# Algorithm of Longest Common Sequence

**LCS-LENGTH (X, Y)**

```
 1. m ← length [X]
 2. n ← length [Y]
 3. for i ← 1 to m
 4. do c [i,0] ← 0
 5. for j ← 0 to m
 6. do c [0,j] ← 0
 7. for i ← 1 to m
 8. do for j ← 1 to n
 9. do if xᵢ= yⱼ
10. then c [i,j] ← c [i-1,j-1] + 1
11. b [i,j] ← "↖"
12. else if c[i-1,j] ≥ c[i,j-1]
13. then c [i,j] ← c [i-1,j]
14. b [i,j] ← "↑"
15. else c [i,j] ← c [i,j-1]
16. b [i,j] ← "← "
17. return c and b.
```

## Example of Longest Common Sequence

**Example:** Given two sequences X [1...m] and Y [1.....n]. Find the longest common subsequences to both.

| **x:** A | B | C | B | D | A | B |
|----------|---|---|---|---|---|---|
| **y:** B | D | C | A | B | A | |

```
here X = (A,B,C,B,D,A,B) and Y = (B,D,C,A,B,A)
    m = length [X] and n = length [Y]
    m = 7 and n = 6
Here x₁= x [1] = A    y₁= y [1] = B
    x₂= B   y₂= D
    x₃= C   y₃= C
    x₄= B   y₄= A
    x₅= D   y₅= B
    x₆= A   y₆= A
    x₇= B
Now fill the values of c [i, j] in m x n table
```

```
Initially, for i=1 to 7 c [i, 0] = 0
        For j = 0 to 6 c [0, j] = 0
```

That is:

| i \ j | | 0 | 1 B | 2 D | 3 C | 4 A | 5 B | 6 A |
|---|---|---|---|---|---|---|---|---|
| | | $y_i$ | | | | | | |
| 0 | $X_i$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | A | 0 | | | | | | |
| 2 | B | 0 | | | | | | |
| 3 | C | 0 | | | | | | |
| 4 | B | 0 | | | | | | |
| 5 | D | 0 | | | | | | |
| 6 | A | 0 | | | | | | |
| 7 | B | 0 | | | | | | |

**Now for i=1 and j = 1**

   $x_1$ and $y_1$ we get $x_1 \neq y_1$ i.e. A ≠ B

And    c [i-1,j] = c [0, 1] = 0

   c [i, j-1] = c [1,0 ] = 0

That is, c [i-1,j]= c [i, j-1] so c [1, 1] = 0 and b [1, 1] = ' ↑ '

**Now for i=1 and j = 2**

$x_1$ and $y_2$ we get $x_1 \neq y_2$ i.e. A ≠ D

   c [i-1,j] = c [0, 2] = 0

   c [i, j-1] = c [1,1 ] = 0

That is, c [i-1,j]= c [i, j-1] and c [1, 2] = 0 b [1, 2] = ' ↑ '

**Now for i=1 and j = 3**

   $x_1$ and $y_3$ we get $x_1 \neq y_3$ i.e. A ≠ C

   c [i-1,j] = c [0, 3] = 0

   c [i, j-1] = c [1,2 ] = 0

so c [1,3] = 0      b [1,3] = ' ↑ '

**Now for i=1 and j = 4**

   $x_1$ and $y_4$ we get. $x_1$=$y_4$ i.e A = A

   c [1,4] = c [1-1,4-1] + 1

         = c [0, 3] + 1

         = 0 + 1 = 1

   c [1,4] = 1

   b [1,4] = '  ↖  '

**Now for i=1 and j = 5**

$x_1$ and $y_5$ we get $x_1 \neq y_5$

c [i-1,j] = c [0, 5] = 0

c [i, j-1] = c [1,4 ] = 1

Thus c [i, j-1] >  c [i-1,j] i.e. c [1, 5] = c [i, j-1] = 1. So b [1, 5] = '←'

**Now for i=1 and j = 6**

$x_1$ and $y_6$ we get $x_1 = y_6$

c [1, 6] = c [1-1,6-1] + 1

= c [0, 5] + 1 = 0 + 1 = 1

c [1,6] = 1

b [1,6] = '  ↖  '

| j | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| i | $y_i$ | B | D | C | A | B | A |
| 0  $X_i$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1  A | 0 | ↑ 0 | ↑ 0 | ↑ 0 | ↖ 1 | ← 1 | ↖ 1 |
| 2  B | 0 | | | | | | |
| 3  C | 0 | | | | | | |
| 4  B | 0 | | | | | | |
| 5  D | 0 | | | | | | |
| 6  A | 0 | | | | | | |
| 7  B | 0 | | | | | | |

**Now for i=2 and j = 1**

We get $x_2$ and $y_1$ B = B i.e.  $x_2 = y_1$

c [2,1] = c [2-1,1-1] + 1

= c [1, 0] + 1

= 0 + 1 = 1

c [2, 1] = 1 and b [2, 1] = ' ↖ '

Similarly, we fill the all values of c [i, j] and we get

| j → / i ↓ | | 0 | 1 (B) | 2 (D) | 3 (C) | 4 (A) | 5 (B) | 6 (A) |
|---|---|---|---|---|---|---|---|---|
| 0 | $X_i$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | A | 0 | ↑0 | ↑0 | ↑0 | ↖1 | ←1 | ↖1 |
| 2 | B | 0 | ↖①(1) | ←1 | ←1 | ↑1 | ↖2 | ←2 |
| 3 | C | 0 | ↑1 | ↑1 | ↖②(2) | ←2 | ↑2 | ↑2 |
| 4 | B | 0 | ↖1 | ↑1 | ↑2 | ↑2 | ↖③(3) | ←3 |
| 5 | D | 0 | ↑1 | ↖2 | ↑2 | ↑2 | ↑3 | ↑3 |
| 6 | A | 0 | ↑1 | ↑2 | ↑2 | ↖3 | ↑3 | ↖④(4) |
| 7 | B | 0 | ↖1 | ↑2 | ↑2 | ↑3 | ↖4 | ↑4 |

**Step 4: Constructing an LCS:** The initial call is PRINT-LCS (b, X, X.length, Y.length)

```
PRINT-LCS (b, x, i, j)
  1. if i=0 or j=0
  2. then return
  3. if b [i,j] = ' ↖ '
  4. then PRINT-LCS (b,x,i-1,j-1)
  5. print x_i
  6. else if b [i,j] = ' ↑ '
  7. then PRINT-LCS (b,X,i-1,j)
  8. else PRINT-LCS (b,X,i,j-1)
```

**Example:** Determine the LCS of (1,0,0,1,0,1,0,1) and (0,1,0,1,1,0,1,1,0).

**Solution:** let X = (1,0,0,1,0,1,0,1) and Y = (0,1,0,1,1,0,1,1,0).

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \quad \text{or} \quad j = 0 \\ c[i-1, j-1] + 1 & \text{if } i, j > 0 \quad \text{and} \quad x_i = y_i \\ \max(c[i, j-1], c[i-1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq y_i \end{cases}$$

We are looking for c [8, 9]. The following table is built.

x =(1,0,0,1,0,1,0,1)          y =(0,1,0,1,1,0,1,1,0)

| i \ j | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $y_i$ | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | |
| 0 | $x_i$ 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | ←1 | 1 | 0 | 1 | ←1 |
| 2 | 0 | 0 | 1 | 1 | 2 | ←2 | ←2 | 2 | ←2 | ←2 | 2 |
| 3 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | ←3 | ←3 | 3 |
| 4 | 1 | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | ←4 |
| 5 | 0 | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 |
| 6 | 1 | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 7 | 0 | 0 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 | 6 |
| 8 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 6 |

From the table we can deduct that LCS = 6. There are several such sequences, for instance (1,0,0,1,1,0) (0,1,0,1,0,1) and (0,0,1,1,0,1)
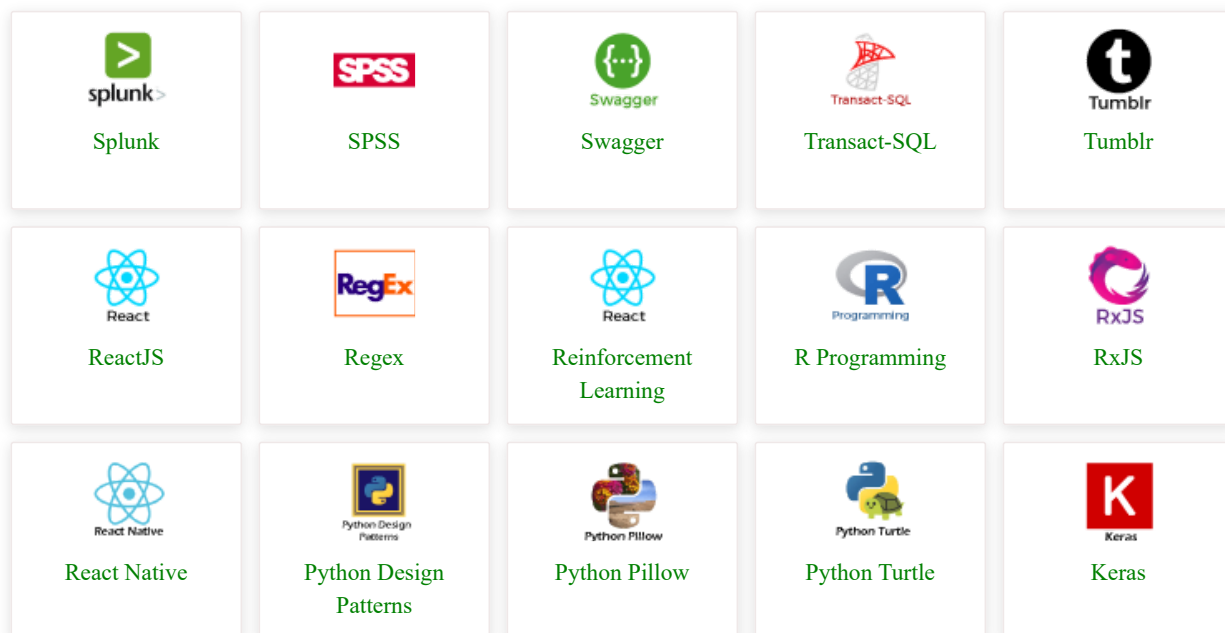
← Prev                                                                Next →

Feedback

- Send your Feedback to feedback@javatpoint.com

## Help Others, Please Share

## Learn Latest Tutorials

| | | | | |
|---|---|---|---|---|
| Splunk | SPSS | Swagger | Transact-SQL | Tumblr |
| ReactJS | Regex | Reinforcement Learning | R Programming | RxJS |
| React Native | Python Design Patterns | Python Pillow | Python Turtle | Keras |

## Preparation

| | | | | |
|---|---|---|---|---|
| Aptitude | Reasoning | Verbal Ability | Interview Questions | Company Questions |

## Trending Technologies
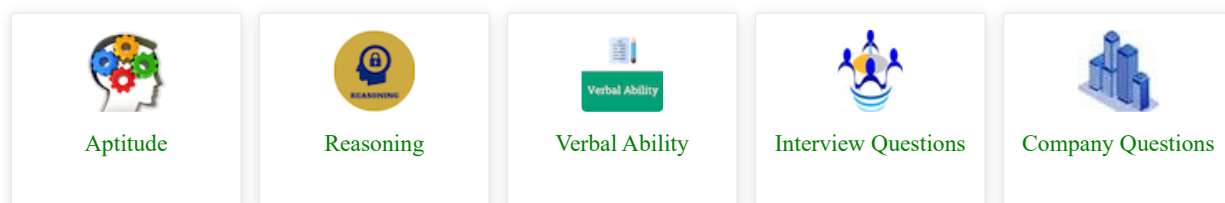
| | | | | |
|---|---|---|---|---|
| Artificial Intelligence | AWS | Selenium | Cloud Computing | Hadoop |
| ReactJS | Data Science | Angular 7 | Blockchain | Git |
| Machine Learning | DEVOPS | | | |
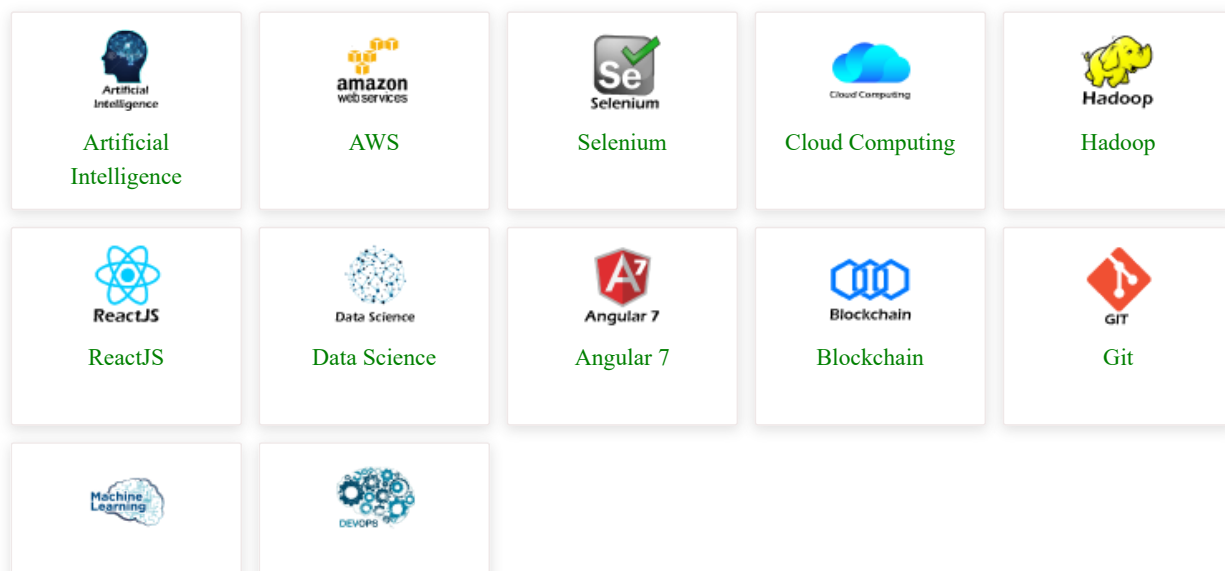
Machine Learning          DevOps

# B.Tech / MCA

DBMS

Data Structures

DAA

Operating System

Computer Network tutorial
Computer Network

Compiler Design tutorial
Compiler Design

Computer Organization and Architecture
Computer Organization

Discrete Mathematics Tutorial
Discrete Mathematics

Ethical Hacking
Ethical Hacking

Computer Graphics Tutorial
Computer Graphics

Software Engineering
Software Engineering

html tutorial
Web Technology

Cyber Security tutorial
Cyber Security

Automata Tutorial
Automata

C Language tutorial
C Programming

C++ tutorial
C++

Java tutorial
Java

.Net Framework tutorial
.Net

Python tutorial
Python

List of Programs
Programs

Control Systems tutorial
Control System

Data Mining Tutorial
Data Mining

Data Warehouse Tutorial
Data Warehouse