

Práctica 3

Pablo Gutiérrez Aguirre
pgutierrez2018@udec.cl

11 de abril 2022

Tabla de contenidos

- 1 Repaso
 - Recursividad
 - DFS
- 2 Ejercicio 1c
 - Búsqueda en profundidad
 - Búsqueda en anchura
- 3 Ejercicio 2
 - Búsqueda en profundidad
 - Búsqueda en anchura
- 4 Ejercicio 3

Repaso: Recursividad

¿Qué es recursividad?

Como ejemplo, veamos la función factorial.

$$n! = n \cdot (n - 1)!$$

De esta forma

$$5! = 5 \cdot 4! = 5 \cdot 4 \cdot 3! = 5 \cdot 4 \cdot 3 \cdot 2! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot \mathbf{1!} = 5 \cdot 4 \cdot 3 \cdot 2 \cdot \mathbf{1} = 120$$

Repaso: DFS

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.

```

Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

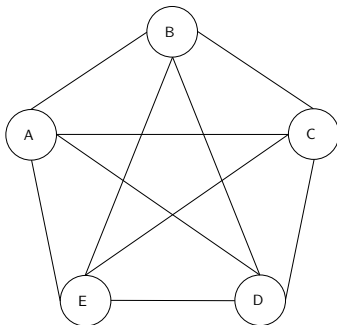
```



Créditos: Mathias Badilla

Ejercicio 1c

Iniciando en el nodo A, explore los siguientes grafos, mediante los algoritmos de **búsqueda en profundidad** y búsqueda en anchura:



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

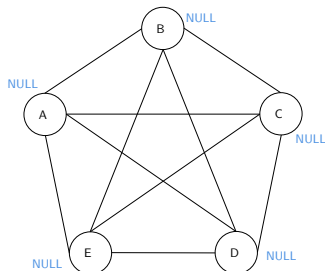
1 for cada vértice  $u \in G.V$  do
2   |  $u.color \leftarrow BLANCO$ 
3   |  $u.\pi \leftarrow NULL$ 
4 end
5 tiempo  $\leftarrow 0$ 
6 for cada vértice  $u \in G.V$  do
7   | if  $u.color = Blanco$  then
8   |   | DFS_VISITAR(G,u)
9   | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

Algorithm: DFS_VISITAR

Input : G: Grafo, u: vertice

```

1 tiempo  $\leftarrow tiempo + 1$ 
2  $u.d \leftarrow tiempo$ 
3  $u.color \leftarrow GRIS$ 
4 for  $v \in G.adyacente(u)$  do
5   | if  $v.color = Blanco$  then
6   |   |  $v.\pi \leftarrow u$ 
7   |   | DFS_VISITAR(G,v)
8   | end
9 end
10  $u.color \leftarrow NEGRO$ 
11 tiempo  $\leftarrow tiempo + 1$ 
12  $u.f \leftarrow tiempo$ 
```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.

```

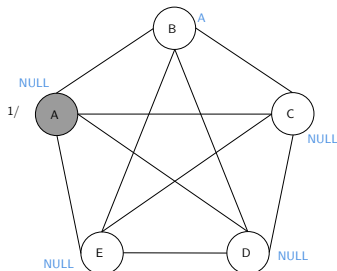
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

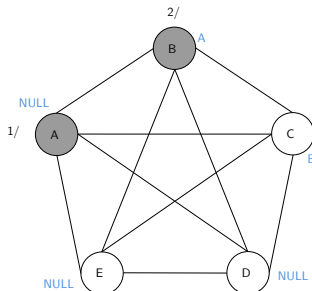
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

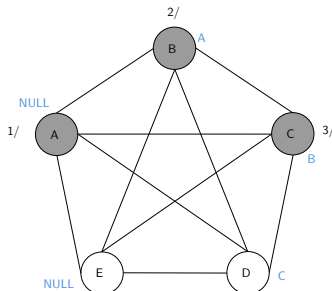
1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 
4 end
5 tiempo  $\leftarrow 0$ 
6 for cada vértice  $u \in G.V$  do
7   if  $u.color = Blanco$  then
8     DFS_VISITAR(G,u)
9   end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

Algorithm: DFS_VISITAR

Input : G: Grafo, u: vertice

```

1 tiempo  $\leftarrow tiempo + 1$ 
2  $u.d \leftarrow tiempo$ 
3  $u.color \leftarrow GRIS$ 
4 for  $v \in G.adyacente(u)$  do
5   if  $v.color = Blanco$  then
6      $v.\pi \leftarrow u$ 
7     DFS_VISITAR(G,v)
8   end
9 end
10  $u.color \leftarrow NEGRO$ 
11 tiempo  $\leftarrow tiempo + 1$ 
12  $u.f \leftarrow tiempo$ 
```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

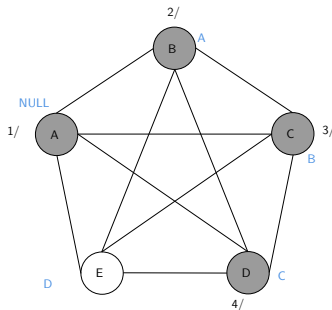
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

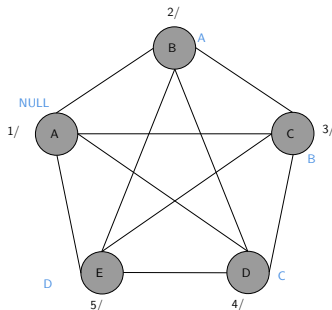
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos finales.
```

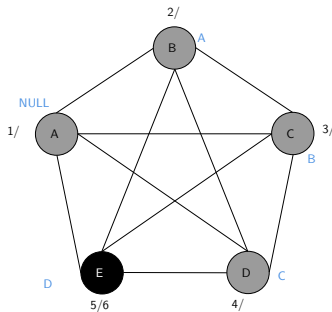
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

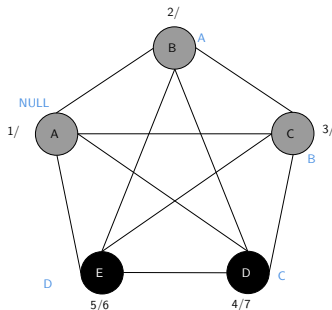
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u. $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6         v. $\pi$   $\leftarrow$  u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

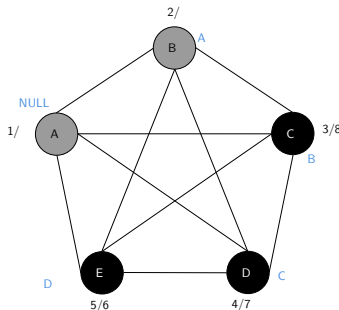
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

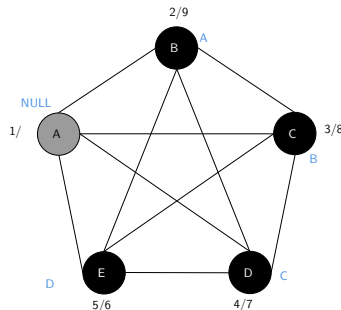
1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 
4 end
5 tiempo  $\leftarrow 0$ 
6 for cada vértice  $u \in G.V$  do
7   if  $u.color = Blanco$  then
8     DFS_VISITAR(G,u)
9   end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

Algorithm: DFS_VISITAR

Input : G: Grafo, u: vertice

```

1 tiempo  $\leftarrow tiempo + 1$ 
2  $u.d \leftarrow tiempo$ 
3  $u.color \leftarrow GRIS$ 
4 for  $v \in G.adyacente(u)$  do
5   if  $v.color = Blanco$  then
6      $v.\pi \leftarrow u$ 
7     DFS_VISITAR(G,v)
8   end
9 end
10  $u.color \leftarrow NEGRO$ 
11 tiempo  $\leftarrow tiempo + 1$ 
12  $u.f \leftarrow tiempo$ 
```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

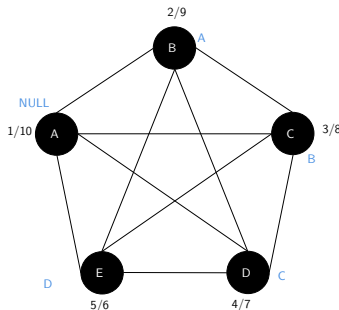
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u. $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6         v. $\pi$   $\leftarrow$  u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



Ejercicio 1c: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2    |  $u.color \leftarrow \text{BLANCO}$ 
3    |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7    | if  $u.color = \text{Blanco}$  then
8    |   | DFS.VISITAR( $G, u$ )
9    |   end
10 end
11 tiempo de descubrimiento / tiempos
   finales.

```

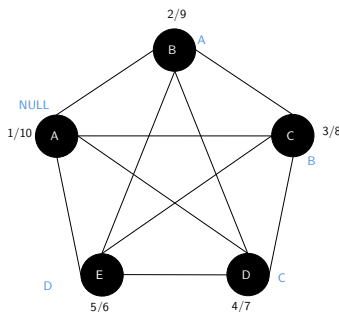
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6         v. $\pi$   $\leftarrow$  u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

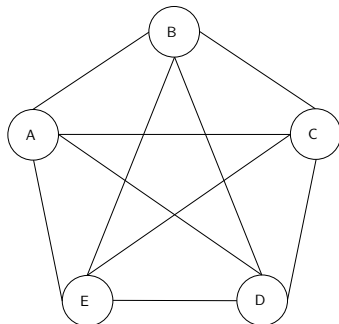
```



- El orden de salida es:
A-B-C-D-E
- Eficiencia $O(|V| + |E|)$

Ejercicio 1c

Iniciando en el nodo A, explore los siguientes grafos, mediante los algoritmos de búsqueda en profundidad y **búsqueda en anchura**:



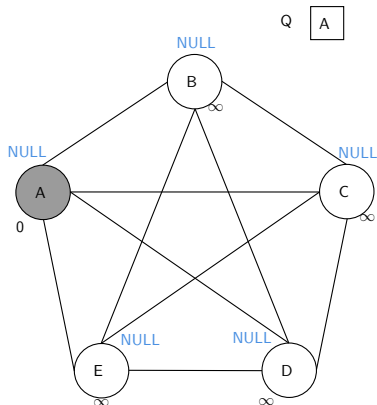
Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vértice
   u ∈ G.V − {s} do
2      u.color ← BLANCO
3      u.d ← ∞
4      u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12     u ← Desencolar(Q)
13     for v ∈ G.adyacentes(u)
14         do
15             if v.color=BLANCO
16                 then
17                     v.color ← GRIS
18                     v.d ← u.d + 1
19                     v.π ← u
20                     Encolar(Q,v)
21             end
22         end
23     end
24     u.color ← NEGRO
25 end
  
```



Salida:

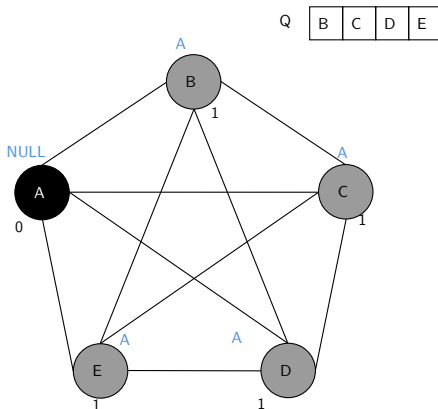
Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vértice
   u ∈ G.V − {s} do
2    u.color ← BLANCO
3    u.d ← ∞
4    u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: A

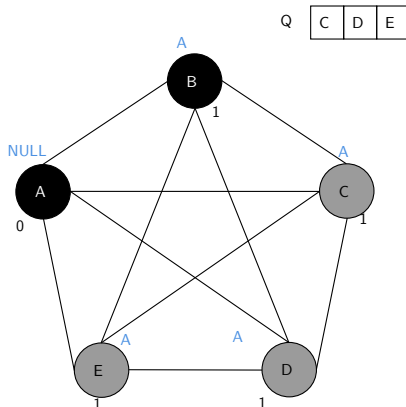
Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vértice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: A,B

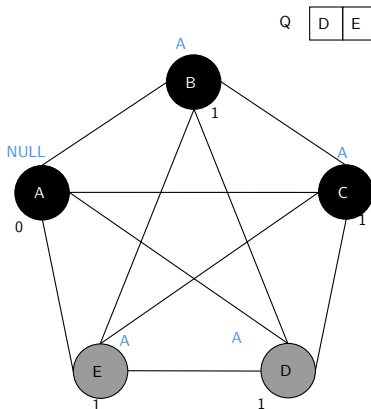
Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vértice
   u ∈ G.V − {s} do
2    u.color ← BLANCO
3    u.d ← ∞
4    u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: A,B,C

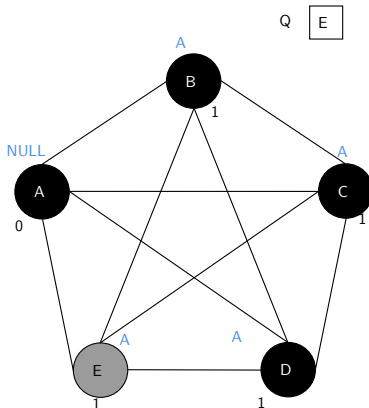
Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



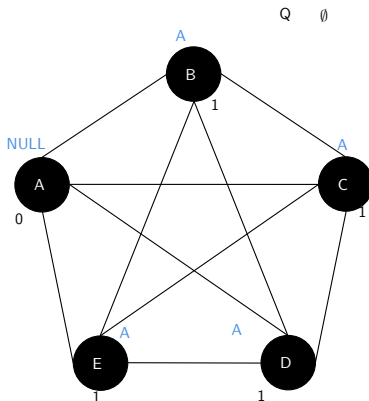
Salida: A,B,C,D

Ejercicio 1c: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

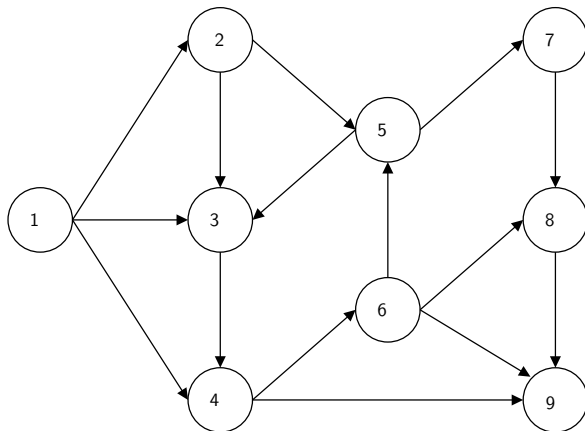
```
1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
```



Salida: A,B,C,D,E

Ejercicio 2

Explore el siguiente grafo a partir del nodo 1, usando **DFS** y **BFS**.



Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

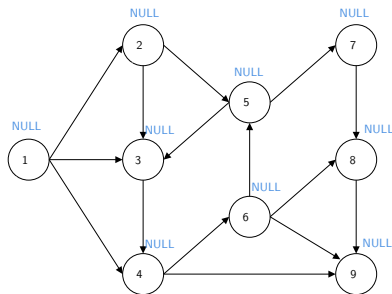
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.$ adyacente( $u$ ) do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR( $G, v$ )
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida:

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

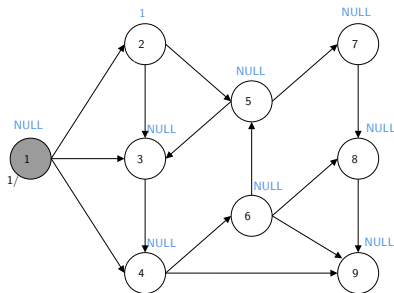
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos finales.
```

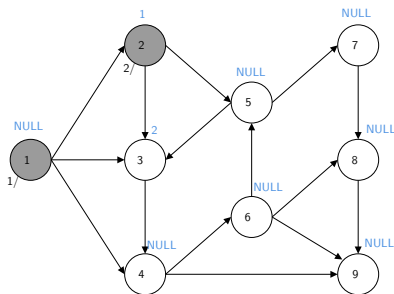
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1,2

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 

```

4 end

5 tiempo $\leftarrow 0$

```

6 for cada vértice  $u \in G.V$  do
7   if  $u.color = \text{Blanco}$  then
8     | DFS_VISITAR( $G, u$ )
9   end

```

10 end

11 tiempo de descubrimiento / tiempos finales.

Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

1 tiempo $\leftarrow tiempo + 1$

2 $u.d \leftarrow tiempo$

```
3  u.color ← GRIS
```

```

4 for  $v \in G.adjacente(u)$  do

```

```

5   if  $v.color = Blanco$  then
6        $v.\pi \leftarrow u$ 
7       DFS_VISITAR( $G, v$ )

```

8 | end

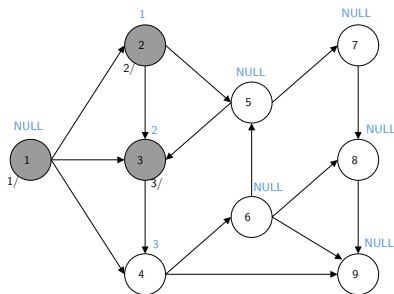
9 end

```
10 u.color ← NEGRO
```

```

11 tiempo  $\leftarrow$  tiempo + 1

```

12 $u.f \leftarrow tiempo$ 

- Salida: 1,2,3

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1 for cada vértice  $u \in G.V$  do
2   |  $u.color \leftarrow \text{BLANCO}$ 
3   |  $u.\pi \leftarrow \text{NULL}$ 
4 end
5 tiempo  $\leftarrow 0$ 
6 for cada vértice  $u \in G.V$  do
7   | if  $u.color = \text{Blanco}$  then
8     |   DFS.VISITAR( $G, u$ )
9     | end
10 end
11 tiempo de descubrimiento / tiempos finales.
```

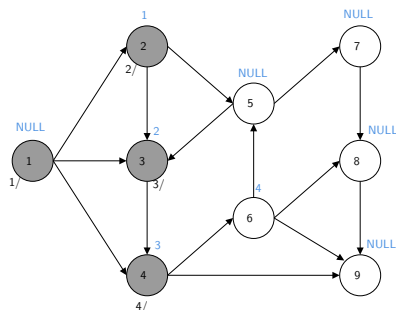
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1,2,3,4

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

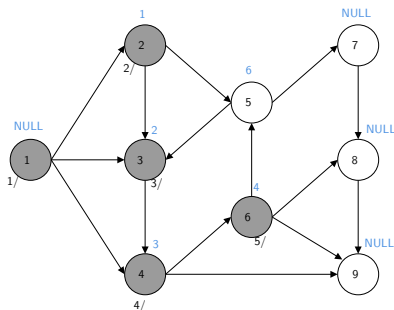
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1,2,3,4,6

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 

```

4 end

5 tiempo $\leftarrow 0$

```

6 for cada vértice  $u \in G.V$  do
7   if  $u.color = \text{Blanco}$  then
8     | DFS_VISITAR( $G, u$ )
9   end

```

10 end

11 tiempo de descubrimiento / tiempos finales.

Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

1 tiempo $\leftarrow tiempo + 1$

2 u.d \leftarrow tiempo

```
3  u.color ← GRIS
```

```

4 for  $v \in G_{advacente(u)}$  do

```

```

5   |   if  $v.color = Blanco$  then

```

$$6 \quad \nu, \pi \leftarrow \mu$$

```

7 DFS_VISITAR(G,v)

```

8 | end

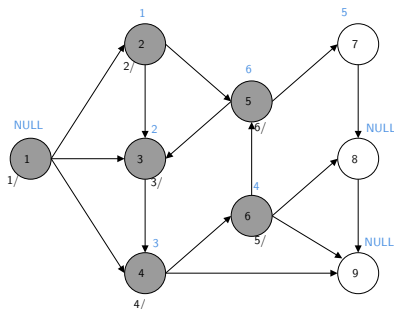
9 end

```
10 u.color ← NEGRO
```

```

11 tiempo  $\leftarrow$  tiempo + 1

```

12 $u.f \leftarrow tiempo$ 

- Salida: 1,2,3,4,6,5

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 

```

4 end

5 tiempo $\leftarrow 0$

```

6 for cada v3rtice  $u \in G.V$  do
7   if  $u.color = Blanco$  then
8     | DFS_VISITAR( $G, u$ )
9   end

```

10 end

11 tiempo de descubrimiento / tiempos finales.

Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

1 tiempo $\leftarrow tiempo + 1$

2 u.d \leftarrow tiempo

```
3  u.color ← GRIS
```

```

4 for  $v \in G_{advacente(u)}$  do

```

```

5   |   if  $v.color = Blanco$  then

```

$$6 \quad \nu, \pi \leftarrow \mu$$

```

7      DFS_VISITAR(G,v)

```

8 | end

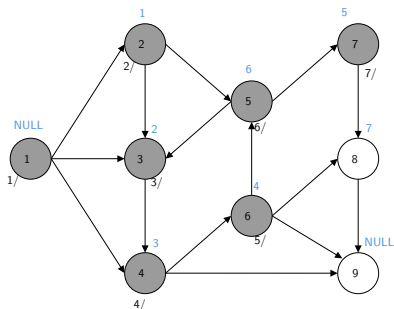
9 end

```
10 u.color ← NEGRO
```

```

11 tiempo  $\leftarrow$  tiempo + 1

```

12 $u.f \leftarrow tiempo$ 

- Salida: 1,2,3,4,6,5,7

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | DFS_VISITAR( $G, u$ )
9      | end
10 end
11 tiempo de descubrimiento / tiempos
    finales.
```

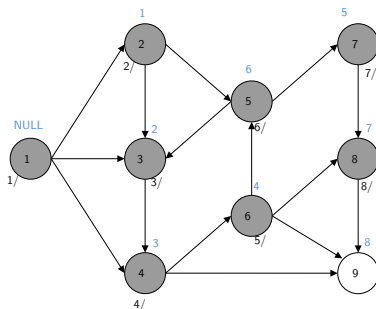
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1,2,3,4,6,5,7,8

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

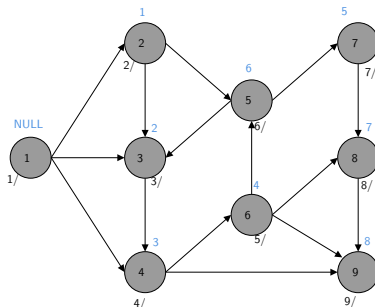
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo ← tiempo + 1
2 u.d ← tiempo
3 u.color ← GRIS
4 for v ∈ G.adyacente(u) do
5     if v.color = Blanco then
6         v.π ← u
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color ← NEGRO
11 tiempo ← tiempo + 1
12 u.f ← tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

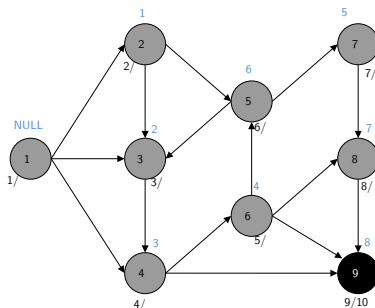
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1 for cada vértice  $u \in G.V$  do
2    $u.color \leftarrow BLANCO$ 
3    $u.\pi \leftarrow NULL$ 

```

4 end

5 tiempo $\leftarrow 0$

```

6 for cada v3rtice  $u \in G.V$  do
7   if  $u.color = Blanco$  then
8     | DFS_VISITAR( $G, u$ )
9   end

```

10 end

11 tiempo de descubrimiento / tiempos finales.

Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

1 tiempo $\leftarrow tiempo + 1$

2 u.d \leftarrow tiempo

```

3  u.color ← GRIS

```

4 **for** $v \in G_{advacente(u)}$ **do**

```

5 | if  $v.color = Blanco$  then

```

$$6 \quad \nu, \pi \leftarrow \mu$$

```

7      DFS_VISITAR(G_v)

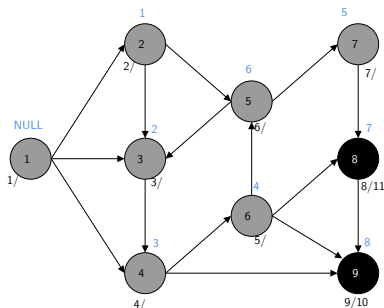
```

end

9 end

```
10 u.color ← NEGRO
```

```
11 tiempo  $\leftarrow$  tiempo + 1
```

12 $\mu.f \leftarrow tiempo$ 

- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

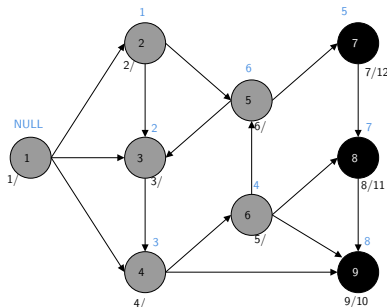
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G : Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos finales.
```

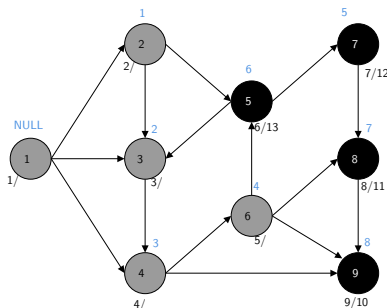
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.

```

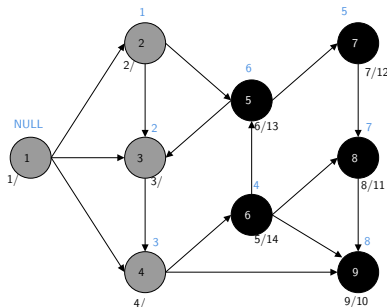
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

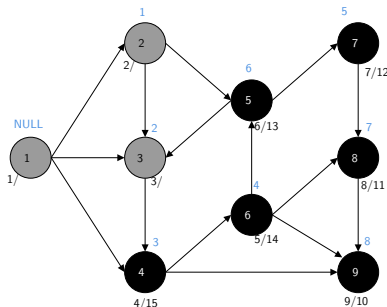
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.$ adyacente( $u$ ) do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR( $G, v$ )
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8      |     | DFS_VISITAR( $G, u$ )
9      |     end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

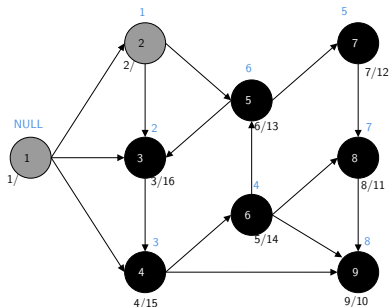
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.$ adyacente( $u$ ) do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR( $G, v$ )
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G: Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow \text{BLANCO}$ 
3      |  $u.\pi \leftarrow \text{NULL}$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = \text{Blanco}$  then
8          | | DFS_VISITAR( $G, u$ )
9          | end
10 end
11 tiempo de descubrimiento / tiempos
   finales.
```

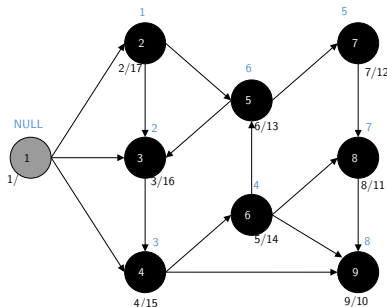
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

```



- Salida: 1,2,3,4,6,5,7,8,9

Ejercicio 2: Búsqueda en profundidad

Algorithm: DFS

Input : G : Grafo

```

1  for cada vértice  $u \in G.V$  do
2      |  $u.color \leftarrow BLANCO$ 
3      |  $u.\pi \leftarrow NULL$ 
4  end
5  tiempo  $\leftarrow 0$ 
6  for cada vértice  $u \in G.V$  do
7      | if  $u.color = Blanco$  then
8          | DFS_VISITAR( $G, u$ )
9      end
10 end
11 tiempo de descubrimiento / tiempos
    finales.

```

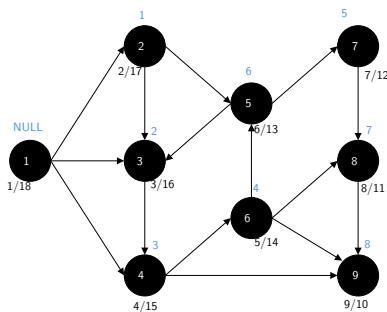
Algorithm: DFS_VISITAR

Input : G: Grafo, u:vertice

```

1 tiempo  $\leftarrow$  tiempo + 1
2 u.d  $\leftarrow$  tiempo
3 u.color  $\leftarrow$  GRIS
4 for  $v \in G.adyacente(u)$  do
5     if v.color = Blanco then
6          $v.\pi \leftarrow u$ 
7         DFS_VISITAR(G,v)
8     end
9 end
10 u.color  $\leftarrow$  NEGRO
11 tiempo  $\leftarrow$  tiempo + 1
12 u.f  $\leftarrow$  tiempo

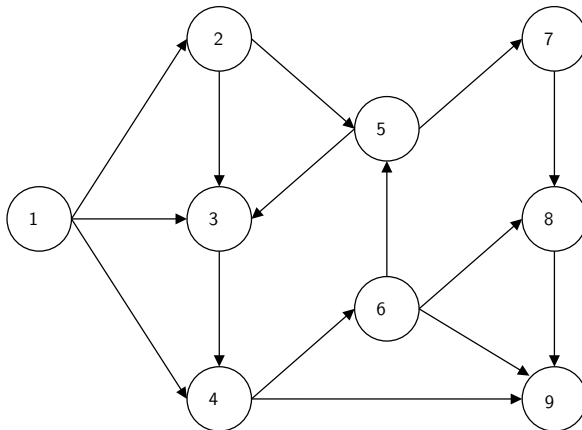
```



- Salida: 1,2,3,4,6,5,7,8,9
- Eficiencia $O(|V| + |E|)$

Ejercicio 2

Explore el siguiente grafo a partir del nodo 1, usando DFS y **BFS**.



Ejercicio 2: Búsqueda en anchura

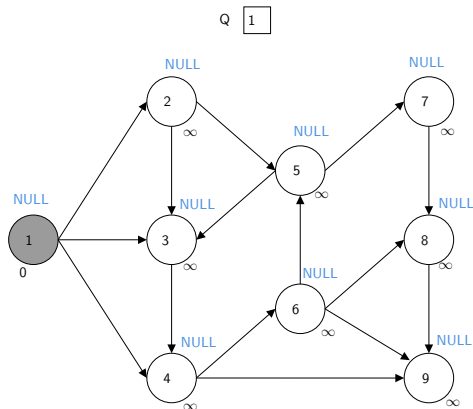
Algorithm: BFS

Input : G: Grafo,s:vertice inicial

```

1  for cada vértice
2      u ∈ G.V − {s} do
3          u.color ← BLANCO
4          u.d ← ∞
5          u.π ← NULL
6  end
7  s.color ← GRIS
8  s.d ← 0
9  s.π ← NULL
10 Q ← ∅
11 while Q ≠ ∅ do
12     u ← Desencolar(Q)
13     for v ∈ G.adyacentes(u)
14         do
15             if v.color = BLANCO
16                 then
17                     v.color ← GRIS
18                     v.d ← u.d + 1
19                     v.π ← u
20                     Encolar(Q, v)
21                 end
22             end
23         end
24     end
25     u.color ← NEGRO
26 end

```



Salida:

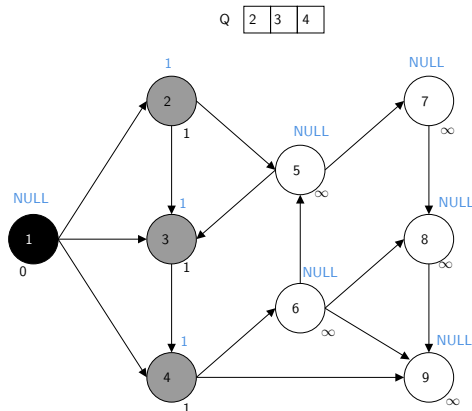
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vertice
   u ∈ G.V − {s} do
2    u.color ← BLANCO
3    u.d ← ∞
4    u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23     end
24   end
25   u.color ← NEGRO
26 end
  
```



Salida: 1

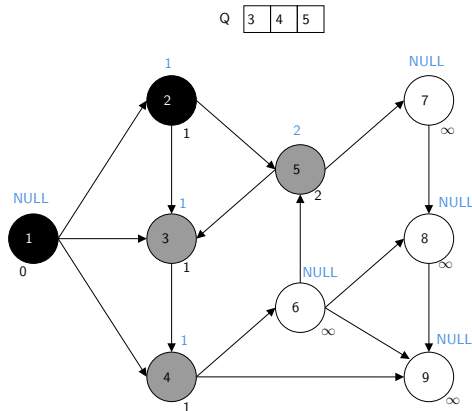
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2

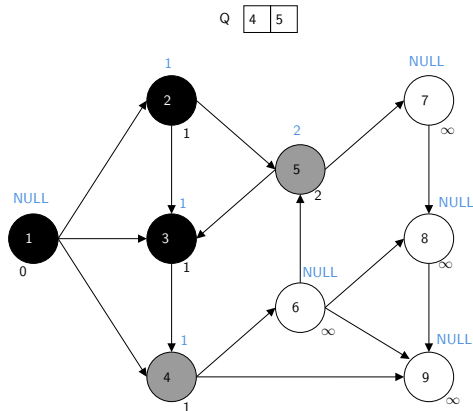
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2,3

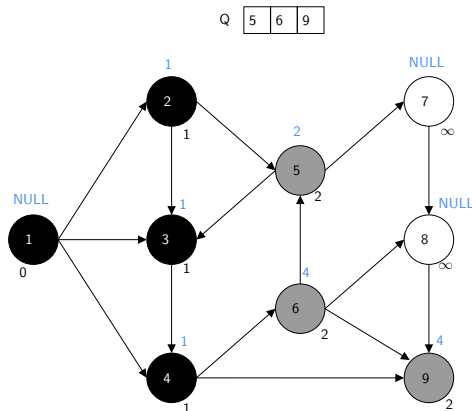
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo,s:vertice
inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           V.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2,3,4

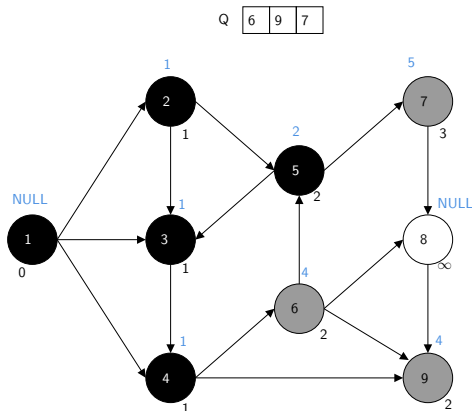
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2,3,4,5

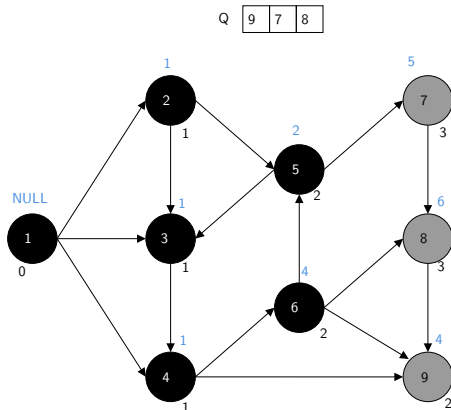
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2,3,4,5,6

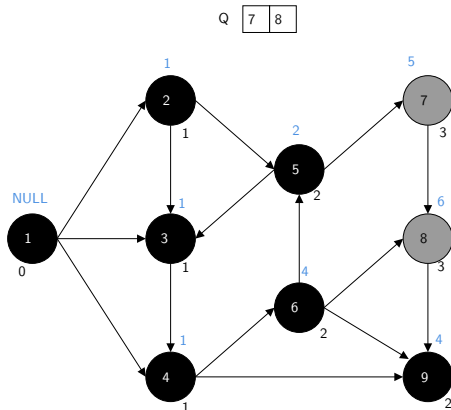
Ejercicio 2: Búsqueda en anchura

Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1 for cada vertice
  u ∈ G.V − {s} do
2   u.color ← BLANCO
3   u.d ← ∞
4   u.π ← NULL
5 end
6 s.color ← GRIS
7 s.d ← 0
8 s.π ← NULL
9 Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end
  
```



Salida: 1,2,3,4,5,6,9

Ejercicio 2: Búsqueda en anchura

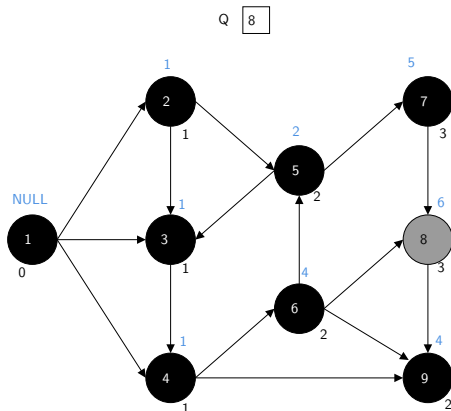
Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vertice
   u ∈ G.V − {s} do
2    u.color ← BLANCO
3    u.d ← ∞
4    u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12   u ← Desencolar(Q)
13   for v ∈ G.adyacentes(u)
14     do
15       if v.color=BLANCO
16         then
17           v.color ← GRIS
18           v.d ← u.d + 1
19           v.π ← u
20           Encolar(Q,v)
21         end
22       end
23   end
24   u.color ← NEGRO
25 end

```



Salida: 1,2,3,4,5,6,9,7

Ejercicio 2: Búsqueda en anchura

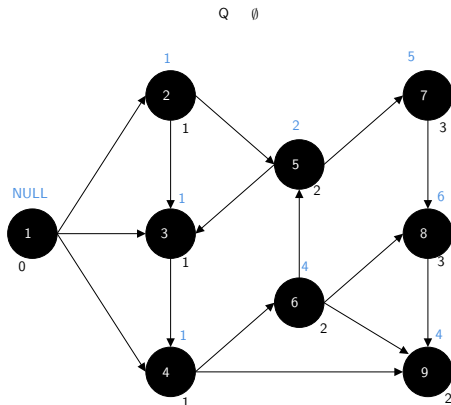
Algorithm: BFS

Input : G: Grafo, s: vertice inicial

```

1  for cada vertice
   u ∈ G.V − {s} do
2      u.color ← BLANCO
3      u.d ← ∞
4      u.π ← NULL
5  end
6  s.color ← GRIS
7  s.d ← 0
8  s.π ← NULL
9  Q ← ∅
10 Encolar(Q,s)
11 while Q ≠ ∅ do
12     u ← Desencolar(Q)
13     for v ∈ G.adyacentes(u)
14         do
15             if v.color=BLANCO
16                 then
17                     v.color ← GRIS
18                     V.d ← u.d + 1
19                     v.π ← u
20                     Encolar(Q,v)
21             end
22         end
23     end
24     u.color ← NEGRO
25 end

```



Salida: 1,2,3,4,5,6,9,7,8

Ejercicio 3

Dado un grafo $G = (V, E)$, escriba un script en Python que determine si G es un árbol o no.

- **Entrada:** La primera línea de la entrada corresponde al número de casos de prueba N , con $1 \leq N \leq 50$. Las líneas siguientes corresponden a los casos de prueba. Cada caso de prueba comienza con una línea que tiene dos enteros $n = |V|$ y $m = |E|$, separados por un espacio, con $2 \leq n \leq 20000$ y $0 \leq m \leq 50000$. Las siguientes m líneas contienen dos enteros separados por un espacio, entre 0 y $n - 1$, que representan a las aristas del grafo.
- **Salida:** Para cada caso de prueba, imprimir la línea Caso x:, seguido de la frase ES UN ÁRBOL, si G es un árbol, o NO ES UN ÁRBOL, en caso contrario.

Ejercicio 3

- **Ejemplo:**

Entrada	Salida
2	Caso #1: NO ES UN ARBOL Caso #2: ES UN ARBOL
4 3	
0 1	
0 2	
1 2	
5 4	
0 1	
0 2	
2 3	
3 4	

Gracias por su atención

Dudas o consultas:

pgutierrez2018@udec.cl