 69, 53, 74, 14, 34, 9, 61, 1, 44, 7, 9, 5, 56, 74, 68, 20, 72, 71, 53, 42, 30,
61, 38, 37, 13, 37, 24, 57, 18, 42, 37, 2, 65, 43, 73, 39, 56, 52, 32, 24, 56, 45, 48, 27, 41, 3]
7 [2, 2, 4, 2, 2, 1, 1]
2
4 1
```



<p>94 [6, 67, 87, 91, 19, 13, 83, 77, 74, 23, 83, 70, <b>24</b>, 59, 18, 2, 81, 20, <b>27</b>, 80, 71, 43, 83, 3, <b>24</b>, 9, <b>47</b>, 40, 84, 56, 12, 55, 3, 49, 85, 72, 43, 62, 88, 55, <b>29</b>, 89, 31, 65, 71, 12, <b>44</b>, 76, 72, 42, 38, 76, 72, 19, 50, 13, 33, 54, 65, 82, 87, 51, 43, 37, 64, 80, 49, 39, 65, 90, 3, 85, <b>47</b>, 65, 93, 14, 84, 64, 72, 29, 80, 18, 37, 50, 56, 65, 80, 45, 59, 56, 88, 37, 57, 14]</p> <p>60 [53, 6, 21, 31, 1, 47, 53, 5, <b>24</b>, 8, 28, <b>27</b>, 42, 37, 41, 7, 12, 11, 25, 36, 26, 52, <b>24</b>, 6, 32, 8, 57, 17, 41, 25, 18, 14, 44, 55, 9, 21, 33, 21, 11, <b>47</b>, <b>29</b>, 51, 50, <b>44</b>, 58, 3, 25, 11, 53, 14, 9, 9, <b>47</b>, 40, 33, 24, 38, 2, 31, 21]</p> <p>10 6 24 27 24 9 47 29 44 3 47</p>
<p>28 [1, 3, 27, 24, <b>16</b>, 10, 5, 8, <b>23</b>, 14, 13, <b>22</b>, 24, 8, <b>25</b>, 5, 13, 24, 2, <b>18</b>, 8, 27, <b>23</b>, 6, <b>15</b>, 6, 25, 1]</p> <p>93 [7, 79, 84, 34, 65, 34, 75, 56, 40, 91, 23, 64, 47, 12, <b>16</b>, 56, 36, 69, 21, 73, <b>23</b>, 29, 21, 72, 70, 89, 55, <b>22</b>, 22, 54, 31, 88, 76, 58, 63, 35, 54, 23, 22, 33, 21, 82, 63, 48, 26, 73, <b>25</b>, 34, 62, 53, 44, 41, 32, <b>18</b>, 37, 91, 69, 22, 10, 19, 72, <b>23</b>, 80, 86, 88, 17, 21, 7, 18, 82, 36, 89, 43, 82, 79, 79, 10, 18, 1, 51, 63, 23, 35, <b>15</b>, 11, 69, 64, 46, 48, 44, 55, 17, 44]</p> <p>7 16 23 22 25 18 23 15</p>
<p>36 [27, 7, 33, 24, 10, <b>14</b>, 17, 23, 26, 3, 18, 26, 21, 34, 21, 20, 2, 22, 23, 1, 19, 19, 23, 11, 9, 26, 26, 8, 9, <b>16</b>, 9, 19, 30, 9, 28, 12]</p> <p>18 [14, 12, 7, 8, 1, <b>14</b>, 14, <b>16</b>, 14, 5, 10, 4, 7, 13, 6, 9, 13, 17]</p> <p>4 7 14 16 9</p>
<p>95 [61, 9, 84, 88, 74, 66, 81, 57, 31, 51, 10, 1, 58, 92, 87, 57, 51, 90, 72, 79, 5, 72, <b>18</b>, 80, 89, 35, 50, 79, 10, 11, 58, 44, 16, 72, 35, 3, 75, 60, 83, 23, 39, 37, 84, 2, 86, 60, 50, 48, 42, 57, <b>11</b>, 77, 23, 73, 76, 52, 68, 19, 14, 68, 53, 76, 85, 62, 49, 78, 52, 70, 14, 16, 75, 90, 87, 89, 78, 84, 42, 7, 40, <b>20</b>, 23, 24, 6, 74, 54, 93, 72, 21, 83, 7, 46, 9, 48, 31, 11]</p> <p>32 [16, 21, 19, 9, 22, 4, 23, 27, 31, 4, <b>18</b>, 6, 2, 9, 21, 1, 17, <b>11</b>, 1, 12, <b>20</b>, 8, 12, 10, 11, 21, 14, 31, 7, 1, 4, 10]</p> <p>8 9 31 18 2 11 20 21 7</p>
<p>99 [<b>15</b>, 71, 5, 78, 37, 59, 4, 97, 72, 38, 9, 60, 27, 9, 98, 3, 62, 29, 57, 49, 46, 43, 38, 46, <b>15</b>, 36, 94, 71, 94, 75, 80, 37, 28, 69, 20, 98, 14, 38, 50, 73, 9, 9, 74, 40, 38, 48, 48, 23, 40, 49, 36, 59, 52, 80, 88, 74, 25, 72, 72, 5, 54, 94, 16, 94, 77, 8, 8, 89, 32, 3, 69, 36, 19, 40, 24, 18, 67, 77, 8, 83, 6, 38, 59, 71, 44, 49, 38, 34, 30, 47, 87, 97, 18, 58, 38, 44, 95, 46, 85]</p> <p>22 [18, 4, <b>15</b>, 8, 20, 11, 3, 16, 7, 16, <b>15</b>, 1, 10, 1, 16, 12, 10, 7, 5, 6, 2, 20]</p> <p>5 15 3 15 5 6</p>
<p>71 [12, 57, 24, 45, 47, <b>20</b>, 6, 36, 46, <b>25</b>, 7, 31, <b>24</b>, 14, 34, 54, 60, 41, 47, 70, 68, 52, 62, 44, <b>18</b>, 13, 47, 14, 39, 28, 48, 11, 19, 36, 3, 44, 46, 31, 53, 64, 19, 31, 33, 58, 5, 37, 16, 16, 44, 61, 33, 35, 24, 9, 42, 28, 5, 8, 44, 60, 31, 29, 14, 56, <b>22</b>, 52, 7, 51, 64, 50, 7]</p> <p>34 [5, 11, 15, 15, 1, 33, 19, 1, 25, 19, 7, 13, <b>20</b>, 16, 3, 28, <b>25</b>, 7, 8, 12, 15, 27, 11, <b>24</b>, <b>18</b>, 30, 10, 23, 1, <b>13</b>, 8, 30, <b>22</b>, 19]</p> <p>8 20 25 7 24 18 13 8 22</p>
<p>41 [1, 3, 28, <b>19</b>, 6, 26, 35, 40, 2, 4, <b>11</b>, 13, 19, 23, 23, 20, 32, 30, 37, <b>18</b>, 8, 21, 27, 1, 15, 34, 6, 36, 8, 5, 17, 18, 27, 23, 5, 13, 20, 26, 31, 37, 24]</p> <p>23 [2, 15, 16, 19, 5, 16, 17, 13, 16, 4, <b>11</b>, 3, 17, <b>18</b>, 14, 1, 4, 4, 3, 21, 11, 7, 9]</p> <p>5 19 4 11 18 21</p>