

Taller en Sala Nro. 10 Programación Dinámica



En la vida real, la solución de programación dinámica del problema de la subsecuencia común más larga es utilizado en la implementación del comando diff, para comparación de archivos, disponible en sistemas Unix. También tiene muchas aplicaciones en bioinformática.



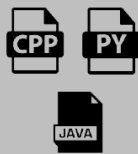
Trabajo en
Parejas



Hoy, plazo
máximo de
entrega



Docente entrega
código suelto en
GitHub



Sí .cpp, .py
o .java



No .zip, .txt,
html o .doc



Alumnos
entregan
código suelto
por GitHub

Ejercicio a resolver

1. El problema de la **subsecuencia común más larga** es el siguiente. Dadas dos secuencias, encontrar la **longitud** de la secuencia más larga presente en ambas. Una subsecuencia es una secuencia que aparece en el mismo orden relativo, pero no necesariamente de forma contigua

Teniendo en cuenta lo anterior, realicen un programa que calcule la longitud de la subsecuencia común más larga a dos cadenas de caracteres

2. **[Opcional]** Modifique el método para retornar, no la longitud de la subsecuencia común más larga, sino la subsecuencia común más larga (es decir, retornar *String* en lugar de *int*)

Ayudas para resolver el Ejercicio

Ayudas para el Ejercicio 1..... [Pág. 3](#)

Ayudas para el Ejercicio 2..... [Pág. 3](#)

Ayudas para resolver el Ejercicio 1



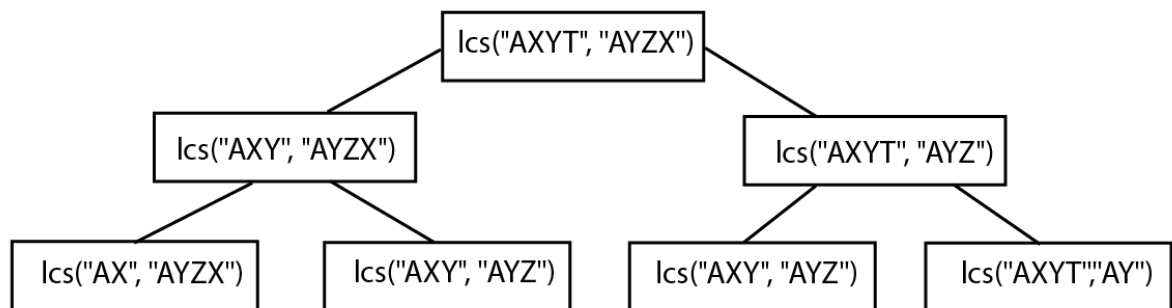
Como un ejemplo, “abc”, “abg”, “bdf”, “aeg” y “acefg” son subsecuencias de “abcdefg”. Entonces, para una cadena de longitud n existen 2^n posibles subsecuencias.

Ayudas para resolver el Ejercicio 2



Como un ejemplo, considere los siguientes ejemplos para el problema:

Para “ABCDGH” y “AEDFHR” es “ADH” y su longitud es 3. Para “AGGTAB” y “GXTXAYB” es “GTAB” y su longitud es 4. Una forma de resolver este problema es usando backtracking, como un ejemplo, para las cadenas “AXYT” y “AYZX”, dada una función recursiva lcs que resuelve el problema, se obtendría el siguiente árbol (parcial) de recursión:



Pista 1: Usando backtracking para ese problema, el problema $\text{lcs}(\text{"AXY"}, \text{"AYZ"})$ se resuelve dos veces. Si dibujamos el árbol de recursión completo, veremos que aparecen más y más problemas repetidos, así como en el caso de serie de Fibonacci. Este problema se puede solucionar guardando la soluciones, que ya se han calculado para los subproblemas, en una tabla; es decir, usando programación dinámica.



Pista 2: Complete el siguiente método:

```
// Precondición: Ambas cadenas x, y son no vacías
public static int lcsdyn(String x, String y) {
    ...
}
```

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 **Ext.** 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agende una cita con él a través de <http://bit.ly/2gzVg10> , en la pestaña *Semana*. *Si no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.*