Práctica 2: Árboles de Sufijos

Javier Fumanal Idocin, 684229

Contenido

[Construcción del Árbol 4](#_Toc472806482)

[Buscar La Copia Más Larga 4](#_Toc472806483)

# Construcción del Árbol

Para construir el árbol se introduce cada sufijo de la palabra en el árbol. Para introducir cada sufijo se construye con el algoritmo de coste O(n2). Este algoritmo consiste en aplicar los siguientes pasos, empezando desde la raíz:

* Comprobamos si nuestro sufijo es distinto al vacío.
  + Si lo es: creamos un nodo cuya transición sea el símbolo de sufijo terminado y terminamos.
  + Si no lo es:
    - Buscamos un nodo hijo del nodo actual cuya transición tenga como carácter el primero de nuestro sufijo.
      * Si existe: eliminamos la primera letra de nuestro sufijo, y repetimos recursivamente el algoritmo con el nodo encontrado.
      * Si no existe: creamos el nodo. Luego eliminamos la primera letra de nuestro prefijo y repetimos el algoritmo recursivamente con el nodo recién creado.

Una vez creado el árbol, se tiene que reducir aun a su tamaño mínimo. El objetivo de esta técnica es que los nodos del árbol tengan varios hijos, o bien su único hijo indique fin de sufijo. Para ello se aplique lo siguiente, empezando por la raíz:

* Si nuestro nodo tiene varios hijos: aplicamos el algoritmo recursivamente a estos.
* Si nuestro nodo tiene un único hijo: comprobamos que sea el símbolo de finalización de sufijo.
  + Si lo es: no hacemos nada.
  + Si no lo es: añadimos el símbolo de nuestro nodo hijo al nodo actual, y todos los hijos del primero pasan a ser hijos del segundo. Una vez hecho esto, el nodo hijo se elimina, y se vuelve a aplicar recursivamente el algoritmo sobre el mismo nodo.

Una vez aplicado el algoritmo, todos los nodos o bien poseen dos hijos, o bien poseen un hijo que es el símbolo de finalización, o bien son nodos hoja, con el símbolo de finalización de sufijo.

# Buscar La Copia Más Larga

Para buscar la repetición más larga en una cadena de caracteres se busca el camino más profundo del árbol. Este algoritmo funciona debido a que el árbol creado tiene la siguiente propiedad: cada nodo del árbol que posee dos hijos es el camino común de al menos dos sufijos. Esto significa que si se busca el nodo con dos hijos más profundo del árbol, teniendo en cuenta que por profundidad se entiende longitud del sufijo, se encuentra la cadena repetida más larga, que es exactamente la que se busca.

Para encontrar este nodo, se realiza lo siguiente, empezando desde la raíz:

* Si el nodo sólo tiene un hijo (que será el símbolo de finalización de sufijo): devuelve 0. No se quiere contar el tamaño de este nodo ya que es la parte que no compartirán los dos sufijos, y por tanto no estará repetido.
* Si el nodo tiene varios hijos:
  + Se aplica recursivamente el algoritmo a cada uno de los hijos, sumando en cada uno de ellos la longitud del camino hasta el nodo actual.
  + Se compara la longitud de los caminos obtenidos. Se devuelve el camino mayor.

Una vez calculada la secuencia de nodos, es fácil reconstruir la cadena repetida pasando una vez por cada uno de ellos.

El coste total de esta operación es:

Coste de Construir el Árbol + Coste de la búsqueda del camino más profundo + Coste de visitar cada nodo del camino = O(n2) + O(n) + O(m) = O(n2)

Sin contar el coste de construir el árbol: O(n).

Además, se ha implementado una versión alternativa, basada en fuerza bruta y sin árboles de sufijos, que utiliza tres bucles anidados para calcular esta misma cadena. El coste de esta versión es O(n3).

# Buscar Todas las Copias Maximales

Para buscar todas las copias maximales, se divide el problema en dos partes:

* Se buscan todas las repeticiones presentes en la cadena, independientemente de si son maximales o no, junto con la posición en la que aparecen.
* Se filtran las repeticiones que aparecen más de una vez y aquellas que no son maximales.

## Búsqueda de Copias en el Árbol

Para buscar copias en el árbol se utiliza la misma propiedad que en el apartado anterior.

Como todos los nodos que poseen dos hijos son la parte común de dos sufijos distintos de la cadena, deben repetirse en algún momento de forma obligatoria. Se guarda en una lista el camino recorrido cada vez que el algoritmo se encuentra con un nodo con dos hijos al recorrer el árbol. Este recorrido empieza en la raíz, y se hace en inorden. Recorrer en este orden el árbol hace que los sufijos aparezcan ordenados en la lista sin ningún tipo de cálculo adicional, lo que hará más fácil duplicar las copias detectadas más de una vez. Además, para poder filtrar posteriormente las copias no maximales, guardaremos la posición de cada sufijo en la palabra original.

Coste de la operación: el recorrido se hace una única vez por cada nodo del árbol, luego es O(n).

## Filtrar Copias no Maximales y Repetidos

Lectura de Ficheros Fasta

Tiempos de Ejecución