

Recorridos en Grafos

Estructuras de Datos

Gabriel Ávila

Pontificia Universidad Javeriana

Departamento de Ingeniería de Sistemas

Recorridos en Grafos

Recorridos en Grafos

- Visitar todos los vértices y/o todas las aristas de un grafo.
- El orden de visita puede llegar a ser importante (depende del algoritmo).
- Durante el recorrido, es necesario llevar cuenta de los nodos ya visitados (marcarlos).

Recorridos en Grafos

¿Qué tipos de problemas se pueden resolver con recorridos en grafos?

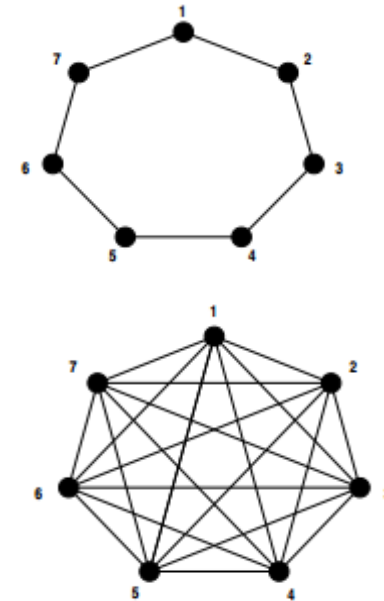
Recorridos en Grafos

¿Qué tipos de problemas se pueden resolver con recorridos en grafos?

- Encontrar todos los vértices alcanzables.
- Encontrar el mejor nodo alcanzable.
- Encontrar la mejor ruta a través del grafo.
- Determinar si un grafo es un Grafo Acíclico Dirigido.
- Ordenar topológicamente un grafo.

Recorridos en Grafos

- .Plano.
- .Preorden.
- .Niveles / Vecindario.
- .Euler:
 - .Todas las aristas una vez.
- .Hamilton:
 - .Todos los vértices una vez.



Recorridos en Grafos

.Plano \rightarrow elementos del conjunto V , ignorando E .

.Preorden.

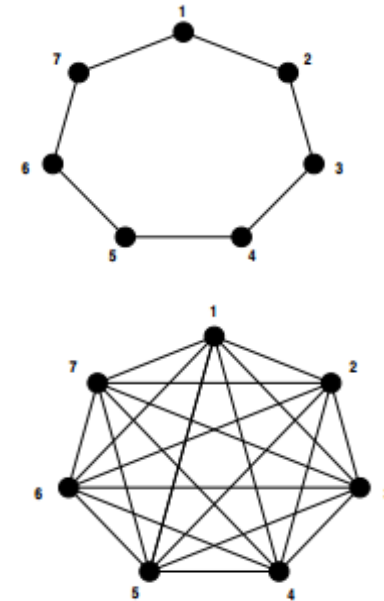
.Niveles / Vecindario.

.Euler:

.Todas las aristas una vez.

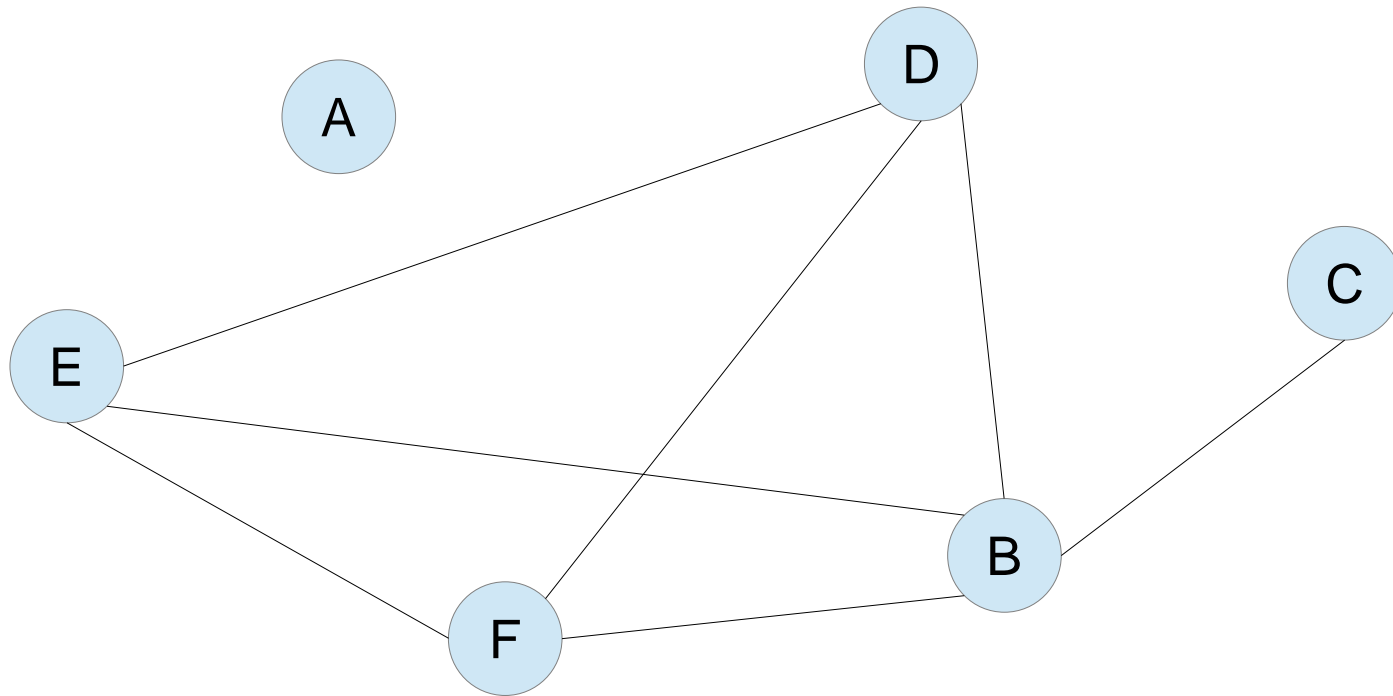
.Hamilton:

.Todos los vértices una vez.



Recorridos en Grafos

.Recorrido plano:



.A – D – C – B – F – E

Recorridos en Grafos

•Plano.

•Preorden → Depth-first search (DFS).

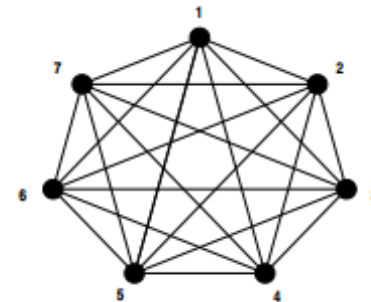
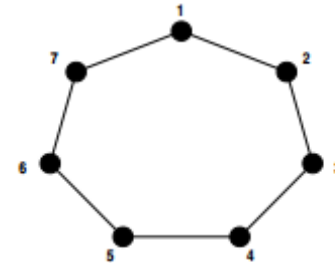
•Niveles / Vecindario → Breadth-first search (BFS).

•Euler:

•Todas las aristas una vez.

•Hamilton:

•Todos los vértices una vez.

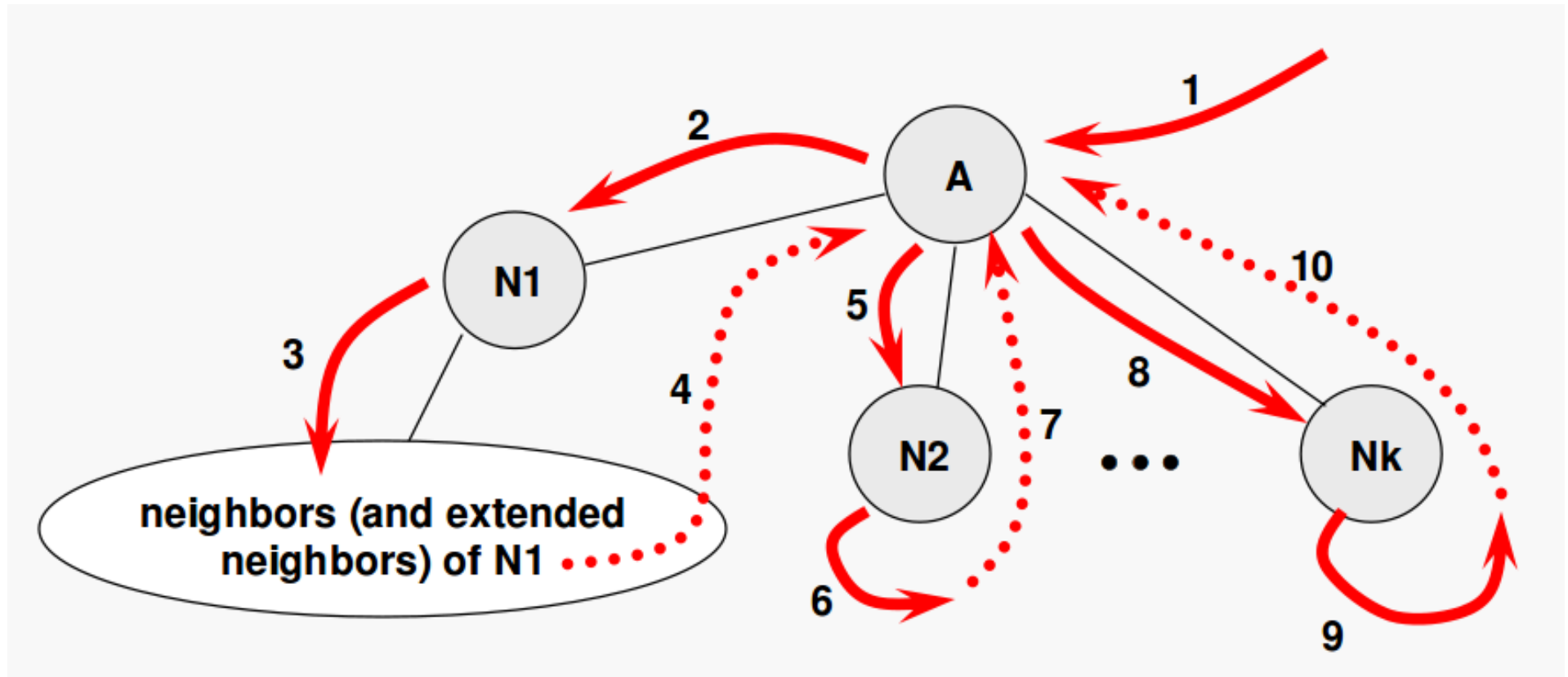


Búsqueda en profundidad (DFS)

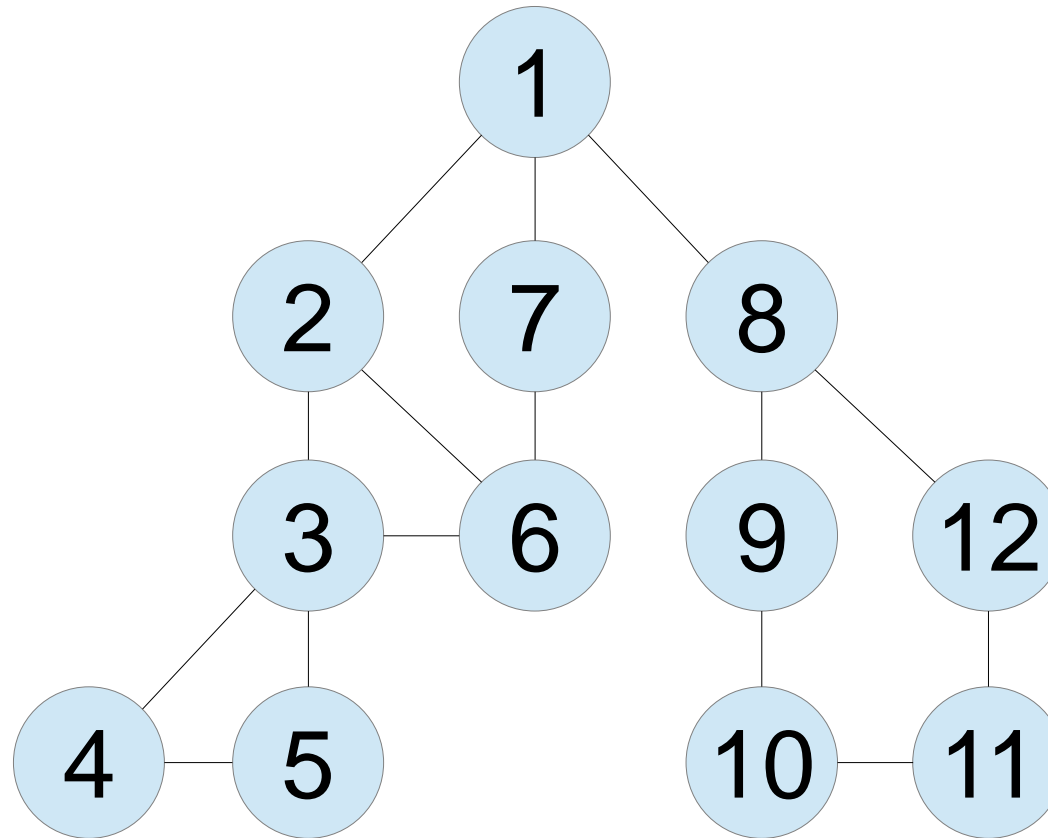
Búsqueda en profundidad: *Depth-first search*.

- Análogo al recorrido en preorden del árbol.
- Para un vértice dado, se visita un vecino y se sigue la ruta hasta que no se pueda avanzar más. En otras palabras, se visita el vértice, y luego de forma recursiva todos sus vecinos en orden.
- Se recorre en profundidad un camino antes de pasar al siguiente (backtracking).
- Complejidad: $O(|E|)$.

Búsqueda en profundidad (DFS)



Búsqueda en profundidad (DFS)



orden en que los nodos son visitados

Búsqueda en profundidad (DFS)

Implementación:

1. Procedimiento recursivo.
2. Utilización de una pila.

Búsqueda en profundidad (DFS)

1. Implementación recursiva:

procedimiento DFS (G, v)

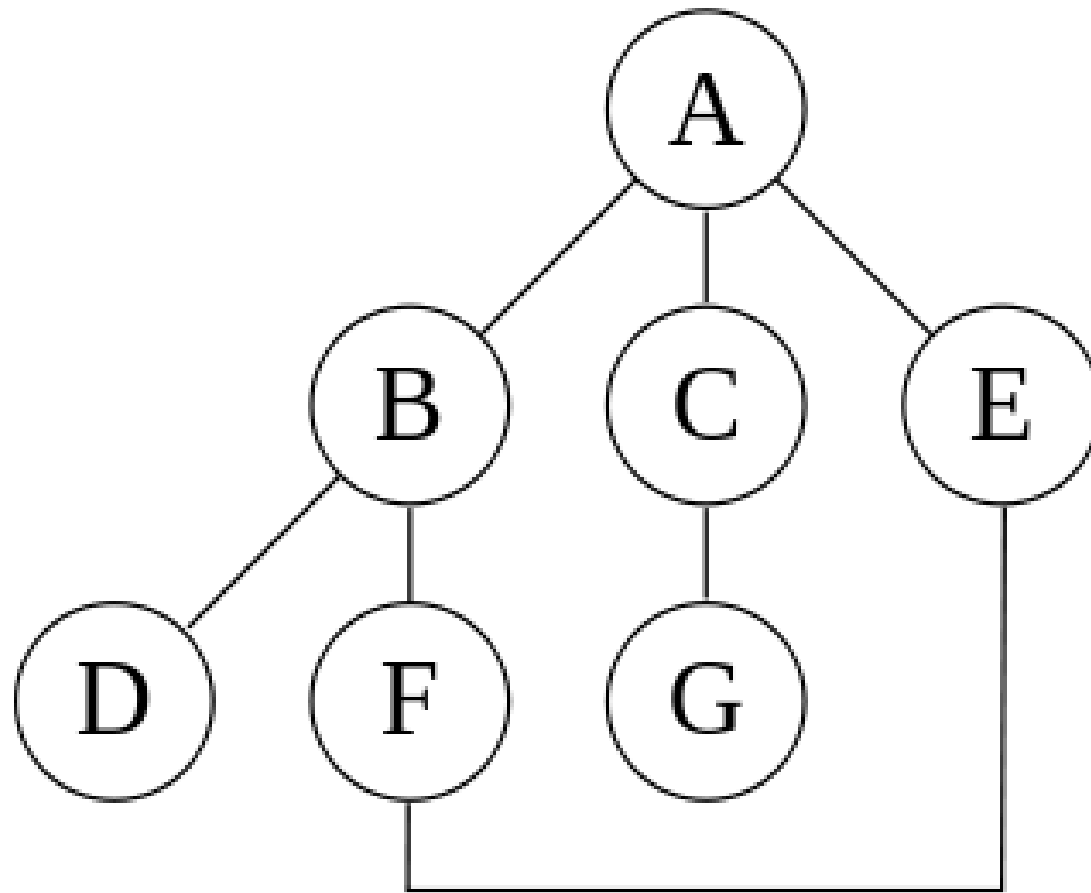
 marcar v como visitado

para todas las aristas (v, w) donde v es origen

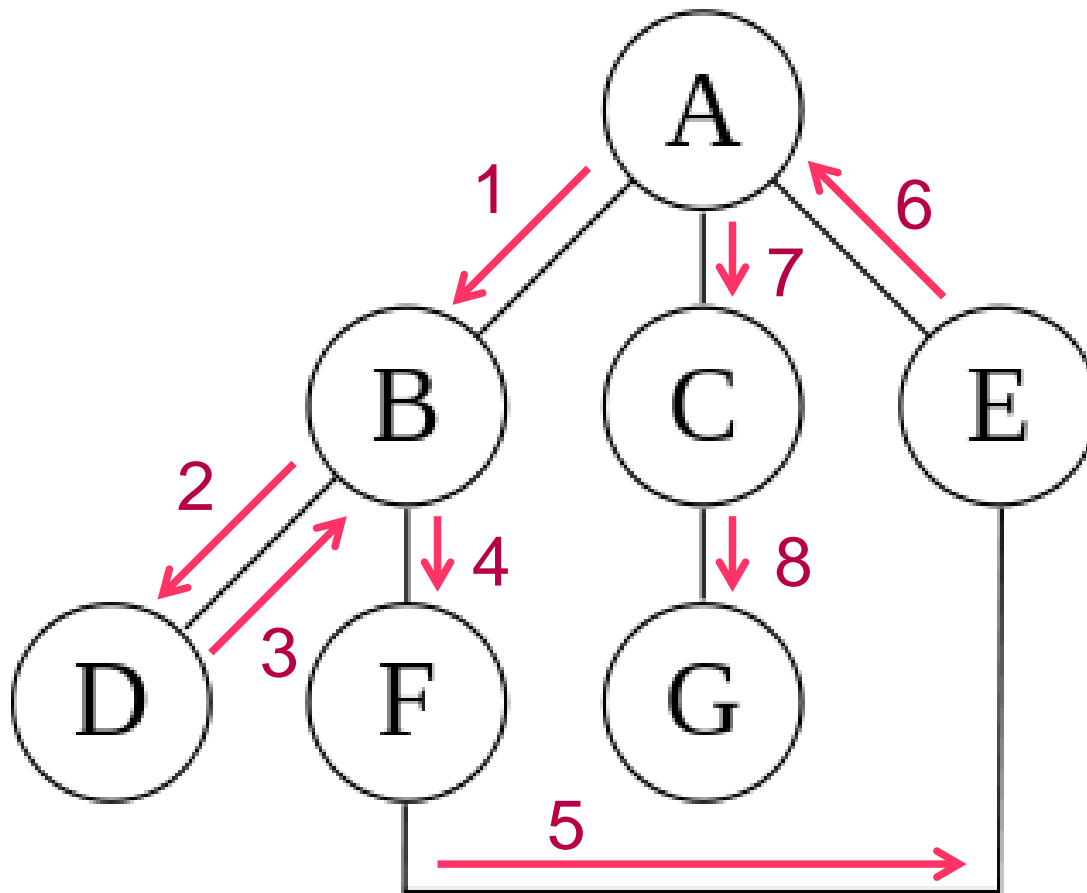
si el vértice w no está marcado como visitado

 llamar recursivamente a DFS (G, w)

Búsqueda en profundidad (DFS)



Búsqueda en profundidad (DFS)



DFS (Grafo, A)
↳ DFS (Grafo, B)
↳ DFS (Grafo, D)
↳ DFS (Grafo, F)
↳ DFS (Grafo, E)
↳ DFS (Grafo, C)
↳ DFS (Grafo, G)

A – B – D – F – E – C – G

Búsqueda en profundidad (DFS)

2. Implementación con pila:

procedimiento DFS-iterativo (G, v)

sea S una pila

$S.\text{push}(v)$

mientras S no esté vacía

$v \leftarrow S.\text{pop}()$

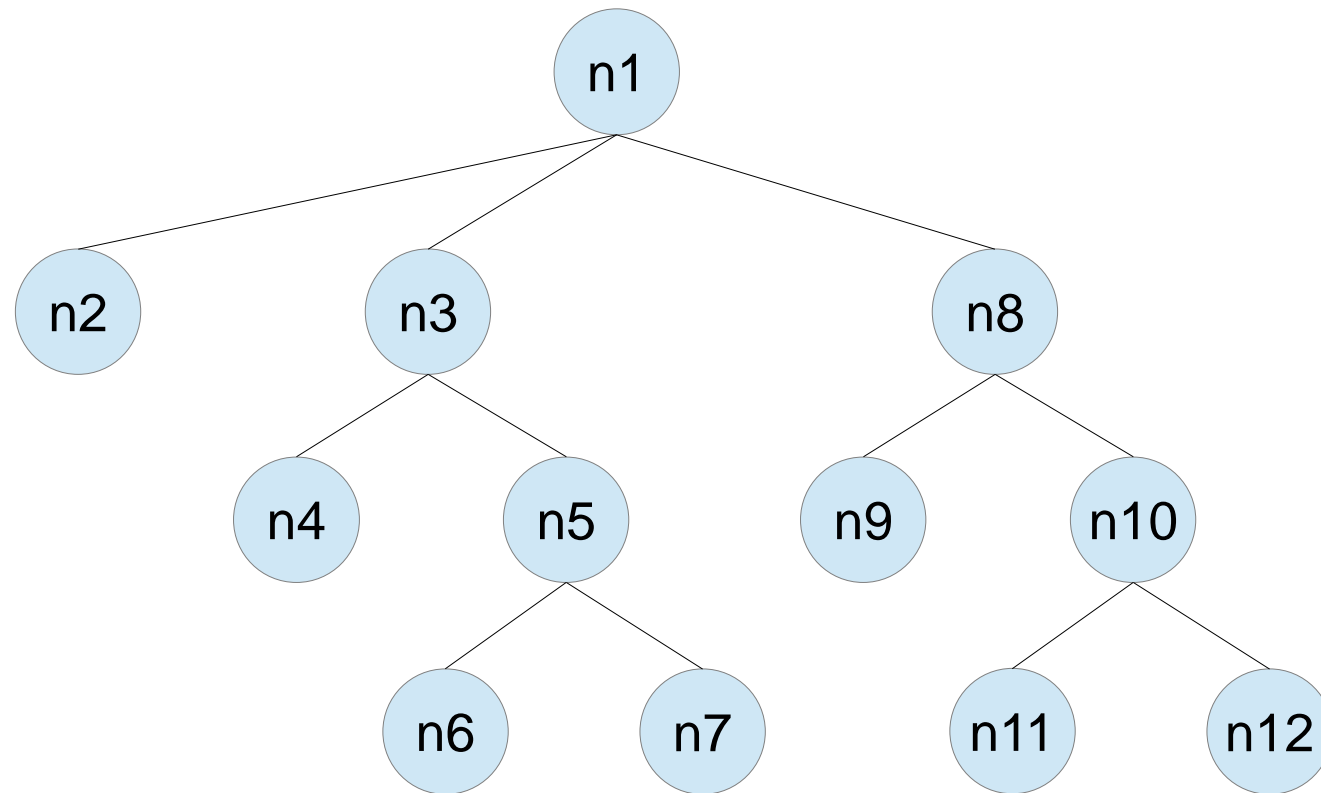
si el vértice v no está marcado como visitado

marcar v como visitado

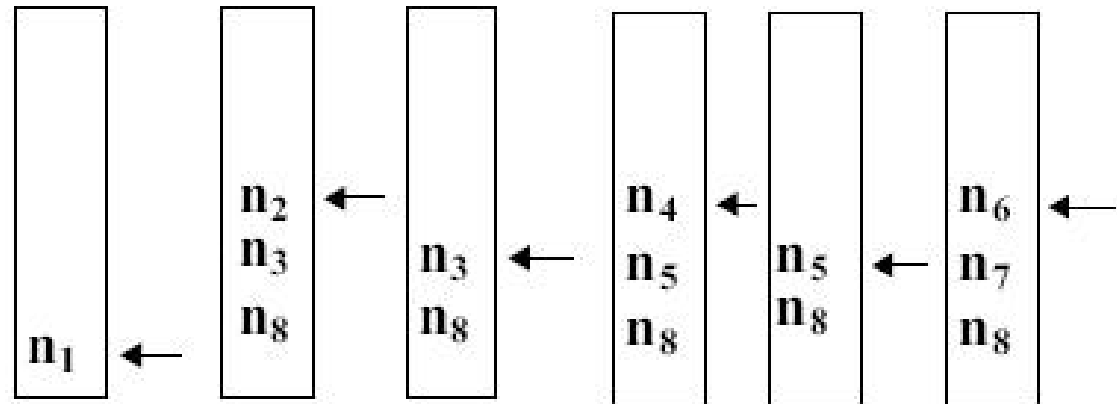
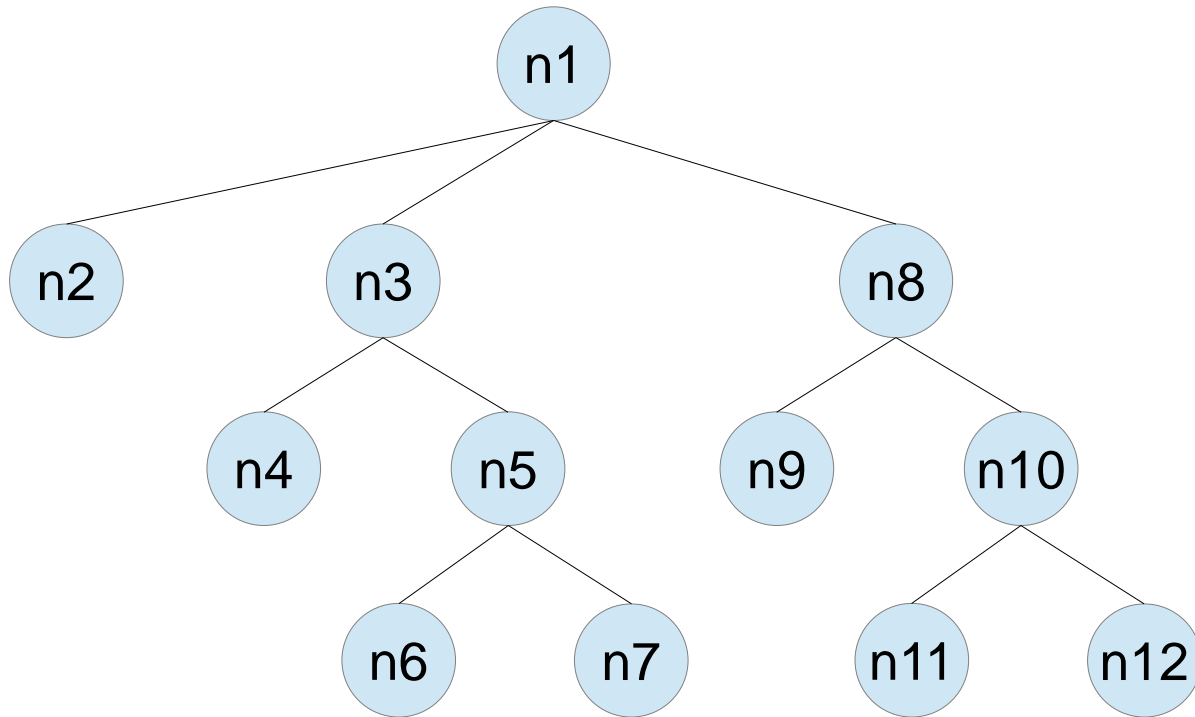
para todas las aristas (v, w) (con v origen)

$S.\text{push}(w)$

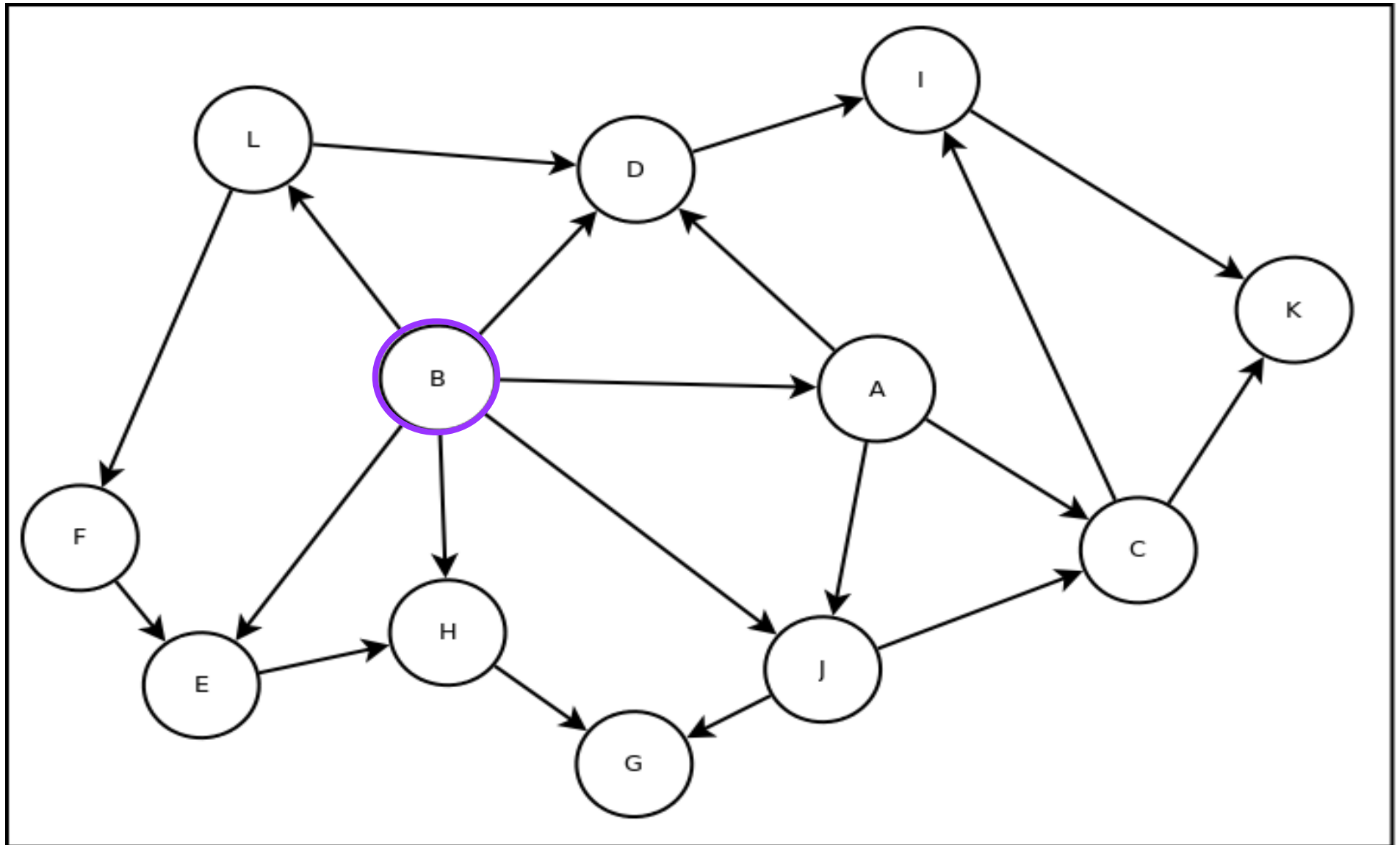
Búsqueda en profundidad (DFS)



Búsqueda en profundidad (DFS)



Búsqueda en profundidad (DFS)



Búsqueda en profundidad (DFS)

DFS (Grafo, B)

↳ DFS (Grafo, A)

↳ DFS (Grafo, C)

↳ DFS (Grafo, I)

↳ DFS (Grafo, K)

↳ DFS (Grafo, D)

↳ DFS (Grafo, J)

↳ DFS (Grafo, G)

↳ DFS (Grafo, E)

↳ DFS (Grafo, H)

↳ DFS (Grafo, L)

↳ DFS (Grafo, F)

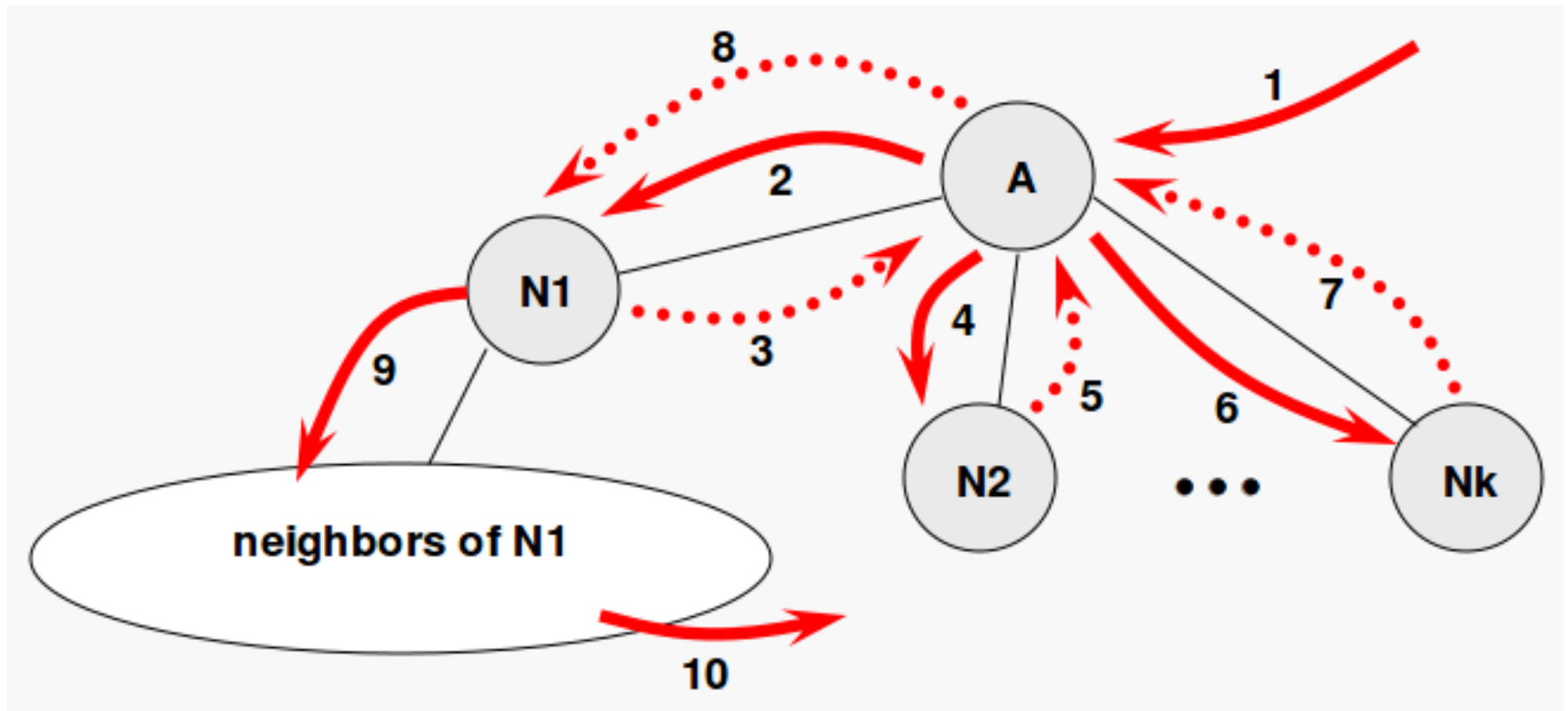
B – A – C – I – K – D – J – G – E – H – L – F

Búsqueda en anchura (BFS)

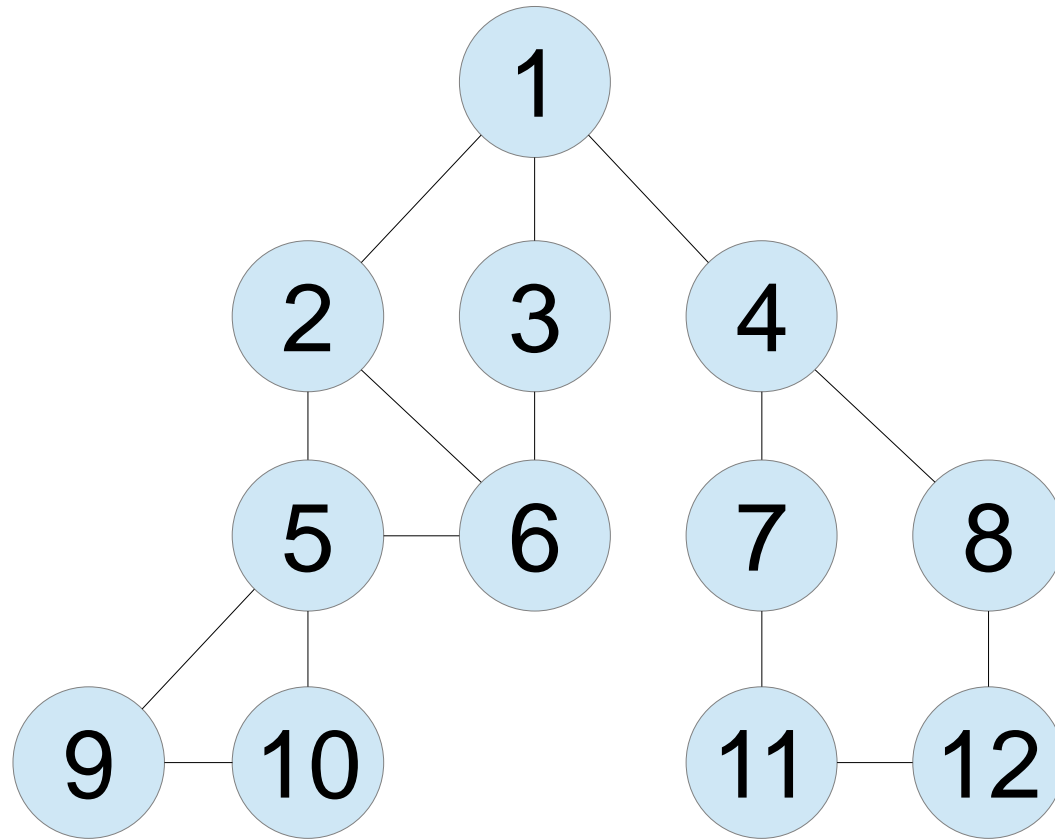
Búsqueda en anchura: *Breadth-first search*.

- Análogo al recorrido por niveles del árbol.
- Para un vértice dado, se visitan todos sus vecinos directos en orden, antes de moverse al siguiente vértice.
- Permite encontrar rutas más cortas entre vértices.
- Complejidad: $O(|E|)$.

Búsqueda en anchura (BFS)



Búsqueda en anchura (BFS)



orden en que los nodos son visitados

Búsqueda en anchura (BFS)

Implementación con cola:

procedimiento BFS (G, v)

sea Q una cola

$Q.push(v)$

mientras Q no esté vacía

$v \leftarrow Q.pop()$

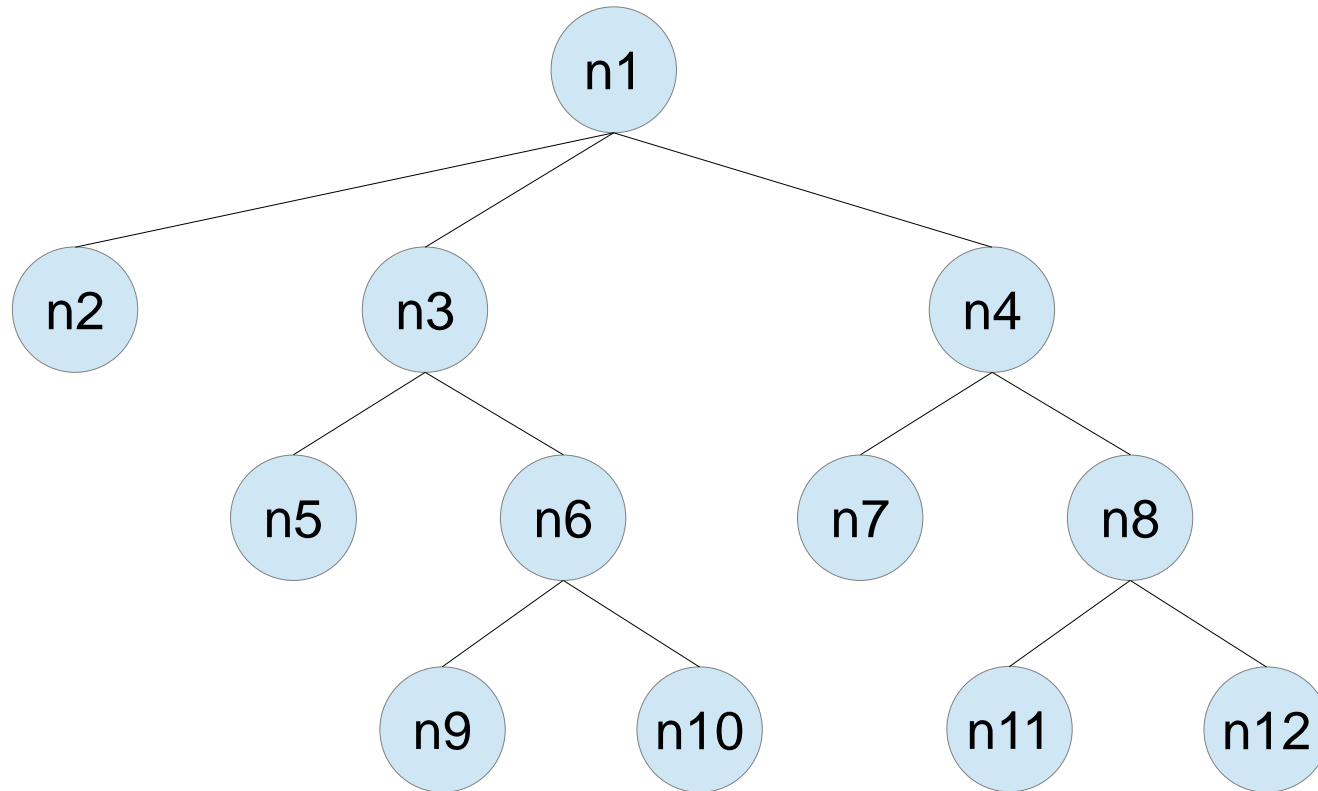
si el vértice v no está marcado como visitado

marcar v como visitado

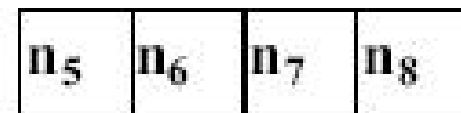
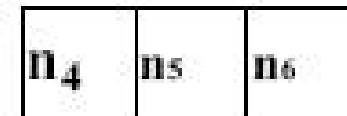
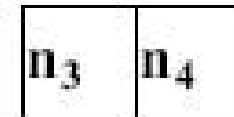
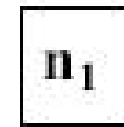
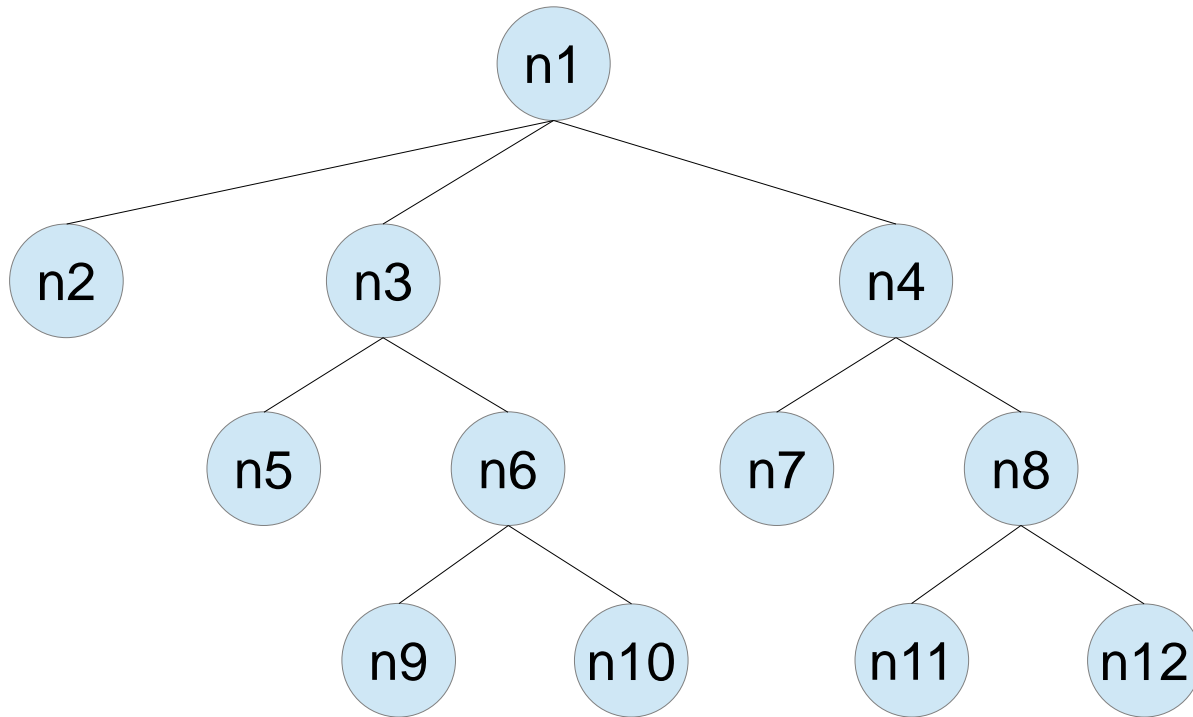
para todas las aristas (v, w) (con v origen)

$Q.push(w)$

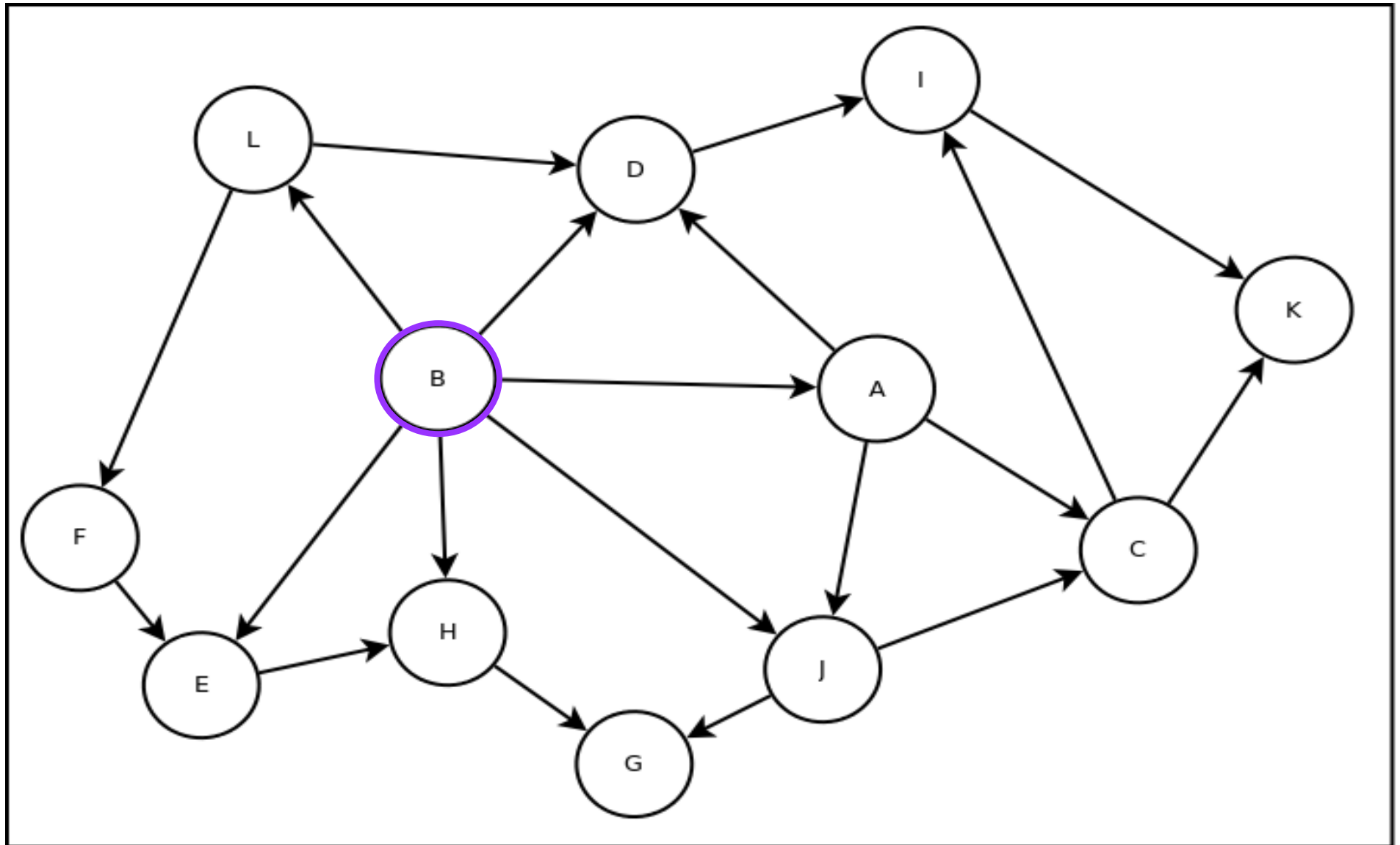
Búsqueda en anchura (BFS)



Búsqueda en anchura (BFS)



Búsqueda en anchura (BFS)



Búsqueda en anchura (BFS)

B

A D E H J L

D E H J L C

E H J L C I

H J L C I

J L C I G

L C I G

C I G F

I G F K

G F K

F K

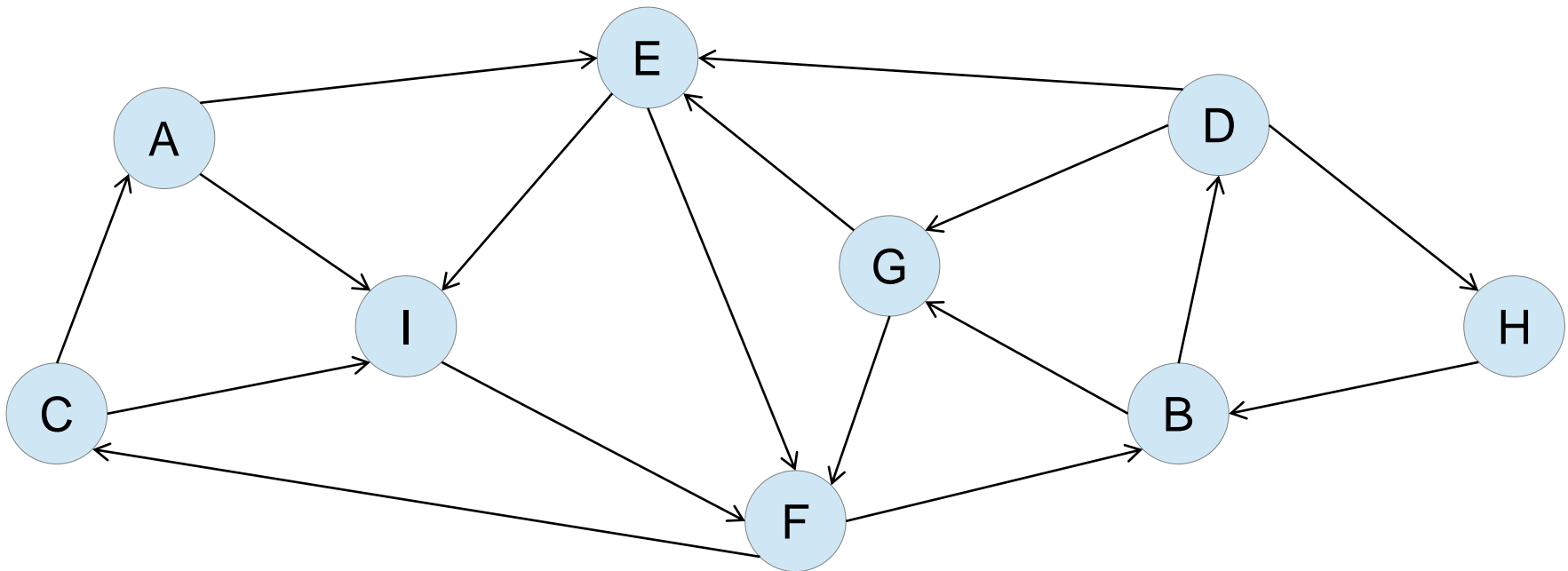
K

B – A – D – E – H – J – L – C – I – G – F – K

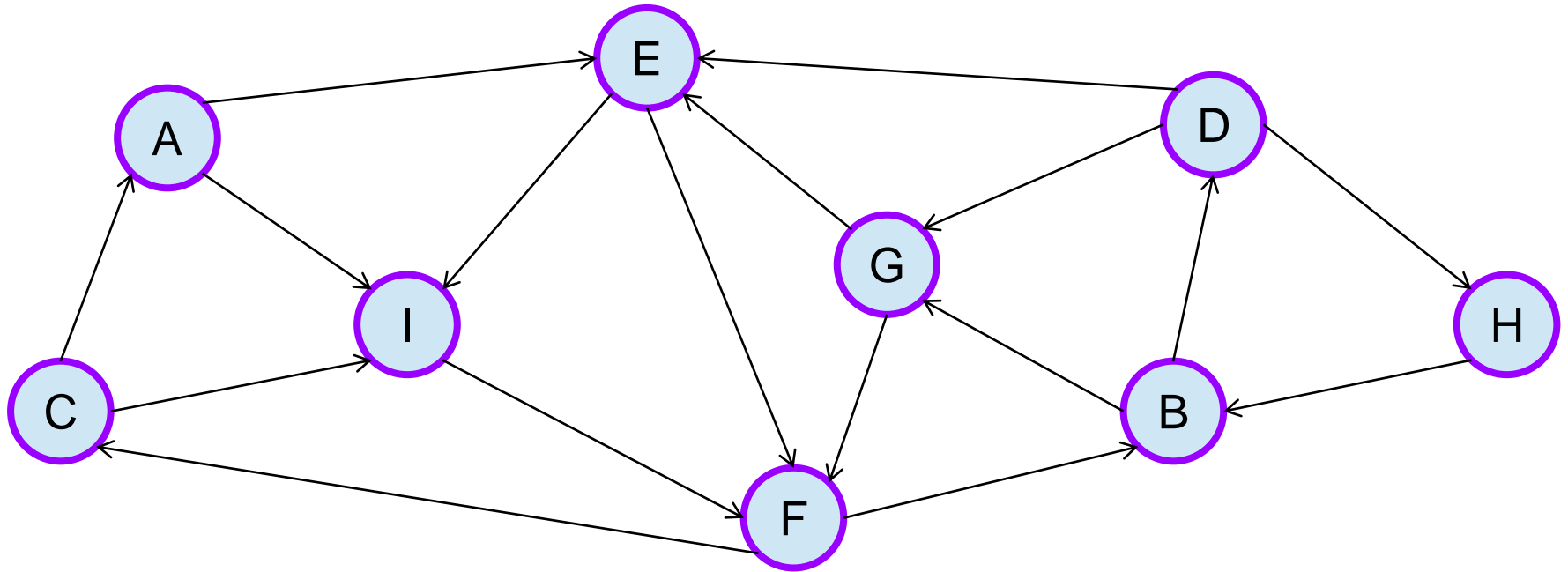
¡ Quiz !

.Mencione una de las aplicaciones de los recorridos en grafos.

.Para el siguiente grafo, genere: recorrido plano, en anchura (desde D), en profundidad (desde H).



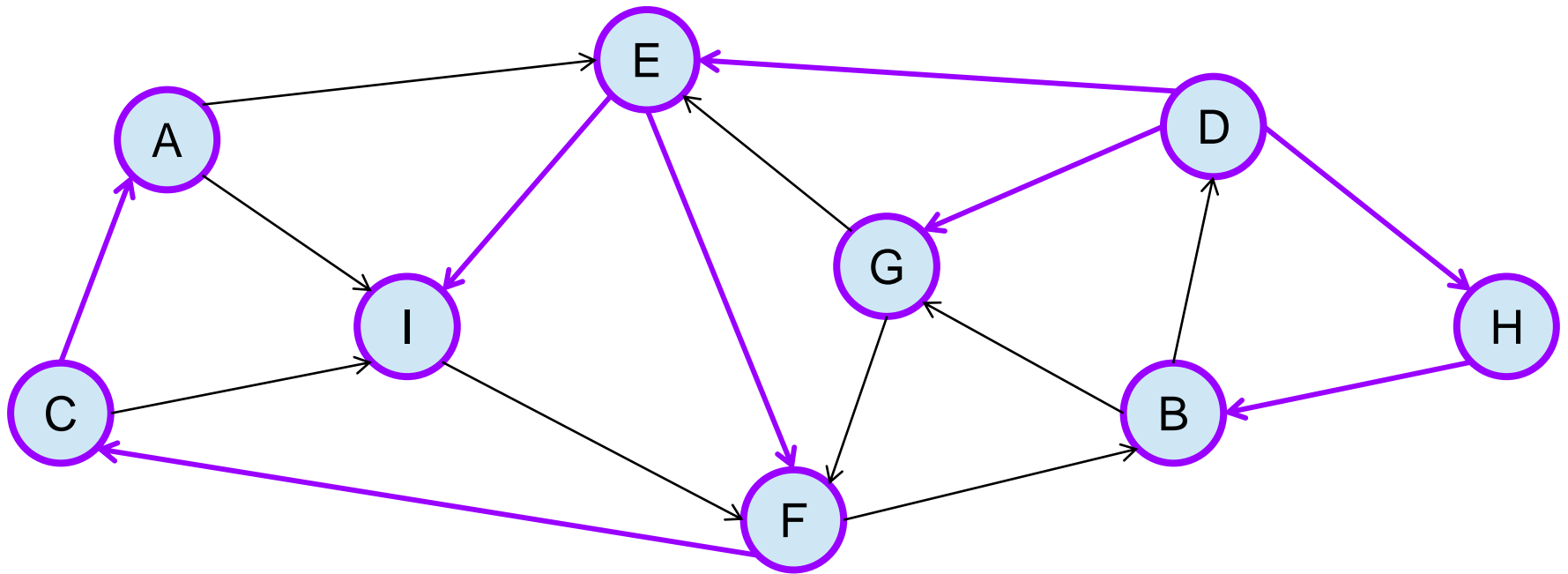
Recorridos en Grafos



.Recorrido plano:

.A – E – D – H – B – G – F – I – C

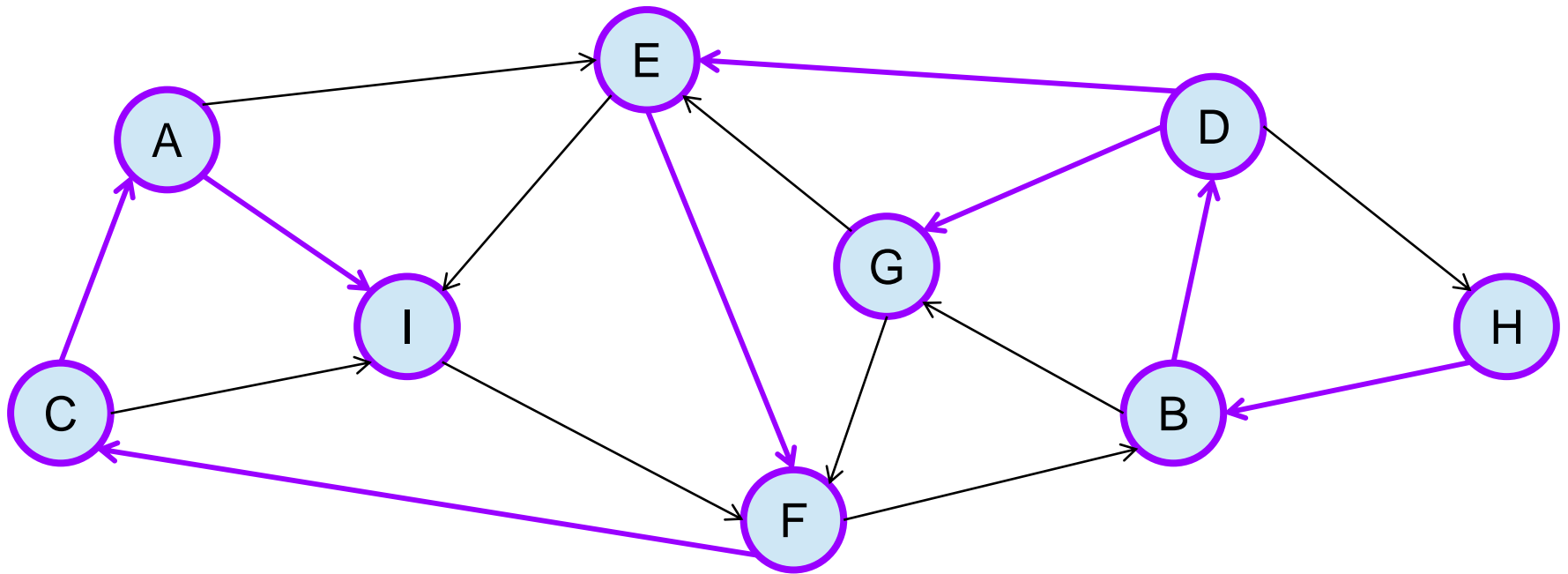
Recorridos en Grafos



.Recorrido en anchura (desde D):

.D – E – G – H – F – I – B – C – A

Recorridos en Grafos



• Recorrido en profundidad (desde H):

• H – B – D – E – F – C – A – I – G

Recorridos en Grafos

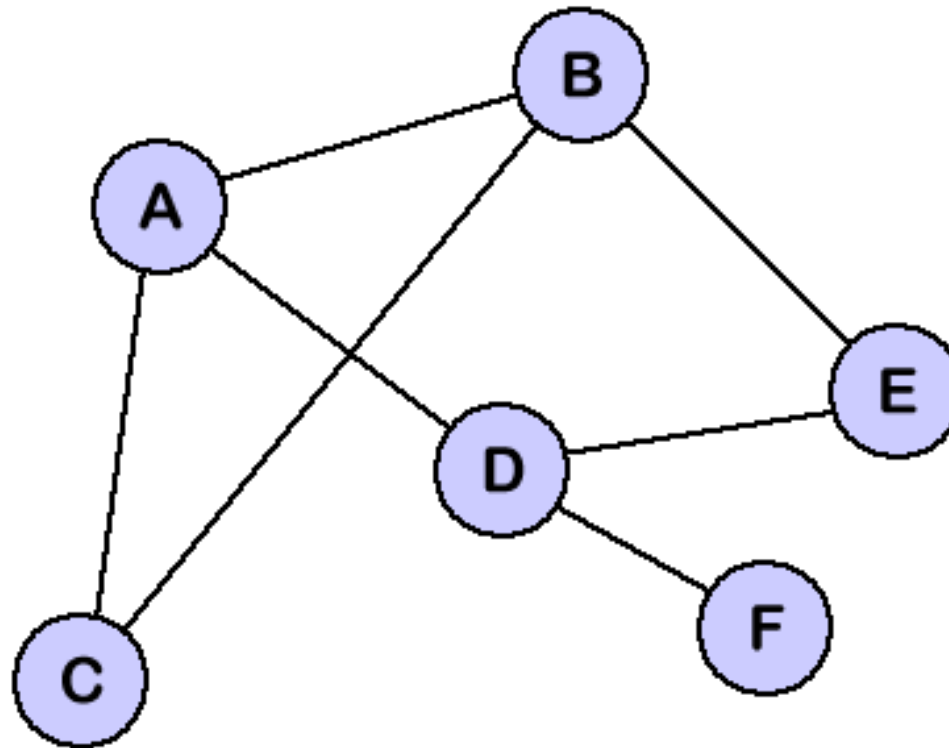
- Exploración:

- Applet de recorridos en grafos:

- <http://visualgo.net/dfsdfs.html>

Recorridos en Grafos

•Ejercicio:



<http://courses.cs.vt.edu/csonline/DataStructures/Lessons/Graphs/index.html>

- Recorrido DFS desde cada nodo (orden alfabético).
- Recorrido BFS desde cada nodo (orden alfabético).

Recorridos en Grafos

.Recorrido DFS (profundidad) desde cada nodo:

.A – B – C – E – D – F

.B – A – C – D – E – F

.C – A – B – E – D – F

.D – A – B – C – E – F

.E – B – A – C – D – F

.F – D – A – B – C – E

Recorridos en Grafos

.Recorrido BFS (anchura) desde cada nodo:

.A – B – C – D – E – F

.B – A – C – E – D – F

.C – A – B – D – E – F

.D – A – E – F – B – C

.E – B – D – A – C – F

.F – D – A – E – B – C

Referencias

[.www.cse.ohio-state.edu/~gurari/course/cis680/cis680Ch14.html](http://www.cse.ohio-state.edu/~gurari/course/cis680/cis680Ch14.html)

[.www.mpi-inf.mpg.de/~mehlhorn/ftp/Toolbox/GraphTraversal.pdf](http://www.mpi-inf.mpg.de/~mehlhorn/ftp/Toolbox/GraphTraversal.pdf)

[.courses.cs.vt.edu/~cs3114/Fall09/wmcquain/Notes/T20.GraphTraversals.pdf](http://courses.cs.vt.edu/~cs3114/Fall09/wmcquain/Notes/T20.GraphTraversals.pdf)

[.webdocs.cs.ualberta.ca/~holte/T26/graph-traversal.html](http://webdocs.cs.ualberta.ca/~holte/T26/graph-traversal.html)

[.en.wikipedia.org/wiki/Graph_traversal](http://en.wikipedia.org/wiki/Graph_traversal)

Referencias

[.www.cs.cornell.edu/courses/CS2112/2012sp/lectures/lec24/lec24-12sp.html](http://www.cs.cornell.edu/courses/CS2112/2012sp/lectures/lec24/lec24-12sp.html)

[.www.comp.nus.edu.sg/~stevenha/visualization/dfsdfs.html](http://www.comp.nus.edu.sg/~stevenha/visualization/dfsdfs.html)

[.www.sci.brooklyn.cuny.edu/~amotz/BC-ALGORITHMS/PRESENTATIONS/traversals.pdf](http://www.sci.brooklyn.cuny.edu/~amotz/BC-ALGORITHMS/PRESENTATIONS/traversals.pdf)