

# AUXILIAR 9: ALGORITMOS Y ESTRUCTURAS DE DATOS

PROFESOR: PATRICIO POBLETE  
AUXILIARES: IVÁN SÍPIRAN - MAURO ESCOBAR  
17 DE JUNIO DE 2010

**P1.** Calcule la función de fracaso del algoritmo KMP para el patrón de búsqueda:

`'alfalfa'`

y aplique el algoritmo para buscar el patrón dentro del texto:

`'alalfa alfa aalfalalfalfafala'`

Muestre paso a paso la alineación del patrón con el texto e indique los caracteres que se comparan.

**P2.** (i) Utilice el algoritmo BHM para buscar el patrón `'bcbcbc'` dentro del texto `'bacbcbcbc'`, mostrando paso a paso la ejecución.

(ii) Se desea encontrar un algoritmo de tipo BHM para buscar un patrón dentro de un texto, pero permitiendo que pueda haber una discrepancia en alguno de los caracteres (pero no más de una). Por ejemplo, se permite que `'hola'` calce con `'hora'`, pero `'hola'` no calza con `'hilo'`.

Se propone el siguiente algoritmo: se va comparando de derecha a izquierda hasta encontrar el *segundo* par de caracteres que no calcen. Si nunca ocurre esto (o sea, hay a lo más un par que no calce), se ha encontrado lo que se buscaba. En caso contrario, se desliza el patrón hacia la derecha usando la regla de BHM y se intenta de nuevo en esta posición.

Discuta si este algoritmo funciona o no.

**P3.** Sea  $A = a_1a_2 \dots a_n$  y  $B = b_1b_2 \dots b_n$  dos strings de largo  $n$ . Se define que  $B$  es una *rotación cíclica* de  $A$  si existe un índice  $k$ ,  $1 \leq k \leq n$ , tal que  $\forall i, 1 \leq i \leq n, a_i = b_{(k+i) \bmod n}$ .  
Describa un algoritmo que utilice KMP para determinar si  $B$  es una rotación cíclica de  $A$ .

**P4.** Se ha almacenado la representación (codificación) binaria de un conjunto de palabras en un árbol digital. Describa e implemente un algoritmo que imprima todas las codificaciones que posean un prefijo dado.

**P5.** Implemente la función `void imprimePrefijos(Nodo raiz-trie)`, que recibe como parámetro un trie (árbol digital) e imprime en pantalla todos aquellos prefijos tales que si inserta una nueva llave en el trie que posea alguno de éstos, la cantidad de nodos del trie no varía.  
La función debe tomar tiempo lineal con respecto al tamaño del trie.

**P6.** Se dice que un grafo no dirigido es *bipartito* si es posible separar sus nodos en dos subconjuntos  $A$  y  $B$  tal que todos los arcos del grafo van desde  $A$  a  $B$ , y no existen arcos que conecten nodos de  $A$  entre sí ni arcos que conecten nodos de  $B$  entre sí.  
Describa (en pseudo-código) un algoritmo que determine si un grafo es *bipartito*. El algoritmo debe operar en tiempo lineal en el tamaño del grafo, esto es,  $O(|V| + |E|)$ .