

# LONGEST COMMON SUBSEQUENCE

Manual de uso

### ¿Qué es la subsecuencia común más larga?

Como el nombre sugiere, la subsecuencia común más larga (LCS, por sus siglas en inglés) es la subsecuencia de mayor longitud entre dos cadenas.

Necesitamos definir qué es una **subsecuencia**.



Comenzar

### Subsecuencia

Digamos que tenemos una cadena "ABC".

Si borramos cero, uno o más caracteres de esta cadena, obtenemos una subsecuencia de esta cadena.

Así que las subsecuencias de la cadena ABC serán: { "A", "B", "C", "AB", "AC", "BC", "ABC", "" }

La cadena vacía también es una subsecuencia.

Para nuestra cadena "ABC", tenemos  $2^3 = 8$  subsecuencias

Entonces, si la longitud de la cadena es  $n$ , existen  $2^n$  subsecuencias de esa cadena.

Para nuestra cadena "ABC", la subsecuencia más larga es "ABC"

**Ejemplo:** Las subsecuencias comunes entre "HELLOM" y "HMLD" son "H", "HL", "HM", etc.

Presione el botón  
"Comenzar"

Alternar entre modo  
Top-Down y  
Bottom-Up

# Subsecuencia Común Más Larga

Acerca de Ayuda ?

[← Volver](#)

Cambiar a bottom-up

## Modo: Top-Down

Cadena 1

AGGTAB

Cadena 2

GXTXAYB

Velocidad animación

Calcular

### Llenado de tabla

```
const lcs = (X, m, Y, n, memo) => {  
  if (memo[m][n] > -1) {  
    return memo[m][n];  
  } else if (m === 0 || n === 0) {  
    memo[m][n] = 0;  
  } else if (X[m - 1] === Y[n - 1]) {  
    memo[m][n] = 1 + lcs(X, m - 1, Y, n - 1, memo);  
  } else {  
    memo[m][n] = Math.max(lcs(X, m - 1, Y, n, memo), lcs(X, m, Y, n - 1, memo));  
  }  
  return memo[m][n];  
}
```

Desde aquí es  
posible ingresar  
cadenas nuevas

# Subsecuencia Común Más Larga

[← Volver](#)

Cambiar a bottom-up

## Modo: Top-Down

Cadena 1

AGGTAB

Cadena 2

GXTXAYB

Velocidad animación

Calcula

### Llenado de tabla

```
const lcs = (X, m, Y, n, memo) => {  
  if (memo[m][n] > -1) {  
    return memo[m][n];  
  } else if (m === 0 || n === 0) {  
    memo[m][n] = 0;  
  } else if (X[m - 1] === Y[n - 1]) {  
    memo[m][n] = 1 + lcs(X, m - 1, Y, n - 1, memo);  
  } else {  
    memo[m][n] = Math.max(lcs(X, m - 1, Y, n, memo), lcs(X, m, Y, n - 1, memo));  
  }  
  return memo[m][n];  
}
```

Ajustar la velocidad  
de la animación

Subsecuencia Común Más Larga

Acerca de Ayuda ?

← Volver

Cambiar a bottom-up

### Modo: Top-Down

Cadena 1 AGGTAB

Cadena 2 GXTXAYB

Velocidad animación

Calcular

#### Llenado de tabla

```
const lcs = (X, m, Y, n, memo) => {  
  if (memo[m][n] > -1) {  
    return memo[m][n];  
  } else if (m === 0 || n === 0) {  
    memo[m][n] = 0;  
  } else if (X[m - 1] === Y[n - 1]) {  
    memo[m][n] = 1 + lcs(X, m - 1, Y, n - 1, memo);  
  } else {  
    memo[m][n] = Math.max(lcs(X, m - 1, Y, n, memo), lcs(X, m, Y, n - 1, memo));  
  }  
  return memo[m][n];  
}
```

Comenzar el cálculo  
con las cadenas  
ingresadas

# Subsecuencia Común Más Larga

Acerca de Ayuda ?

← Volver

Cambiar a bottom-up

Modo: Top-Down

Cadena 1

AGGTAB

Cadena 2

GXTXAYB

Velocidad animación

Calcular

## Llenado de tabla

```
const lcs = (X, m, Y, n, memo) => {  
  if (memo[m][n] > -1) {  
    return memo[m][n];  
  } else if (m === 0 || n === 0) {  
    memo[m][n] = 0;  
  } else if (X[m - 1] === Y[n - 1]) {  
    memo[m][n] = 1 + lcs(X, m - 1, Y, n - 1, memo);  
  } else {  
    memo[m][n] = Math.max(lcs(X, m - 1, Y, n, memo), lcs(X, m, Y, n - 1, memo));  
  }  
  return memo[m][n];  
}
```

Sí desea interrumpir  
el cálculo, presione  
el botón  
"interrumpir"

Subsecuencia Común Más Larga

← Volver

Cambiar a top-down

### Modo: Bottom-Up

Cadena 1 AGGTAB

Cadena 2 GXTXAYB

Velocidad animación

Calcular

**Interrumpir**

*	*	G	X	T	X	A	Y	B
*	0	0	0	0	0	0	0	0
A	0	↑ 0	↑ 0	↑ 0	↑ 0	↖ 1	↖ 1	
G								
G								
T								

**Llenado de tabla**

```

for (let i = 0; i <= m; i++) {
  for (let j = 0; j <= n; j++) {
    if (i === 0 || j === 0) {
      L[i][j] = 0;
    } else if (X[i - 1] === Y[j - 1]) {
      L[i][j] = L[i - 1][j - 1] + 1;
    } else {
      L[i][j] = Math.max(L[i - 1][j], L[i][j - 1]);
    }
  }
}

```

**Obtención de LCS**

```

while (i > 0 && j > 0) {

```