



**Algoritmia**

Adrián Borges Cano

Marco González Martínez

Laboratorio: Miércoles 15:00-17:00

ÍNDICE

Tabla de contenido

[PROBLEMA 3 3](#_Toc101209593)

[PROBLEMA 6 7](#_Toc101209594)

# PROBLEMA 3

**Se tiene un número de Tam cifras almacenado en una cadena de texto; por ejemplo, la cadena dato = 1151451. Diseñar un algoritmo que mediante técnicas de Backtracking encuentre, de la manera más eficiente posible, todos los números distintos de N cifras que puedan formarse con los números de la cadena sin alterar su orden relativo dentro de la misma. Por ejemplo, si N = 4, son números válidos 1151, 1511 y 1541, pero no 4551 o 5411 que aunque pueden formarse con los dígitos de la cadena dato implican una reordenación.**

Descripción del algoritmo:

Para resolver el problema hemos realizado un algoritmo que encuentra todas las combinaciones de índices de posición disponibles respetando los siguientes criterios necesarios para el enunciado del problema:

* Se escogen tantos índices como longitud de cadena de solución requiera el problema.
* Los índices escogidos no estarán repetidos en una misma lista de índices.
* Los índices dentro de una misma lista de índices tendrán un orden ascendente.

Casos de prueba:

Código del algoritmo:

# PROBLEMA 6

**Se tiene la tabla de sustitución que aparece a continuación que se usa de la manera siguiente: en una cadena cualquiera, dos caracteres consecutivos se pueden sustituir por el valor que aparece en la tabla, utilizando el primer carácter como fila y el segundo carácter como columna. Por ejemplo, se puede cambiar la secuencia ca por una b, ya que M[c,a]=b. Implementar un algoritmo Backtracking que, a partir de una cadena no vacía texto y utilizando la información almacenada en una tabla de sustitución M, sea capaz de encontrar la forma de realizar las sustituciones que permite reducir la cadena texto a un carácter final, si es posible. Ejemplo: Con la cadena texto=acabada y el carácter final=d, una posible forma de sustitución es la siguiente (las secuencias que se sustituyen se marcan para mayor claridad):** acabada 🡪 acacda 🡪 abcda 🡪 abcd 🡪 bcd 🡪 bc 🡪 d.

Descripción del algoritmo:

Casos de prueba:

Código del algoritmo: