



**PUCMM**  
Pontificia Universidad Católica  
Madre y Maestra

Por: Junior Hernández

## L5 Recorriendo grafos

- **Planteamiento:**

Considere el siguiente grafo directo, donde cada nodo representa un estado y cada arista dirigida representa una operación.

El grafo lo representamos así:

a → c, b, g, e

c → p, h, d

b → d

g → o

e → g, n, m

d → h, f

g → o

n → o, k

Suponga que se va a buscar en el espacio de estados, comenzando en el nodo "a". Enumere los nodos del espacio de estados en el orden en que se expanden para cada uno de los métodos de búsqueda especificados más abajo. Suponga que la búsqueda considera los nodos secundarios de izquierda a derecha. Indique cualquier otra suposición que haga.

1.- Búsqueda en profundidad (DPS):

2.- Búsqueda en Amplitud (BFS)

3.- Profundización iterativa

Repita para el siguiente grafo

a → b, c

b → c, d

c → e, f

d → e, i, h

e → f

f → i

h → j, i

i → j

Para cada grafo usamos vectores de adyacencia. Cada línea nos da el nodo seguido de una lista de nodos adyacentes.

- **Solución:**

Código para el primer grafo

```
grapho(a, b).
grapho(a, c).
grapho(a, b).
grapho(a, g).
grapho(a, e).
grapho(c, p).
grapho(c, h).
grapho(c, d).
grapho(g, o).
grapho(e, g).
grapho(e, n).
grapho(e, m).
grapho(d, h).
grapho(d, f).
grapho(n, o).
grapho(n, k).
```

```
cola(k).
```

#### **%Busqueda en Profundidad (DPS)**

```
busquedaDPS(Inicio, BusquedaDPS, Final) :-
busqueda(Inicio, BusquedaDPS, Final).
busqueda(N, [N], _) :- cola(N).
busqueda(N, [N|S], M) :- M > 0,
grapho(N, N1),
M1 is M-1,
busqueda(N1, S, M1).
```

#### **%Busqueda en Amplitud (BFS)**

```
busquedaBFS(H, T) :- busquedas([[H]], T).
busquedas([[J|U]|_], [J|U]) :- cola(J).
busquedas([N|I], O) :- extender(N, R),
append(I, R, E),
busquedas(E, O).
extender([J|U], R) :- bagof([A, J|U],
(grapho(J, A),
\+ (member(A, [J|U]))),
R), !.
extender(_ , []).
```

#### **%Profundización iterativa**

```
iterativa(Nodo, Path) :-
buscar(Nodo, Meta, Path),
cola(Meta).
buscar(J, J, []).
buscar(U, N, [N|I]) :- buscar(U, O, I),
grapho(O, N),
\+ member(N, I).
```

Salidas para el primer grafo

### Búsqueda en Profundidad (DPS)

 <code>busquedaDPS(n,R,10).</code>	  
<b>R</b> = [n, k]	
 <code>busquedaDPS(a,R,10).</code>	  
<b>R</b> = [a, e, n, k]	
 <code>busquedaDPS(k,R,10).</code>	  
<b>R</b> = [k]	
 <code>busquedaDPS(e,R,10).</code>	  
<b>R</b> = [e, n, k]	

### Búsqueda en Amplitud (BFS)

 <code>busquedaBFS(e,R).</code>	  
<b>R</b> = [k, n, e]	
 <code>busquedaBFS(a,R).</code>	  
<b>R</b> = [k, n, e, a]	
 <code>busquedaBFS(n,R).</code>	  
<b>R</b> = [k, n]	
 <code>busquedaBFS(k,R).</code>	  
<b>R</b> = [k]	

### Profundización iterativa

 <code>iterativa(k,R).</code>	  
<b>R</b> = [k]	
 <code>iterativa(a,R).</code>	  
<b>R</b> = [k, n, e, a]	
 <code>iterativa(e,R).</code>	  
<b>R</b> = [k, n, e]	
 <code>iterativa(n,R).</code>	  
<b>R</b> = [k, n]	

Código para el segundo grafo

```
grapho(a, b).
grapho(a, c).
grapho(b, c).
grapho(b, d).
grapho(c, e).
grapho(c, f).
grapho(d, e).
grapho(d, i).
grapho(d, h).
grapho(e, f).
grapho(f, i).
grapho(h, j).
grapho(h, i).
grapho(i, j).
```

```
cola(j).
```

#### **%Busqueda en Profundidad (DPS)**

```
busquedaDPS(Inicio, BusquedaDPS, Final) :-
    busqueda(Inicio, BusquedaDPS, Final).
busqueda(N, [N], _) :- cola(N).
busqueda(N, [N|S], M) :- M > 0,
    grapho(N, N1),
    M1 is M-1,
    busqueda(N1, S, M1).
```

#### **%Busqueda en Amplitud (BFS)**

```
busquedaBFS(H, T) :- busquedas([[H]], T).
busquedas([J|U], [J|U]) :- cola(J).
busquedas([N|I], O) :- extender(N, R),
    append(I, R, E),
    busquedas(E, O).
extender([J|U], R) :- bagof([A, J|U],
    (grapho(J, A),
    \+ (member(A, [J|U]))),
    R), !.
extender(_, []).
```

#### **%Profundización iterativa**

```
iterativa(Nodo, Path) :-
    buscar(Nodo, Meta, Path),
    cola(Meta).
buscar(J, J, [J]).
buscar(U, N, [N|I]) :- buscar(U, O, I),
    grapho(O, N),
    \+ member(N, I).
```

Salidas para el segundo grafo

### Búsqueda en Profundidad (DPS)

 <code>busquedaDPS(e,R,10).</code>	  
<code>R = [e, f, i, j]</code>	
 <code>busquedaDPS(f,R,10).</code>	  
<code>R = [f, i, j]</code>	
 <code>busquedaDPS(a,R,10).</code>	  
<code>R = [a, b, c, e, f, i, j]</code>	
 <code>busquedaDPS(i,R,10).</code>	  
<code>R = [i, j]</code>	

### Búsqueda en Amplitud (BFS)

 <code>busquedaBFS(i,R).</code>	  
<code>R = [j, i]</code>	
 <code>busquedaBFS(a,R).</code>	  
<code>R = [j, i, d, b, a]</code>	
 <code>busquedaBFS(c,R).</code>	  
<code>R = [j, i, f, c]</code>	
 <code>busquedaBFS(e,R).</code>	  
<code>R = [j, i, f, e]</code>	

### Profundización iterativa

 <code>iterativa(e,R).</code>	  
<code>R = [j, i, f, e]</code>	
 <code>iterativa(a,R).</code>	  
<code>R = [j, i, d, b, a]</code>	
 <code>iterativa(b,R).</code>	  
<code>R = [j, i, d, b]</code>	
 <code>iterativa(c,R).</code>	  
<code>R = [j, i, f, c]</code>	