

Auxiliar 10 - Búsqueda en texto y Grafos

Profesores: Jeremy Barbay
Patricio Poblete

Auxiliares: Daniela Campos, Cristóbal Muñoz
Sven Reisenegger, Bernardo Subercaseaux

P1. Búsqueda en texto

P2. Búsqueda en texto

La búsqueda de patrones en un texto es un problema muy importante en la práctica. Sus aplicaciones en computación son variadas, como por ejemplo la búsqueda de una palabra en un archivo de texto. Existen varios algoritmos para resolver este problema, en este auxiliar practicaremos con dos de ellos.

- (a) Calcule la función de fracaso del algoritmo Knuth-Morris-Pratt (KMP) para el siguiente patrón de búsqueda *abababa*, busque el patrón utilizando KMP en el texto *abbbaabababababab*. Muestre paso a paso cada operación realizada.

Para la función de fracaso sólo se necesita el patrón (*abababa*):

i	0	1	2	3	4	5	6
S	a	b	a	b	a	b	a
$f(i)$	0	0	1	2	3	4	5

Esta se calcula viendo el largo del prefijo propio más largo que también es sufijo propio, de esta manera, cuando encontramos un error a la hora de comparar el patrón con el texto, si es que fallamos antes de un sufijo que también es prefijo, no es necesario que empecemos a comparar todo el patrón nuevamente, sino que partimos del punto en el que fallamos. A continuación explicaremos paso a paso como llenar la tabla:

- Siempre el primer valor (índice 0) es $f(0) = 0$
- Definimos dos indicadores, j que apunta al índice 0 (letra a) e i que apunta al índice 1 (letra b).
- Comparamos las letras de ambos indicadores y como en este caso no son iguales, incrementamos i , es decir ahora i está en el índice 2. El valor de $f(i = 1) = 0$
- Volvemos a comparar la letras de ambos indicadores y ahora son iguales. Como son iguales ahora la función de fracaso en i , será igual al índice de j mas uno, es decir $f(i = 2) = j + 1 = 1$. Incrementamos j e i , ahora $j = 1$ y $i = 3$.
- Comparamos las letras de ambos indicadores y nuevamente son iguales por lo que la función de i será $f(i = 3) = j + 1 = 2$. Este proceso se repite con el resto del patrón.

Utilizaremos esta tabla para decidir como nos movemos por el texto y el patrón. La ejecución se ve en la tabla 1. El texto en el que buscamos nuestro patrón es la línea superior y cada línea de abajo es cada paso que hace el algoritmo para encontrar el patrón. Indicaremos con una X cuando se encuentre que el patrón y el texto no coinciden.

a	b	b	b	a	a	b	b	a	b	a	b	a	b	a	b
a	b	a	...												
		X													
		a	b	...											
		X													
			a	b	...										
			X												
				a	b	...									
					X										
					a	b	a	...							
							X								
							a	b	...						
							X								
								a	b	a	b	a	b	a	

Table 1: Ejecución de KMP

A continuación explicaremos en detalle los pasos que siguió el algoritmo.

- Definiremos dos indicadores, p que se mueve por el patrón y t que se mueve por el texto. Al iniciar tanto p como t están en 0. Vamos comparando carácter a carácter y si coinciden ambos indicadores se incrementan en uno.
- El primer error se encuentra en $p = 2$ y $t = 2$. Para saber como nos seguimos moviendo miramos la tabla. Como el error lo encontramos en $p = 2$, miramos el valor para $p - 1$, es decir 1 (esto lo hacemos porque sabemos que ya encontramos en el texto al patrón hasta $p-1$, en este caso, ya sabemos que ab está en el texto), $f(1) = 0$, por lo que ahora $p = 0$, es decir debemos volver al principio del patrón. El valor de t se mantiene igual.
- Continuamos comparando y ahora hay un error en $p = 0$ y $t = 2$. Como estamos viendo el primer carácter del patrón y encontramos un error, el indicador que cambia es $t = t + 1 = 3$
- Lo mismo ocurre en el siguiente paso.
- Ahora $p = 0$ y $t = 4$. Comparamos y encontramos un error en $p = 1$ y $t = 5$. Miramos la tabla en $p - 1$ y actualizamos a $p = 0$
- Seguimos comparando y ahora encontramos un error en $p = 2$ y $t = 7$. Actualizamos $p = 0$.

- vii. Comparamos y tenemos un error en $p = 0$ y $t = 7$. Como estamos en el primer caracter del patrón incrementamos t , $t = 8$
- viii. Ahora al comparar todos los caracteres, estos calzan por lo que hemos encontrado el patrón.
- (b) Para el patrón y el texto del ejercicio anterior muestre las operaciones realizadas por el algoritmo Boyer-Moore-Horspool.

Al igual que con KMP, sólo se necesita el patrón para calcular la tabla. Esta es la tabla para el patrón.

caracter	desplazamiento
a	6 4 2
b	5 3 1
*	7

Table 2: Tabla de desplazamientos, calculada con los índices de la tabla ??

A continuación explicaremos como obtenemos esta tabla:

- i. La tabla tiene una fila por cada letra diferente que tenga le patrón y una fila mas para cualquier caracter que no esté en el patrón(*)
- ii. El valor que se le asigna a cada fila es $desplazamiento = largoPatron - indice - 1$. Esto no aplica para el último caracter del patron, el cual si es que no ha sido asignado, se le da el valor del largo
- iii. Ahora empezamos a movernos por el patrón, estamos en el índice 0, letra a, le asignamos $desplazamiento = 7 - 0 - 1 = 6$.
- iv. En el índice 1, letra b, le asignamos $desplazamiento = 7 - 1 - 1 = 5$.
- v. En el índice 2, letra a, el desplazamiento ya está asignado, pero lo reemplazamos por $desplazamiento = 7 - 2 - 1 = 4$.
- vi. En el índice 3, letra b, el desplazamiento ya está asignado, pero lo reemplazamos por $desplazamiento = 7 - 3 - 1 = 3$.
- vii. En el índice 4, letra a, el desplazamiento ya está asignado, pero lo reemplazamos por $desplazamiento = 7 - 4 - 1 = 2$.
- viii. En el índice 5, letra b, el desplazamiento ya está asignado, pero lo reemplazamos por $desplazamiento = 7 - 5 - 1 = 1$.
- ix. En el índice 6, letra a, este es el último caracter, pero ya está asignado, por lo que no modificamos la tabla.
- x. Por último, a * le asignamos $desplazamiento = 7$.

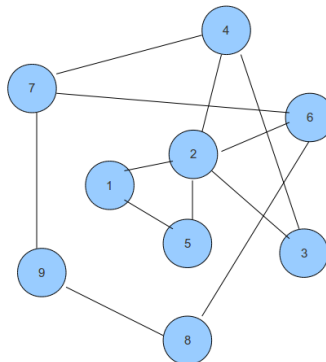
Para buscar en el texto, alineamos el texto con nuestro patrón y empezamos a comparar caracter a caracter, pero de derecha a izquierda. Al encontrar un error miramos el caracter del texto y lo buscamos en la tabla, movemos nuestro patrón lo indicado en la tabla para ese caracter, si dicho caracter no está en la tabla, movemos lo indicado para *.

	a	b	b	b	a	a	b	b	a	b	a	b	a	b	a	b
1	a	b	a	b	a	b	a									
							X									
2		a	b	a	b	a	b	a								
								X								
3			a	b	a	b	a	b	a							
							X									
4				a	b	a	b	a	b	a						
							X									
5						a	b	a	b	a	b	a				
							X									
6									a	b	a	b	a	b	a	

Table 3: Ejecución BMH

P3. Grafos y BFS

Considere el siguiente grafo:



(a) Represente el grafo utilizando Matriz de Adyacencia

	1	2	3	4	5	6	7	8	9
1	0	1	0	0	1	0	0	0	0
2	1	0	1	1	1	1	0	0	0
3	0	1	0	1	0	0	0	0	0
4	0	1	1	0	0	0	1	0	0
5	1	1	0	0	0	0	0	0	0
6	0	1	0	0	0	0	1	1	0
7	0	0	0	1	0	1	0	0	1
8	0	0	0	0	0	1	0	0	1
9	0	0	0	0	0	0	1	1	0

Table 4: Matriz de Adyacencia

(b) Represente el grafo utilizando Lista de Adyacencia

1	2 5
2	1 3 4 5 6
3	2 4
4	2 3 7
5	1 2
6	2 7 8
7	4 6 9
8	6 9
9	7 8

Table 5: Lista de Adyacencia

(c) Determine si el grafo es 2-colorable

Un grafo es 2-coloreable, cuando se pueden pintar todos sus nodos con 2 colores(es decir tener 2 grupos) y que ninguna arista toque 2 nodos del mismo color. Para determinar si este grafo es dos coloreable, tendremos dos colores, rojo y azul. Partiremos pintando el nodo 1 de azul, y ahora pintaremos a todos sus vecinos rojos, es decir ahora 2 y 5 son rojos, pero podemos ver que 2 y 5 son vecinos(los conecta una arista), pero son del mismo color, por lo que este grafo no es 2-coloreable.

(d) Programe un función que reciba un grafo representado por una lista de adyacencia y retorne si el grafo es 2-colorable o no.