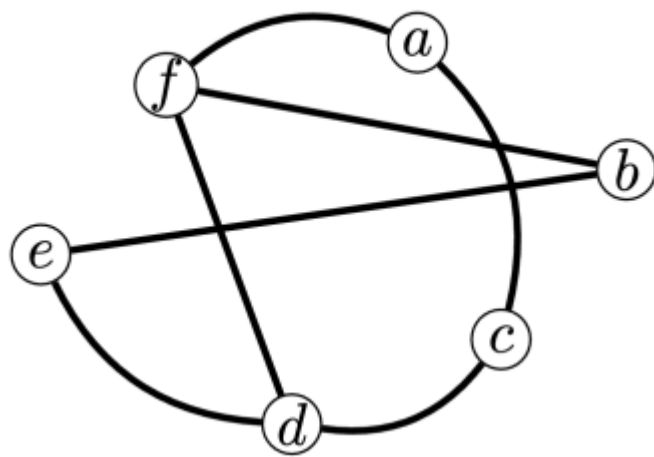


# RECORRIDO DE GRAFOS

Ericka Araya-C20553

## Descripción

Esta tarea consiste en terminar el ejercicio que iniciaron durante la clase del viernes 31 de mayo de 2024. El ejercicio de muestra en el PDF adjunto y consiste en realizar un recorrido en anchura y en profundidad de un grafo.



## Búsqueda en Anchura

### PSEUDOCODE

BFS( $G, s$ )

```

1  for each vertex  $u \in G.V - \{s\}$ 
2       $u.color = \text{WHITE}$ 
3       $u.d = \infty$ 
4       $u.\pi = \text{NIL}$ 
5   $s.color = \text{GRAY}$ 
6   $s.d = 0$ 
7   $s.\pi = \text{NIL}$ 
8   $Q = \emptyset$ 
9  ENQUEUE( $Q, s$ )
10 while  $Q \neq \emptyset$ 
11      $u = \text{DEQUEUE}(Q)$ 
12     for each vertex  $v$  in  $G.Adj[u]$  // search the neighbors of  $u$ 
13         if  $v.color == \text{WHITE}$  // is  $v$  being discovered now?
14              $v.color = \text{GRAY}$ 
15              $v.d = u.d + 1$ 
16              $v.\pi = u$ 
17             ENQUEUE( $Q, v$ ) //  $v$  is now on the frontier
18      $u.color = \text{BLACK}$  //  $u$  is now behind the frontier
    
```

En la iteración 1 se ejecuta de la línea 1 a la 8 del pseudocódigo de la imagen anterior, de ahí en adelante cada iteración corresponde a una corrida del while(línea 10 a la 18).

Aclaración: G es Gris, B es Blanco Y N es Negro, y en la línea del final se muestra como quedaron todos, esta línea no es una iteración como tal. Cada que hay un espacio en blanco significa que para ese vértice no hubo cambios en esa iteración.

Iteración	a			b			c			d			e			f			u	cola
	d	$\pi$	c	d	$\pi$	c	d	$\pi$	c	d	$\pi$	c	d	$\pi$	c	d	$\pi$	c		
1	0	nil	G	$\infty$	nil	B	$\infty$	nil	B	$\infty$	nil	B	$\infty$	nil	B	$\infty$	nil	B		a
2	0	nil	N				1	a	G							1	a	G	a	c f
3							1	a	N	2	c	G							c	f d
4				2	f	G										1	a	N	f	d b
5										2	c	N	3	d	G				d	b e
6				2	f	N													b	e
7													3	d	N					
final	0	nil	N	2	f	N	1	a	N	2	c	N	3	d	N	1	a	N		

## OBJ: Búsqueda en Profundidad

### PSEUDOCODE

DFS( $G$ )

```
1  for each vertex  $u \in G.V$ 
2       $u.color = \text{WHITE}$ 
3       $u.\pi = \text{NIL}$ 
4   $time = 0$ 
5  for each vertex  $u \in G.V$ 
6      if  $u.color == \text{WHITE}$ 
7          DFS-VISIT( $G, u$ )
```

### PSEUDOCODE (continued)

DFS-VISIT( $G, u$ )

```
1   $time = time + 1$                                 // white vertex  $u$  has just been discovered
2   $u.d = time$ 
3   $u.color = \text{GRAY}$ 
4  for each vertex  $v$  in  $G.Adj[u]$  // explore each edge  $(u, v)$ 
5      if  $v.color == \text{WHITE}$ 
6           $v.\pi = u$ 
7          DFS-VISIT( $G, v$ )
8   $time = time + 1$ 
9   $u.f = time$ 
10  $u.color = \text{BLACK}$                                 // blacken  $u$ ; it is finished
```

En la iteración 1 se ejecuta de la línea 1 a la 4 del pseudocódigo(DFS( $G$ )) de la imagen anterior, de ahí en adelante cada iteración corresponde a una corrida del for de la línea 5 de DFS( $G$ ) y del for de la línea 4 de DFS-VISIT( $G, u$ ); cuando no se entra en la condición de “if  $v.color == \text{White}$ ” se omite esta iteración para hacer más simple la tabla, ya que no habría cambios, porque son visitas a vértices que ya están en GRIS O NEGRO, pero debo aclarar que cada vértice sí hace una visita a todos sus adyacentes para verificar que ya estén visitados o recorridos totalmente.

Aclaración: G es Gris, B es Blanco Y N es Negro.

Iteración	tiempo	u	u.d	u.f	u.color	v	v.π
1	0				B		nil
2	1	a	1		G	c	a
3	2	c	2		G	d	c
4	3	d	3		G	e	d
5	4	e	4		G	b	e
6	5	b	5		G	f	b
7	6	f	6		G		
8	7	f	6	7	N		
9	8	b	5	8	N		
10	9	e	4	9	N		
11	10	d	3	10	N		
12	11	c	2	11	N		
13	12	a	1	12	N		