Búsqueda en Texto

Universidad Tecnológica Metropolitana

Búsqueda en texto

Los algoritmos de búsquedas de texto en su propósito resuelven el problema de detectar una ocurrencia de una subcadena (llamada patrón o texto de búsqueda, de largo m) en una cadena de caracteres (denominada texto, de largo n).

Algoritmo de Fuerza Bruta

- Es el algoritmo ms simple posible.
- Consiste en probar todas las posibles posiciones del patrn en el texto.
- Requiere espacio constante.
- Realiza siempre saltos de un carcter.
- Compara de izquierda a derecha.

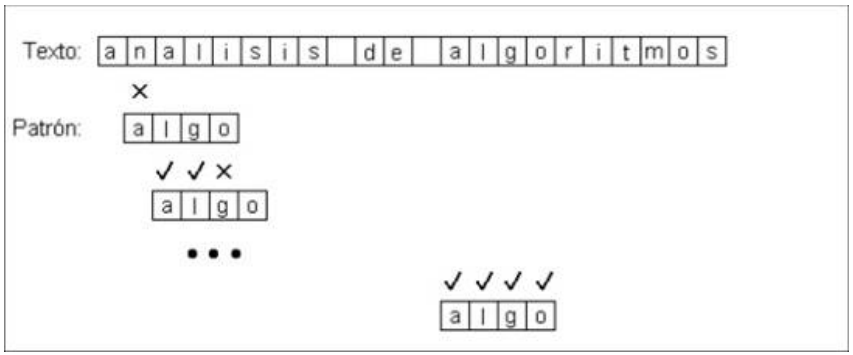
Complejidad

Dado un patrón de M caracteres de longitud, y un texto de N caracteres de longitud:

- Peor caso: compara el patrón con cada subcadena de texto de longitud M. Complejidad: O(MN)
- Mejor caso: encuentra el patrón en las primeras M posiciones del texto. Complejidad: O(N)

Ejemplo

Se alinea la primera posición del patrón con la primera posición del texto, y se comparan los caracteres uno a uno hasta que se acabe el patrón, esto es, se encontró una ocurrencia del patrón en el texto, o hasta que se encuentre una discrepancia.



Algoritmo Knuth Morris - Pratt

- Para hallar una subcadena en una cadena de texto, este algoritmo utiliza la información obtenida en los fallos anteriores, aprovechando la información que contiene el patrón con el cual se realiza la búsqueda (referente a esto se precalcula una tabla). Para de esa manera poder solo iterar una vez la revisión de la cadena de observación.
- Preprocesa el patrón para asociar a cada carácter que lo forma la longitud del prefijo más largo posible que sea a su vez un sufijo de la cadena de caracteres que precede al patrón y que los caracteres siguientes al prefijo y al sufijo sean diferentes.

Ejemplo de Proceso Previo

En general lo que realiza el algoritmo de proceso previo es contabilizar de forma secuencial los prefijos que existen en el patrón, con PP[0]=0.

j	0	1	2	3	4	5	6	7
P[j]	c	o	c	a	c	o	l	a

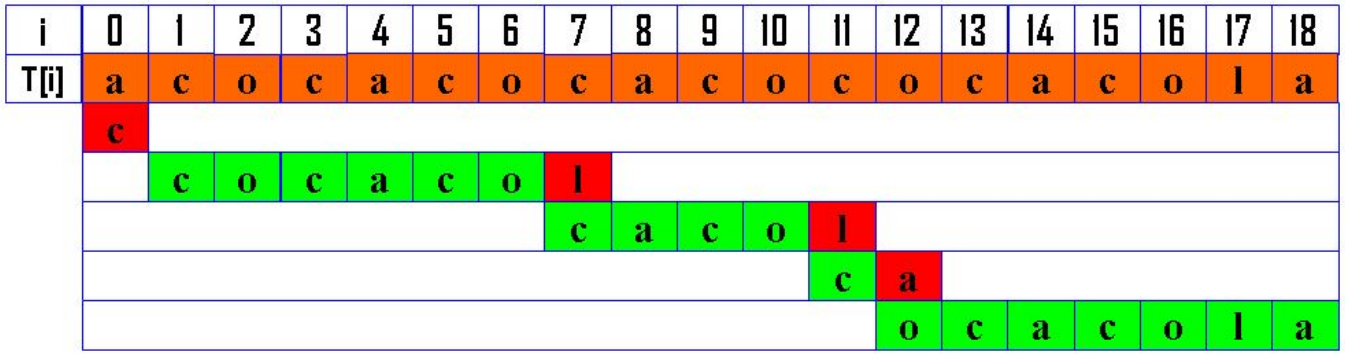
i	0	1	2	3	4	5	6	7
P[i]	c	o	c	a	c	o	l	a
PP[i]	0	0	1	0	1	2	0	0

PATRON PREFIJO

INDICE PROCESO PREVIO

Ejemplo del Algoritmo

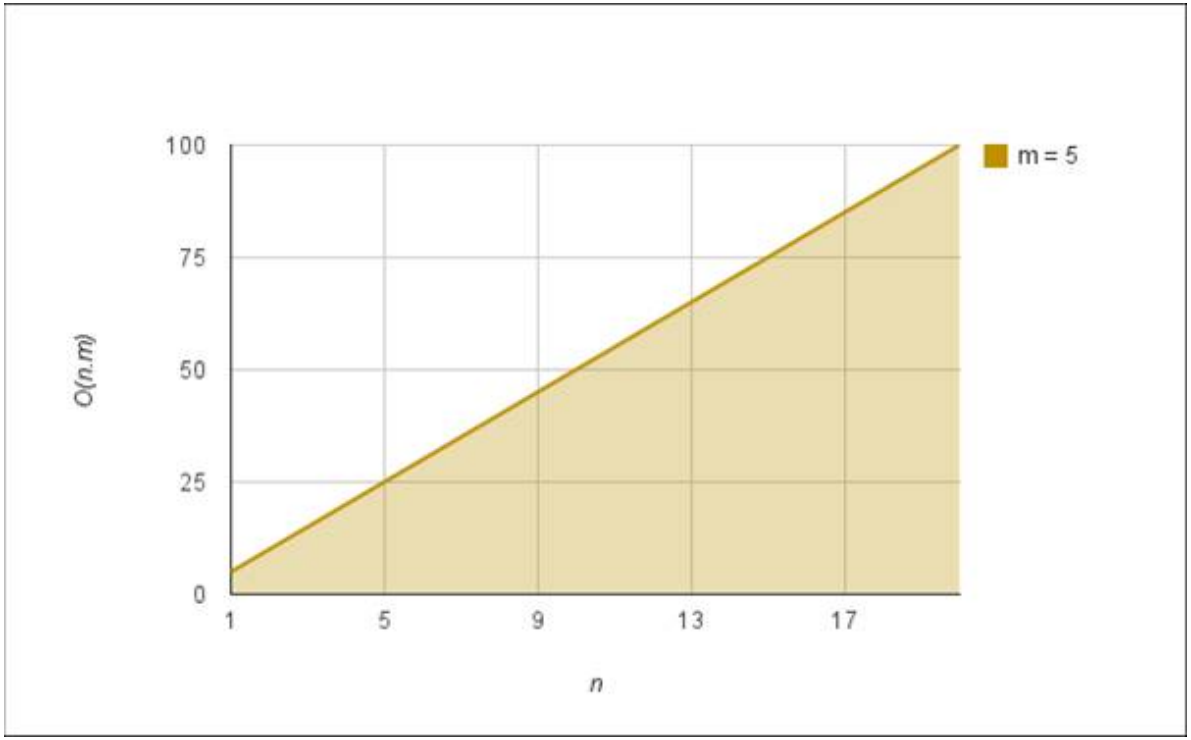
- Caso 1: Si (Pj !=Ti and j=0), el patrón reanuda las comparaciones un lugar mas adelante.
- Caso 2: Si (Pj ==Ti and j != m-1), Tanto el Ti como el Pj pasan al siguiente ndice para volver a comparar.
- Caso 3: Si (Pj !=Ti and j¿0), el patrón cambia de lugar en la posición inicial i=i-PP[j-1] y reanuda las comparaciones en donde falló.



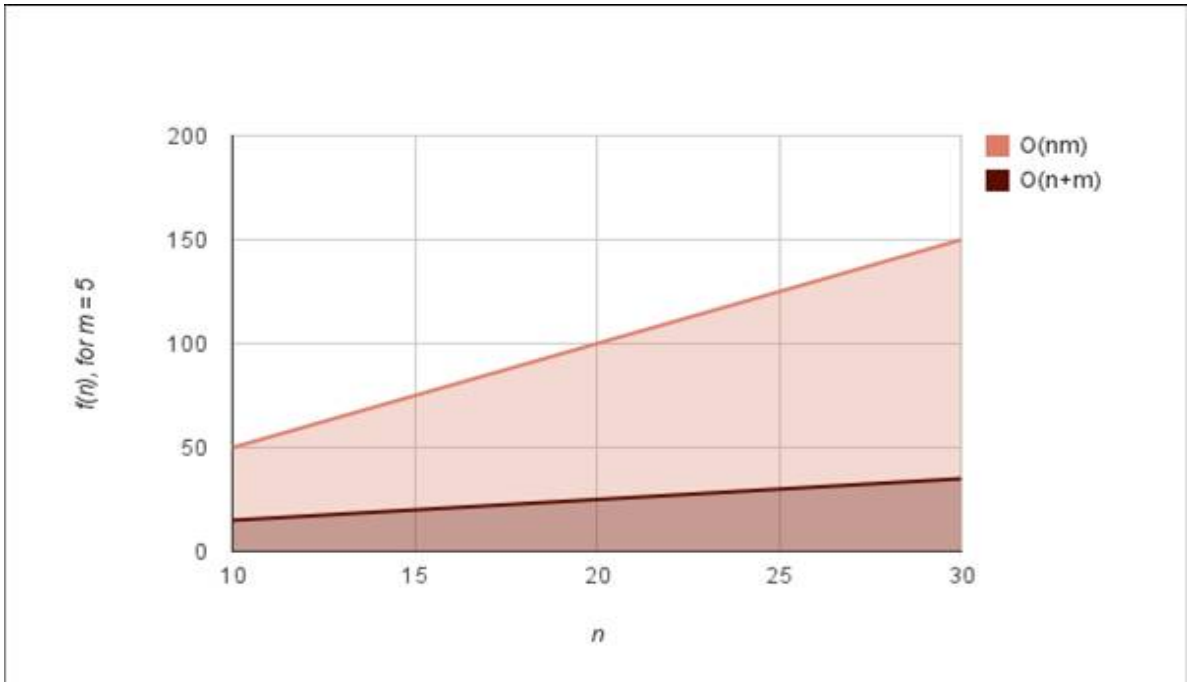
Algoritmo Boyer Moore

- Fue desarrollado por Bob Boyer y Moore en 1977.
- Se considera como el algoritmo de coincidencia de cadena más eficiente en las aplicaciones habituales.
- La comparación en este algoritmo se realiza de derecha a izquierda.
- Demostración en el video.

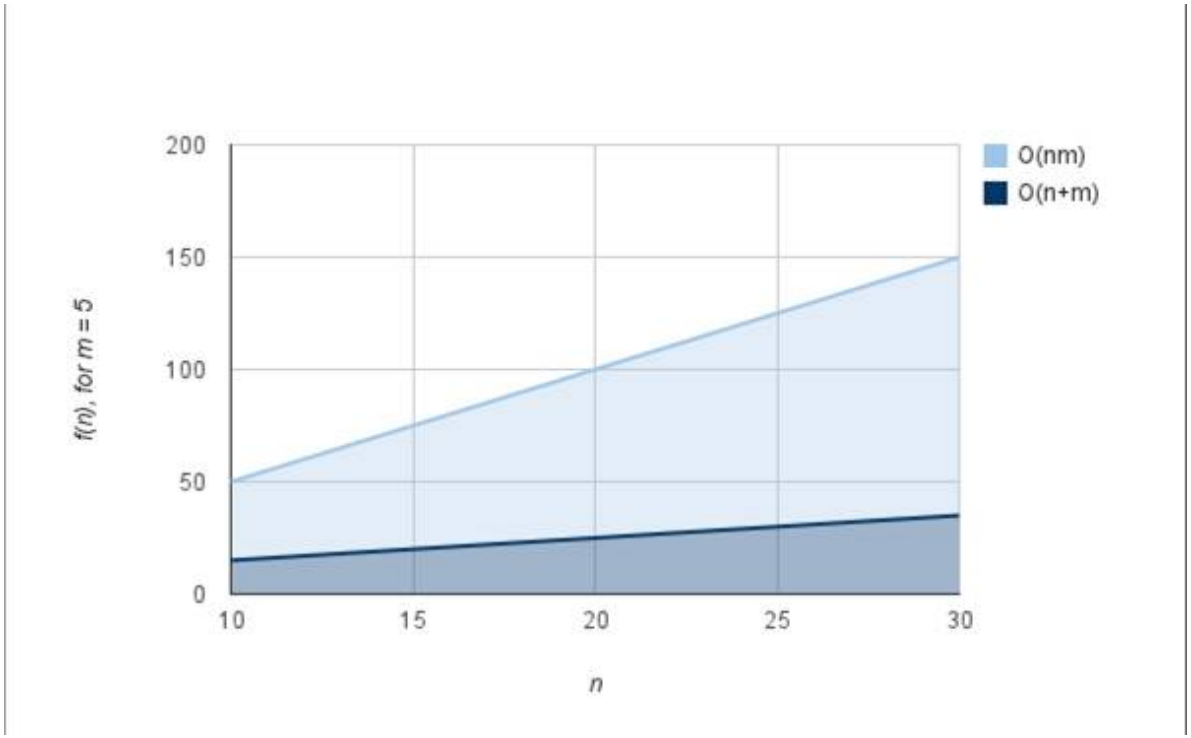
Algoritmo Fuerza Bruta



Algoritmo Knuth Morris Pratt



Algoritmo Boyer Moore



Conclusiones

- Utilizamos la búsqueda en textos desde cosas tan simples como buscar una palabra en un documento cualquiera, hasta resolver problemas relacionados con biología computacional, en donde se requiere buscar patrones dentro de una secuencia de ADN.
- La eficiencia de los algoritmos dependen principalmente del largo del patrón, ya que por ejemplo en Boyer Moore, al ser mas largo un patrón y no encontrar coincidencias se pueden realizar varios saltos en el texto, sin realizar comparaciones.