

# INF221 – Algoritmos y Complejidad

## Clase #12

### Subsecuencia común más larga

Aldo Berrios Valenzuela

Miércoles 7 de septiembre de 2016

#### 1. Subsecuencia común más larga

En inglés conocido como Longest Common Subsequence, LCS. La *idea* consiste en que tenemos dos archivos:  $X$  e  $Y$ , y queremos hacer `diff`<sup>1</sup> sobre ellos, es decir:

```
diff X Y
```

Para ello:

- Hallar la subsecuencia común más larga (LCS) entre ellos.
- Marcar líneas agregadas/borradas

**Ejemplo 1.1.** Supongamos que tenemos dos archivos:  $X$  e  $Y$ , cuyo contenido se muestra en el cuadro 1. En con-

$X$	$Y$
foo	bar
bar	xyzy
baz	plugh
quux	baz
windows	foo
	quux
	linux

Cuadro 1: Contenido de los archivos  $X$  e  $Y$ . Note que cada fila de la tabla representa una línea del archivo correspondiente.

secuencia, si hacemos un

```
diff X Y
```

obtenemos como resultado, la columna “Resultado” del Cuadro 2.

Explicamos con más detalle cómo funciona el algoritmo acorde a lo arrojado por el Cuadro 2:

- Comenzamos leyendo ambos archivos. Como podemos observar en el Cuadro 1, la primera línea de  $Y$  es `bar`. Por lo tanto, buscamos en  $X$  la primera línea que tenga `bar`. En  $X$ , la línea `bar` se encuentra en la línea 2, así que tendremos que eliminar el contenido de la línea 1 de  $X$ . Por lo tanto, en la columna “Resultado” agregamos `-bar`.
- En este momento, ambos archivos comienzan con `bar`, así que no realizamos ningún cambio y agregamos esto a “Resultado”. El Cuadro 3 muestra nuestra dónde estamos ubicados actualmente.

<sup>1</sup>pruebe `man diff` para ver qué es lo que hacer

Línea	X	Y	Resultado
1	foo	bar	- foo
2	bar	xyzzzy	bar
3	baz	plugh	+ xyzzzy
4	quux	baz	+ plugh
5	windows	foo	baz
6		quux	+ foo
7		linux	quux
			- windows
			+ linux

Cuadro 2: La columna “Resultado” se obtiene a través de: “Partiendo de X, ¿cómo obtengo Y”. Los (-) significa que debemos remover la línea asociada para construir Y a partir de X y los (+) es que debemos agregarla.

Línea	X	Y	Resultado
1	foo	bar	- foo
2	bar	xyzzzy	bar
3	baz	plugh	
4	quux	baz	
5	windows	foo	
6		quux	
7		linux	

Cuadro 3: Las celdas azules representan dónde estamos ya sea para el archivo X e Y. Por otro lado, las celdas remarcadas con color rojo representan nuestra siguiente coincidencia.

- Iteramos hasta encontrar la siguiente coincidencia (en este punto nos encontramos en la línea 2 de X y 1 de Y). La siguiente coincidencia es la línea baz. Como podemos observar en el Cuadro 3, para poder construir Y a partir de X agregamos las líneas xyzzzy y plugh. Luego, nos ubicamos en las líneas donde se produjo dicha coincidencia para ambos archivos y aprovechamos a de agregarlos (ver Cuadro 4).

Línea	X	Y	Resultado
1	foo	bar	- foo
2	bar	xyzzzy	bar
3	baz	plugh	+ xyzzzy
4	quux	baz	+ plugh
5	windows	foo	baz
6		quux	
7		linux	

Cuadro 4: Siga la misma convención de colores del Cuadro 3. Por otro lado, agregamos xyzzzy y plugh a “Resultado”.

- La siguiente coincidencia es quux.
  - En X, quux está inmediatamente después de la línea que tiene baz.
  - En Y, para llegar a la línea que tiene quux desde baz tendremos que pasar foo.

Dado lo anterior, en  $X$  tendremos que agregar la línea foo para poder construir  $Y$ . Los resultados se pueden apreciar en el Cuadro 5.

Línea	$X$	$Y$	Resultado
1	foo	bar	- foo
2	bar	xyzzy	bar
3	baz	plugh	+ xyzzy
4	quux	baz	+ plugh
5	windows	foo	baz
6		quux	+ foo
7		linux	quux

Cuadro 5: Siga la misma convención de colores del Cuadro 3. Por otro lado, agregamos foo a “Resultado”.

- Como podemos apreciar en el Cuadro 5, ya no tenemos más coincidencias. Por lo tanto, para terminar simplemente eliminamos windows y agregamos linux<sup>2</sup>. La tabla resultante puede verse a través del Cuadro 2.

■

### 1.1. Aspectos formales

Arreglos  $X[1, \dots, n]$ ,  $Y[1, \dots, m]$ , palabras sobre un alfabeto (“símbolo” es una línea). Hallar la secuencia de pares de índices  $(x_1, y_1), (x_2, y_2), \dots, (x_q, y_q)$  tales que /\* DUDA: ¿los  $x_i$  e  $y_i$  es el número de línea? \*/:

$$\begin{aligned} x_1 < x_2 < \dots < x_q & \quad ; \forall x_i \in \mathbb{N} \\ y_1 < y_2 < \dots < y_q & \quad ; \forall y_i \in \mathbb{N} \\ X[x_i] = Y[y_i] & \quad \text{para } 1 \leq i \leq q \end{aligned}$$

Interesa la secuencia más larga (máximo  $q$ , correspondiente a la cantidad coincidencias). Para ello, consideremos  $X[n]$ ,  $Y[m]$ . Tres opciones:

1.  $X[n]$  queda fuera de la subsecuencia. En consecuencia, lo marcamos con (-)
2.  $Y[m]$  queda fuera de la subsecuencia. Para efectos de diff marcamos con (+)
- + O ambos
3. Si  $X[n] = Y[m]$ , hacer  $x_q = n$ ,  $y_q = m$ .

Notar que si  $X[n] \neq Y[m]$  no pueden pertenecer ambos a la LCS.

Tres opciones. Subproblemas:

1.  $X[1, \dots, n-1]$ ,  $Y$
2.  $X$ ,  $Y[1, \dots, m-1]$
3.  $X[1, \dots, n-1]$ ,  $Y[1, \dots, m-1] \leftarrow X[n] = Y[m]$

Sea  $LCS(A, B)$  la subsecuencia común más larga entre  $A$  y  $B$ . Entonces /\* hasta el momento sólo nos interesa el largo de la subsecuencia óptima, después vemos como encontrar la secuencia \*/:

$$|LCS(X, Y)| = \max\{|LCS(X[1, \dots, n-1], Y)| + 0, |LCS(X, Y[1, \dots, m-1])| + 0, |LCS(X[1, \dots, n-1], Y[1, \dots, m-1])| + 1\}$$

$\rightsquigarrow$  Arreglo  $L[i, j]$ :

$$L[i, j] = |LCS(X[1, \dots, i], Y[1, \dots, j])| \quad (1.1)$$

<sup>2</sup>No nos arrepentimos de nada.

Sabemos  $L[0, j] = L[i, 0] = 0$ . Para calcular  $L[i, j]$  necesitamos  $L[i-1, j]$ ,  $L[i, j-1]$ ,  $L[i-1, j-1]$  (posiblemente). Llenar el arreglo, calculando  $L[i, k]$  para  $i$  de 1 a  $n$ , llenando los  $j$  de 1 a  $m$ .

~> Costo total es  $O(n \cdot m)$