

# Recorridos de Grafos

---

Ing. Luis Humberto González

Los recorridos sobre un grafo más utilizados son:

- ✓ Breadth First (Primero en Anchura)
- ✓ Depth First (Primero en Profundidad)

# Recorridos sobre un Grafo

---

- ✓ A partir de un Nodo de Inicio, este algoritmo “visita” un Nodo y luego a todos sus Vecinos.
- ✓ El orden en el que visita a los Vecinos depende del “orden” de almacenamiento, según la representación que se esté utilizando.
- ✓ Este algoritmo utiliza una FILA auxiliar.
- ✓ Para evitar ciclos infinitos, todos los Nodos guardan un “status” (En Espera, Listo, Procesado)

# Breadth First

---

Iniciar el status de todos los Nodos a “En Espera”.

Para cada Nodo del Grafo:

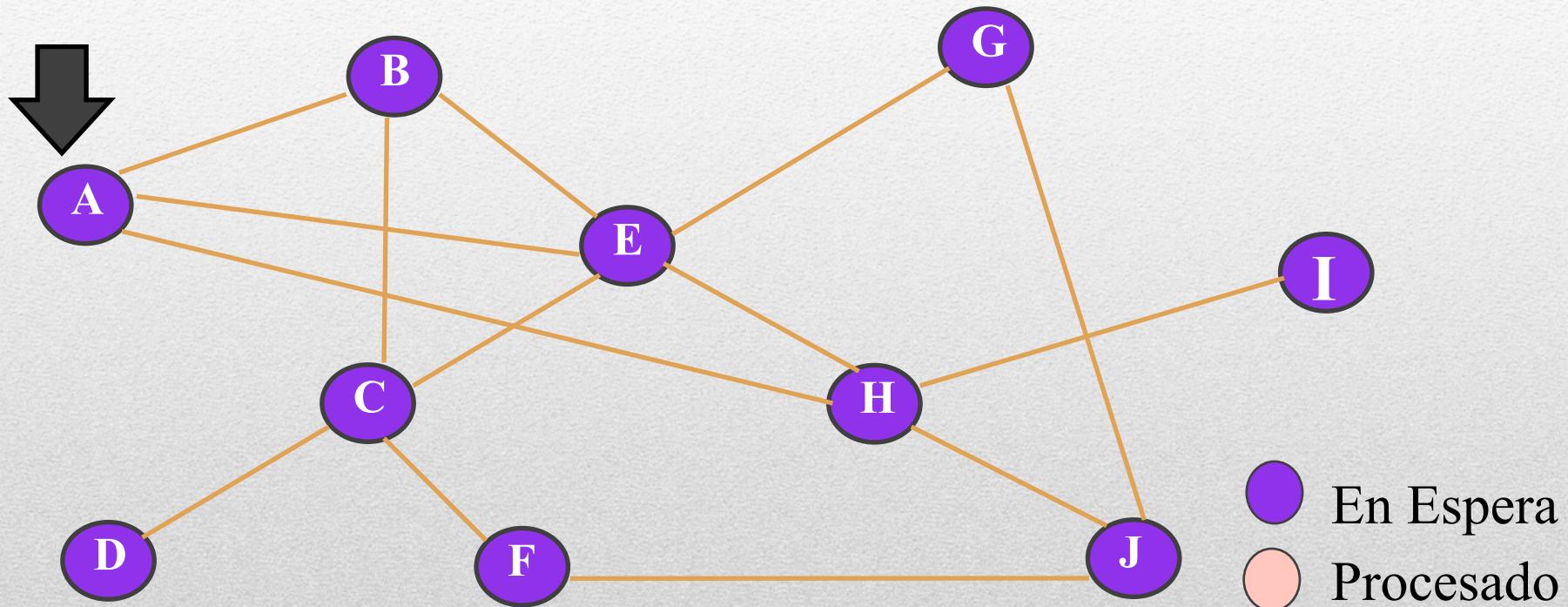
Si el status del Nodo es “En Espera”, entonces:

- Insertar el Nodo en la FILA
- Mientras la FILA no esté vacía:
  - \* Sacar Nodo de la FILA y procesarlo.
  - \* Cambiar su status a “Procesado”.
  - \* Meter a la FILA todos los Vecinos del Nodo que tengan status “En Espera”, cambiándolo a “Listo”.

# Algoritmo Breadth First

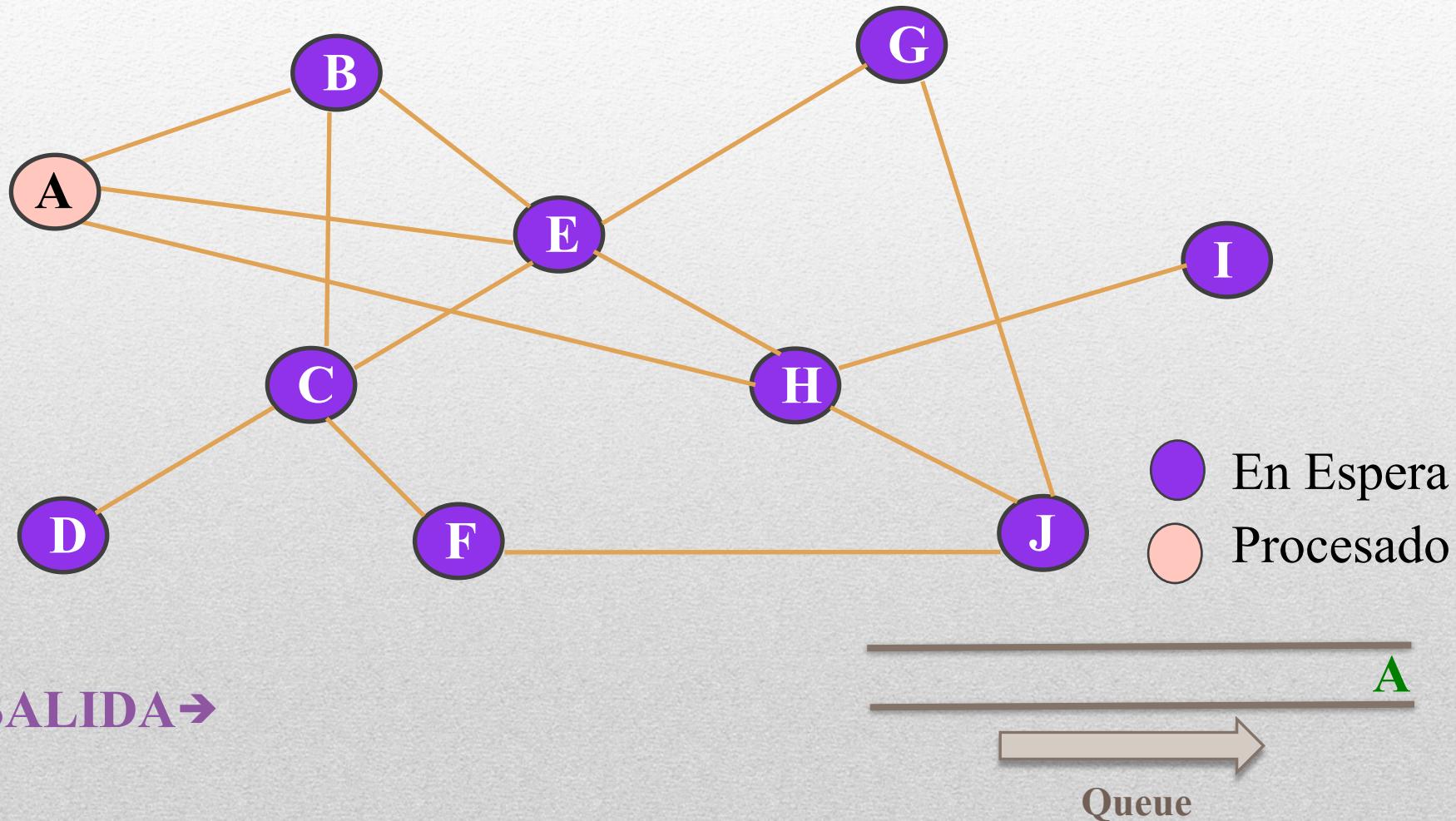
---

# Ejemplo... Breadth First

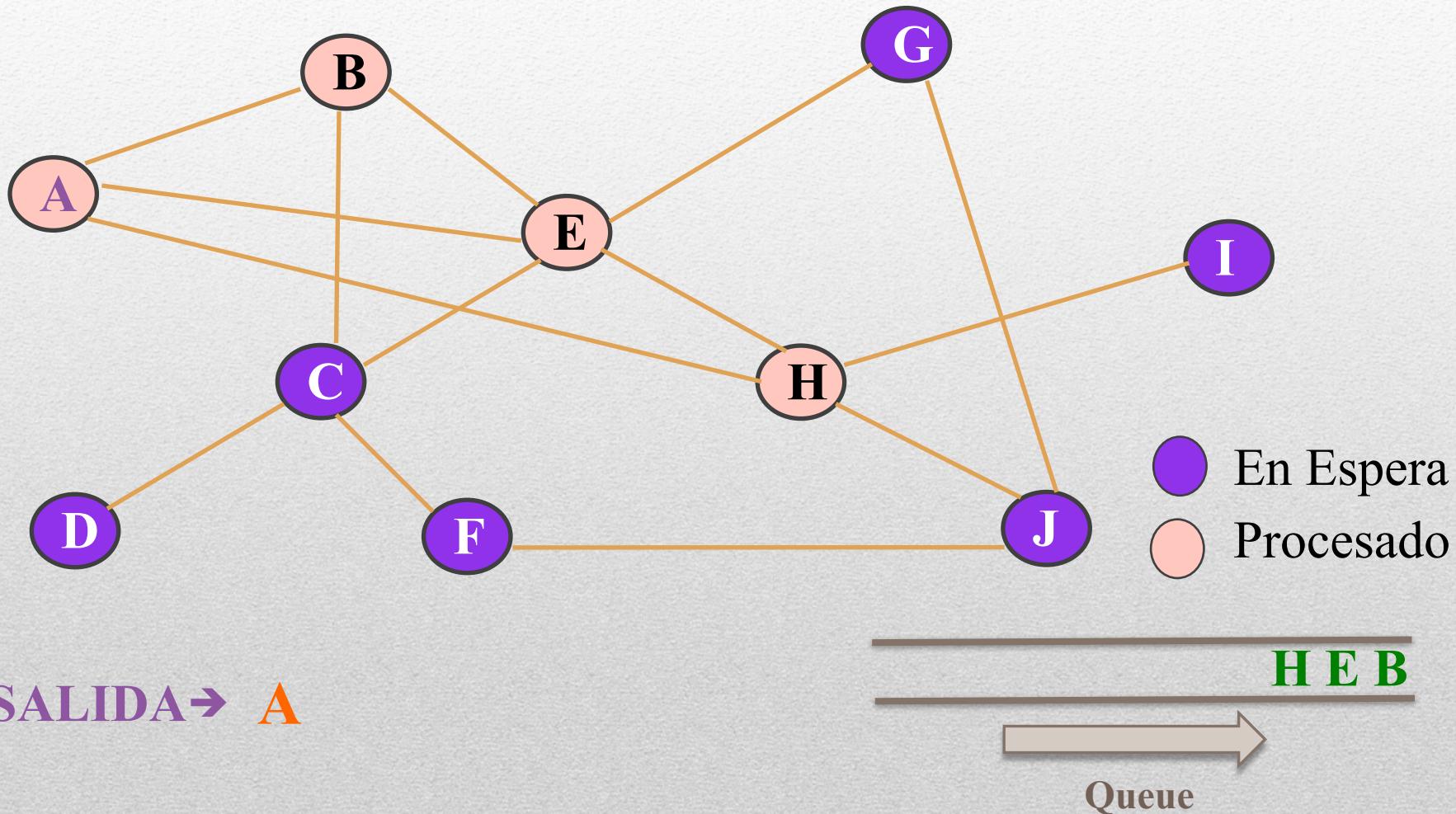


Suponer que están almacenados en orden alfabético.

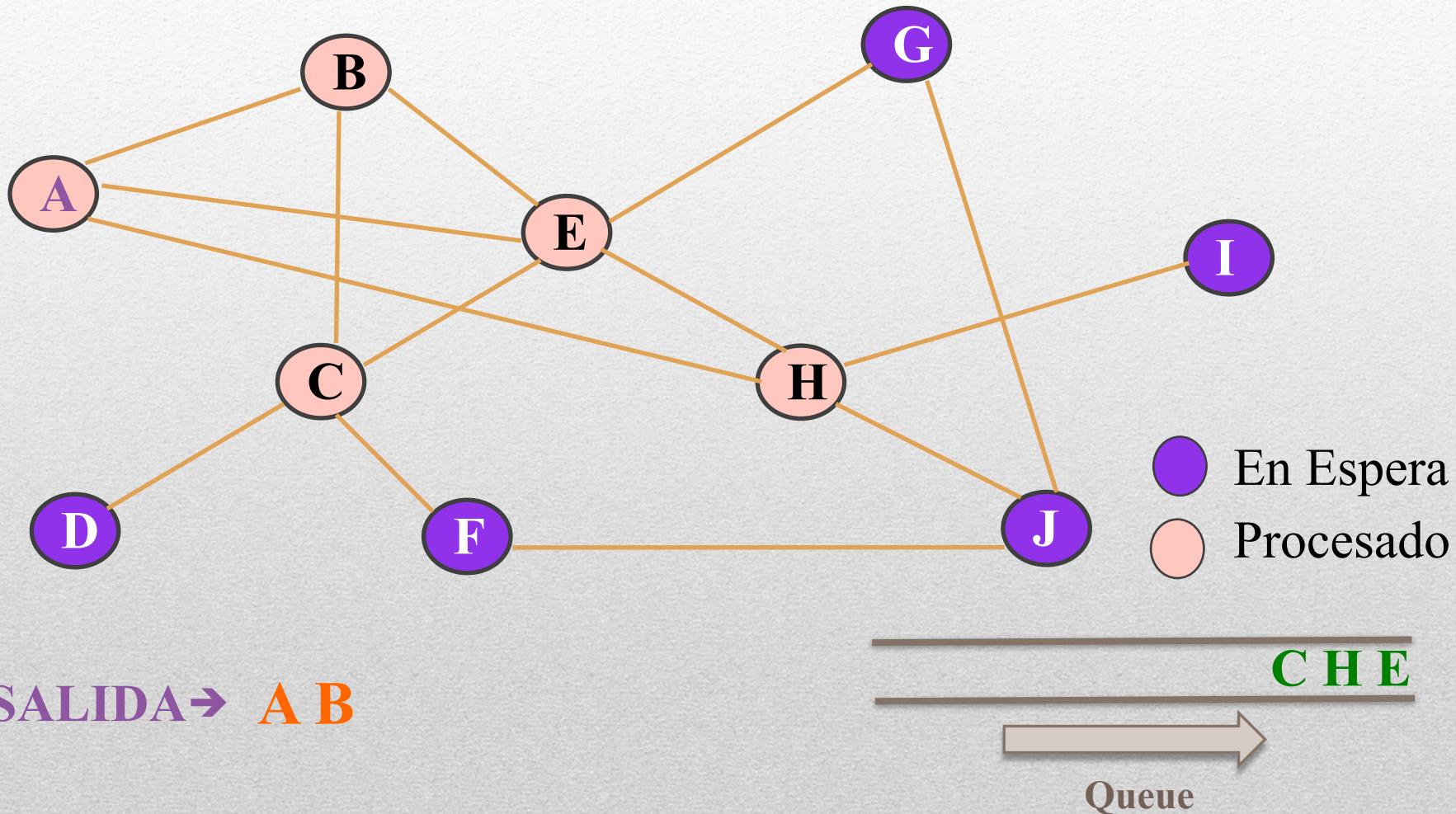
# Ejemplo... Breadth First



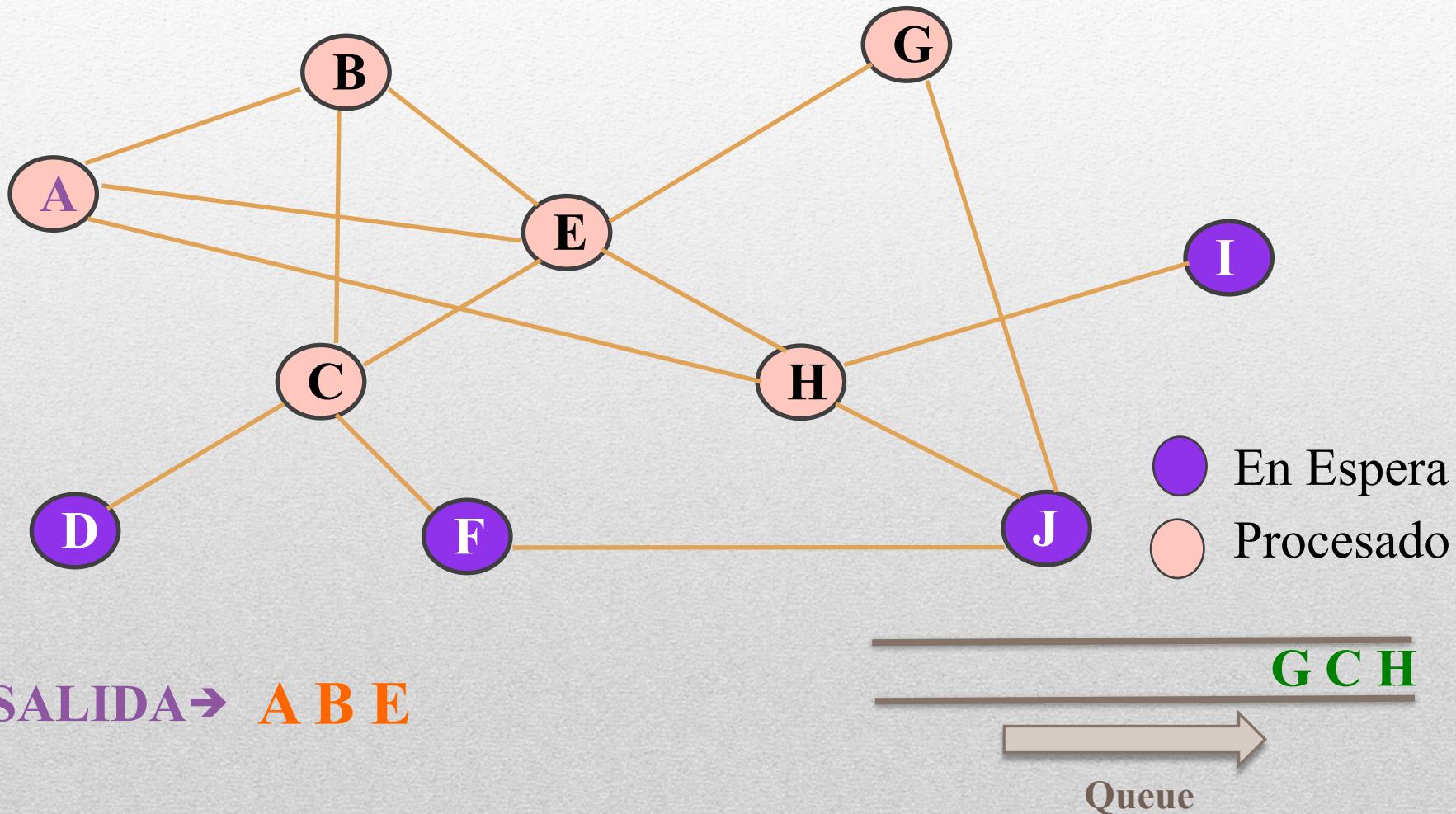
# Ejemplo... Breadth First



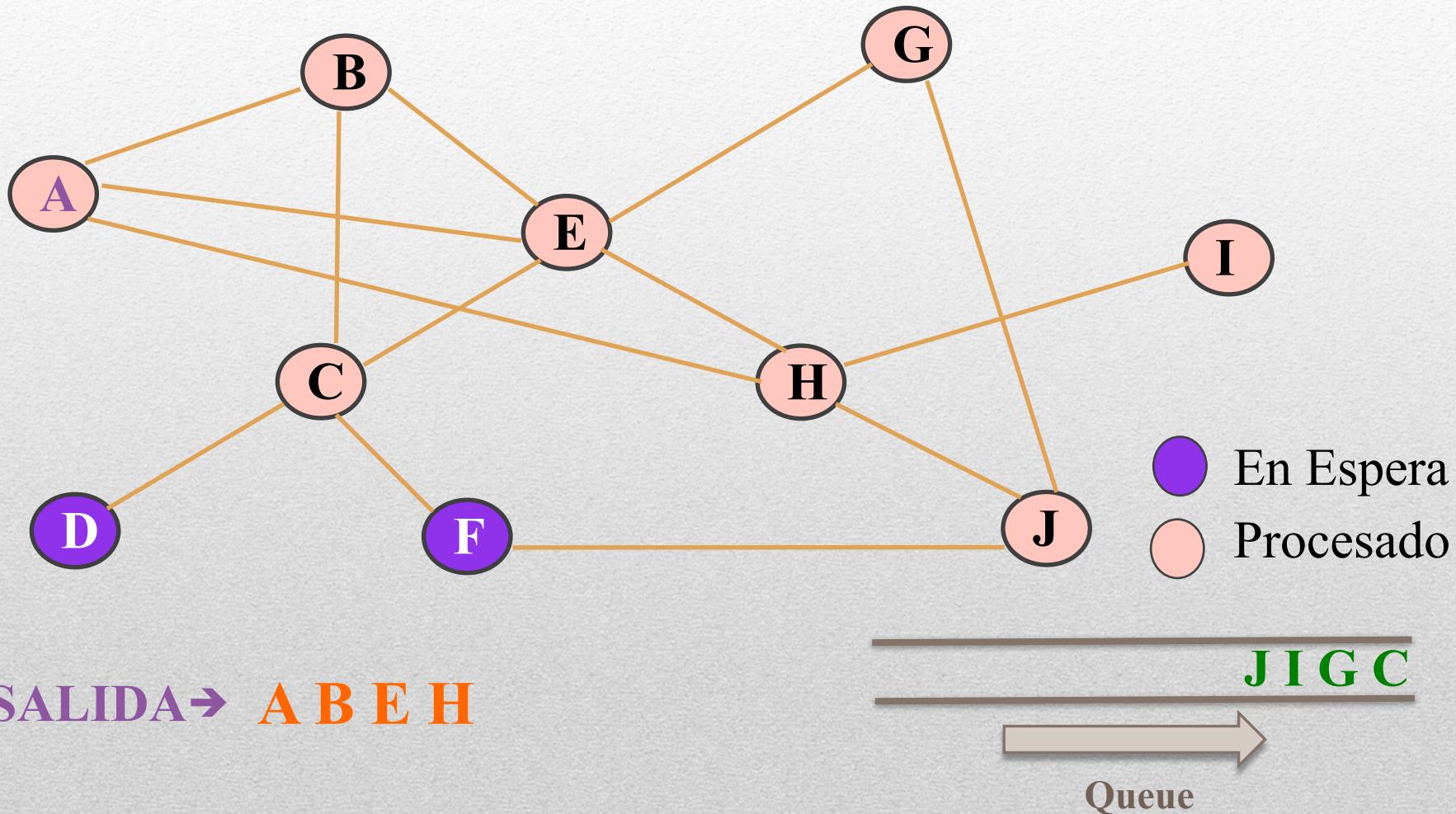
# Ejemplo... Breadth First



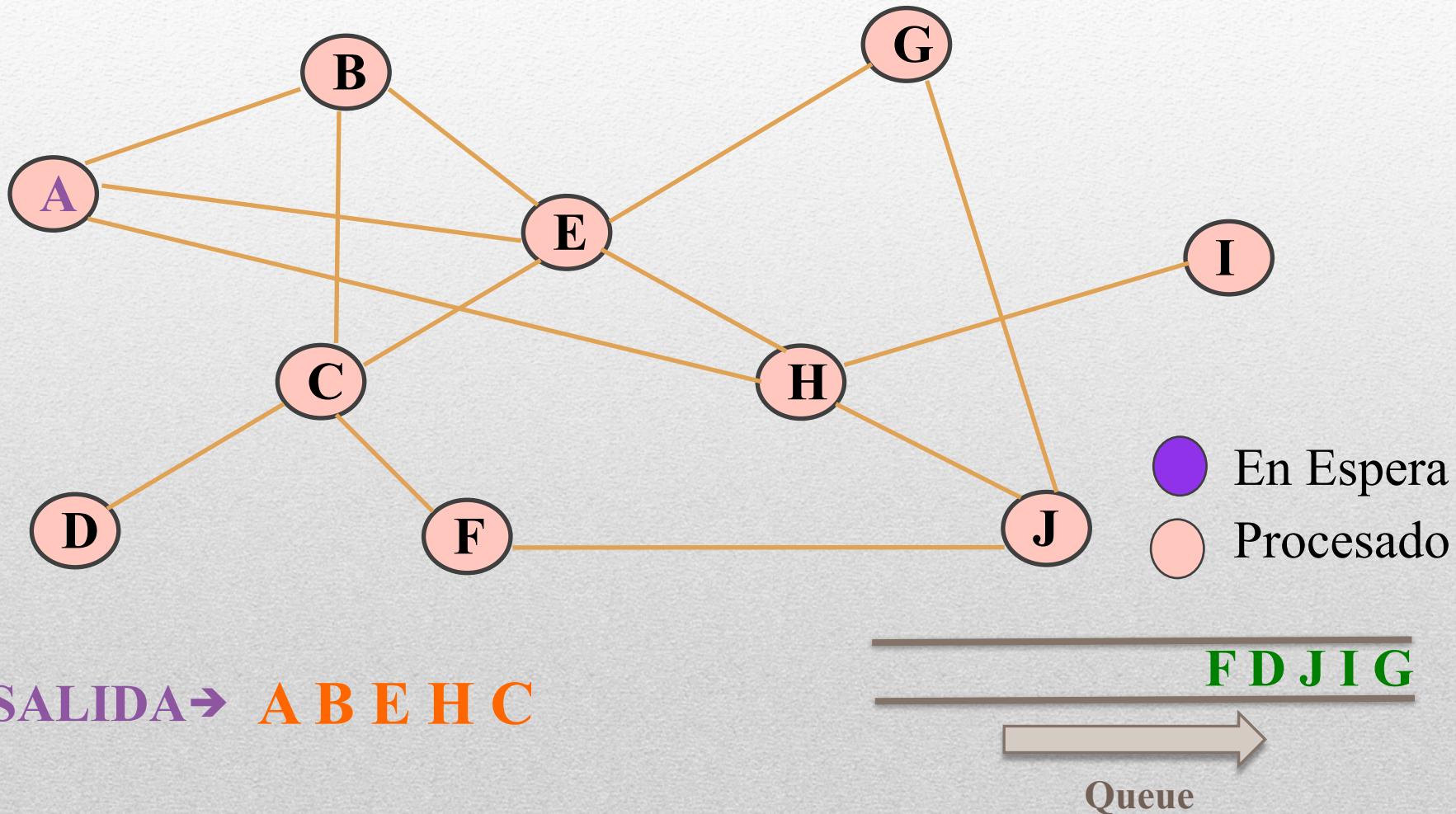
# Ejemplo... Breadth First



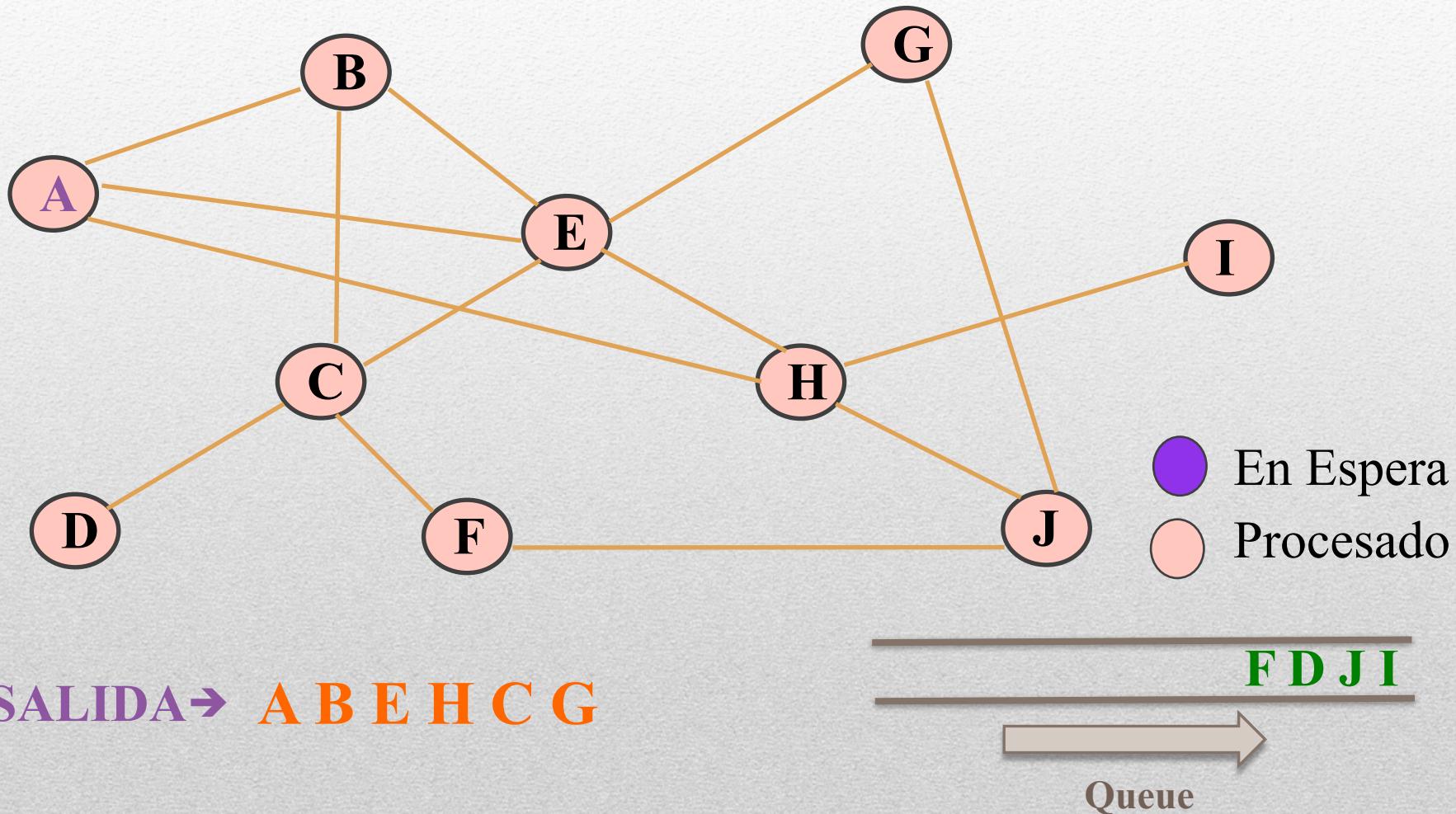
# Ejemplo... Breadth First



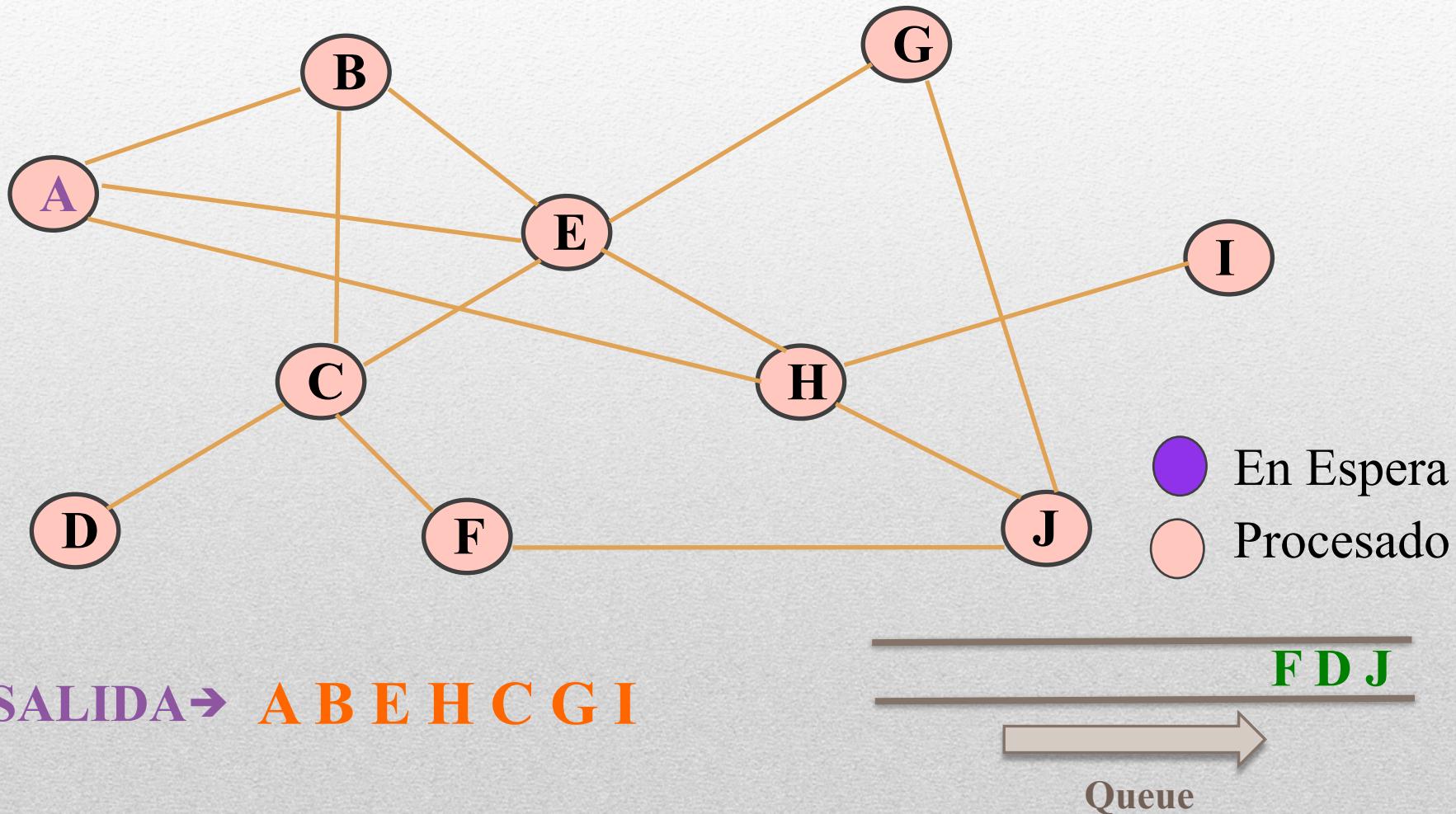
# Ejemplo... Breadth First



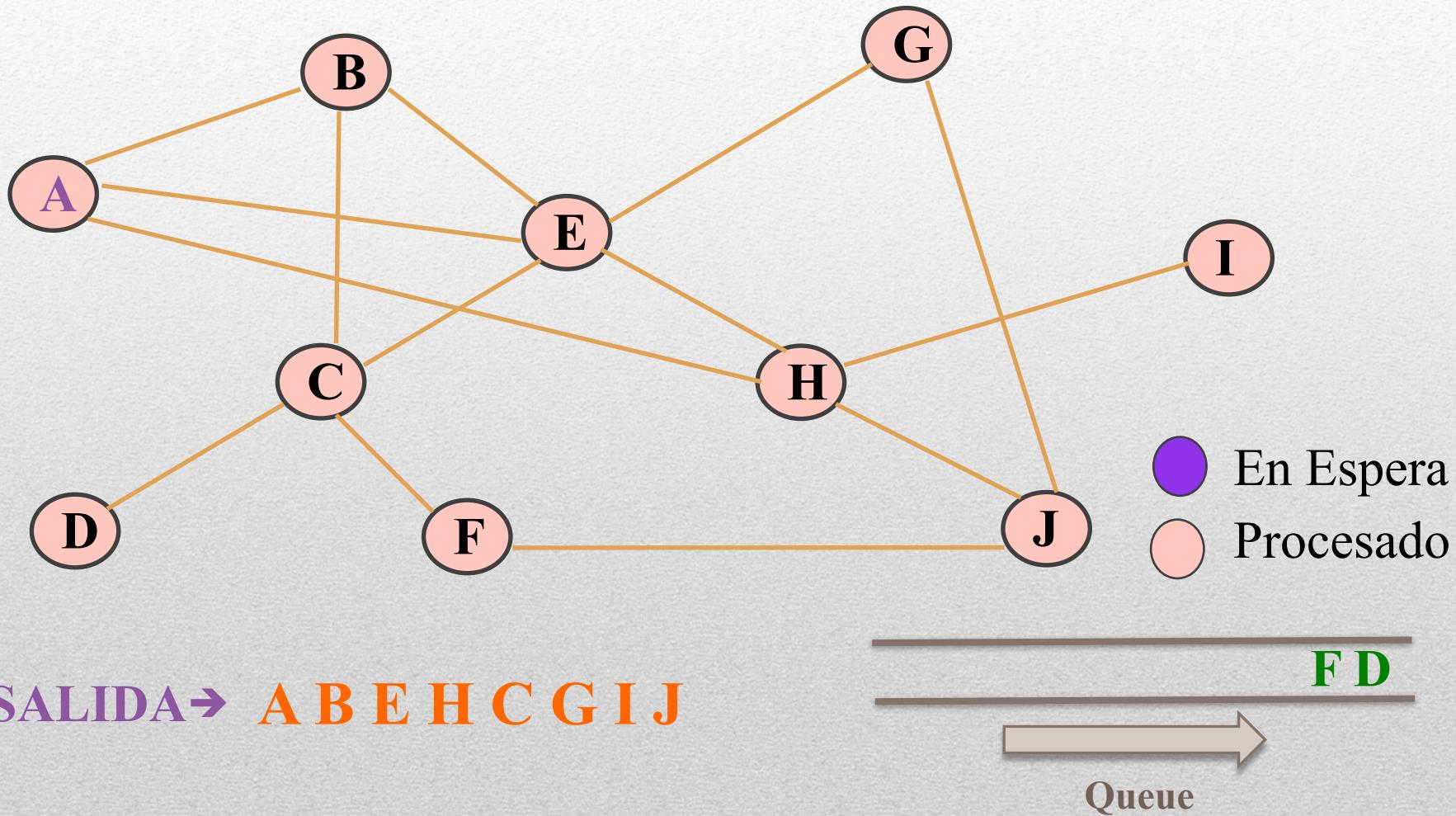
# Ejemplo... Breadth First



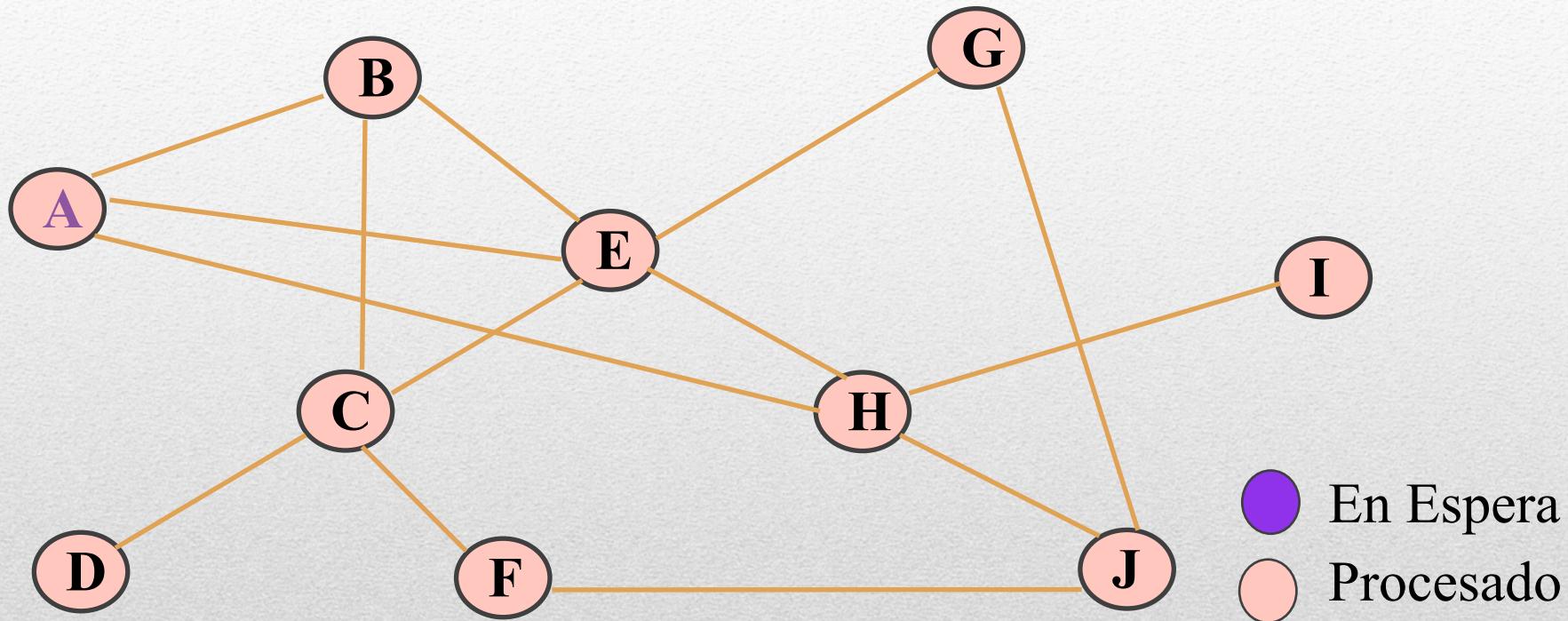
# Ejemplo... Breadth First



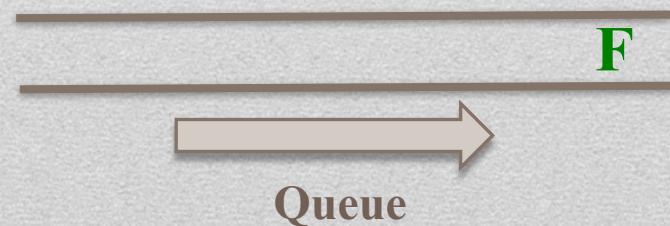
# Ejemplo... Breadth First



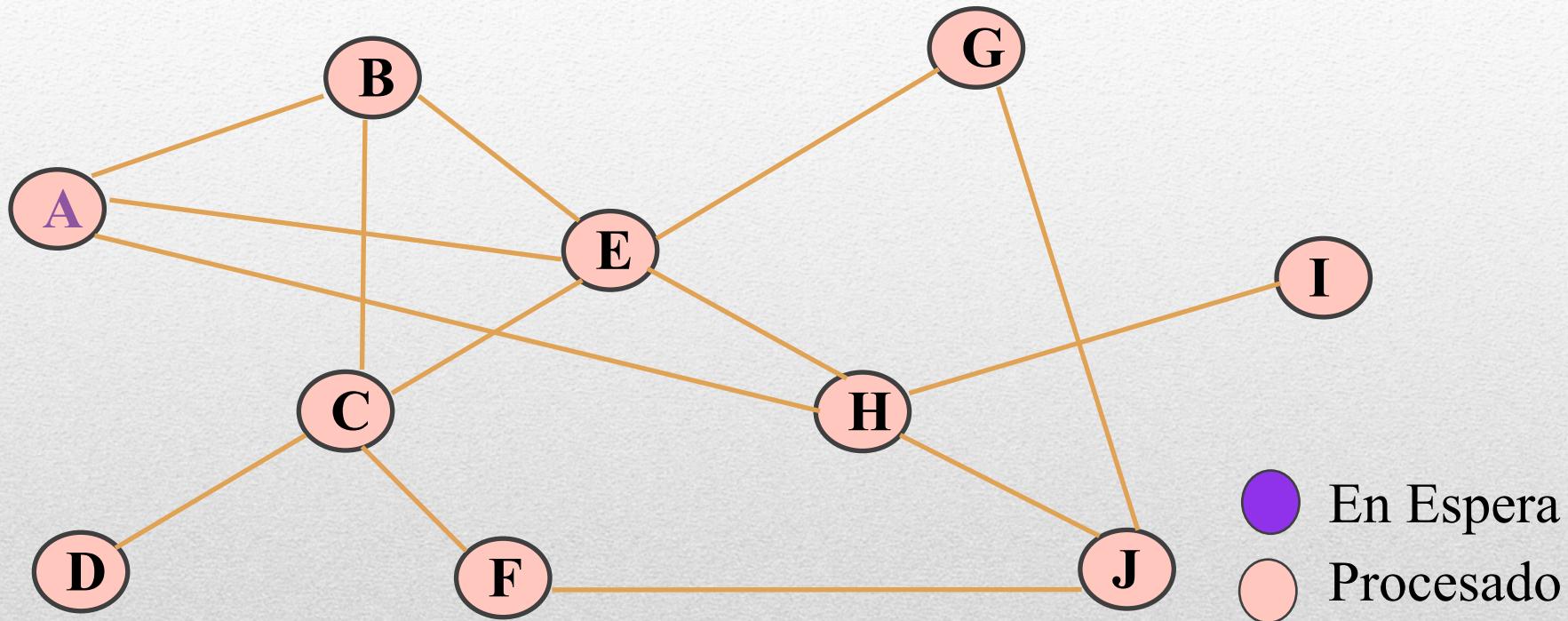
# Ejemplo... Breadth First



SALIDA → A B E H C G I J D



# Ejemplo... Breadth First



SALIDA → A B E H C G I J D F

Queue

- ✓ A partir de un Nodo de Inicio, este algoritmo “visita” el Nodo y luego visita recursivamente a todos sus Vecinos NO procesados.
- ✓ El orden en el que visita a los Vecinos depende del “orden” de almacenamiento, según la representación que se esté utilizando.
- ✓ Este algoritmo utiliza una PILA auxiliar (**para simular la recursividad**).
- ✓ Para evitar ciclos infinitos, todos los Nodos guardan un “**status**” (En Espera, Procesado)

# Depth First

---

Iniciar el status de todos los Nodos a “En Espera”.  
Para cada Nodo del Grafo:

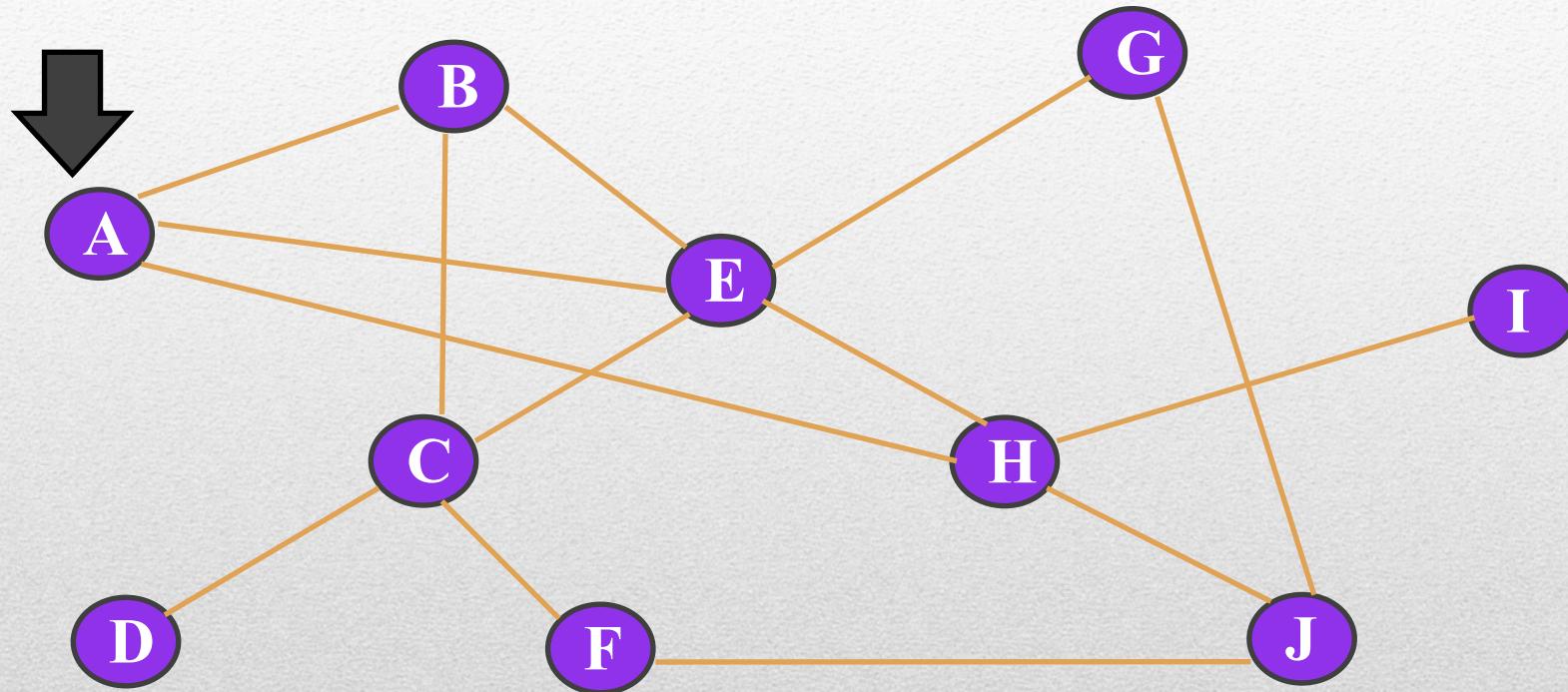
Si el status del Nodo es “En Espera”, entonces:

- Insertar el Nodo en la PILA.
- Mientras la PILA no esté vacía:
  - \* Sacar Nodo de la PILA y procesarlo.
  - \* Cambiar su status a “Procesado”.
  - \* Meter a la PILA todos los Vecinos del Nodo que NO tengan status “Procesado”.
  - \* Si un Nodo ya existía, se deja el último que llegó.

# Algoritmo Depth First

---

# Ejemplo... Depth First

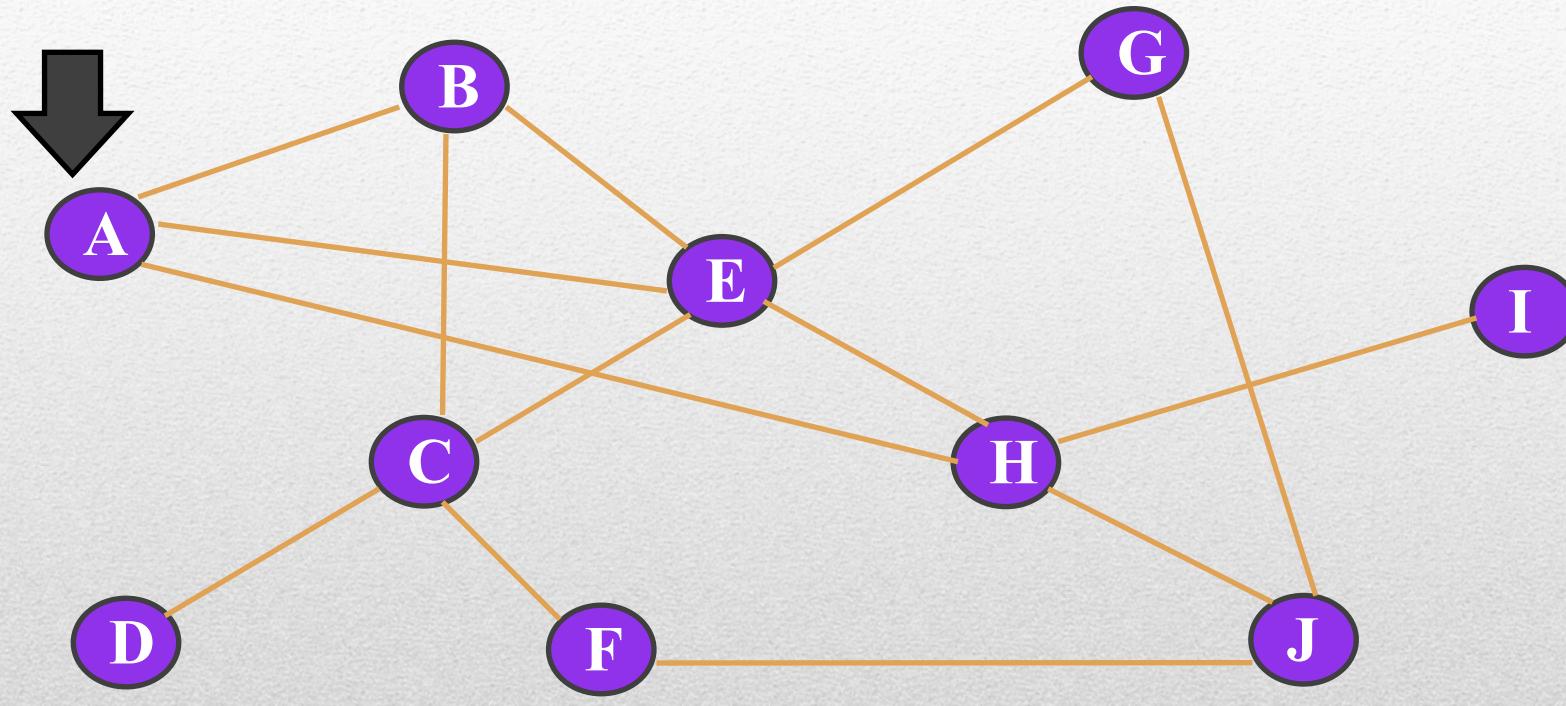


- En Espera
- Procesado

Suponer que están almacenados en orden alfabético.

Stack

# Ejemplo... Depth First



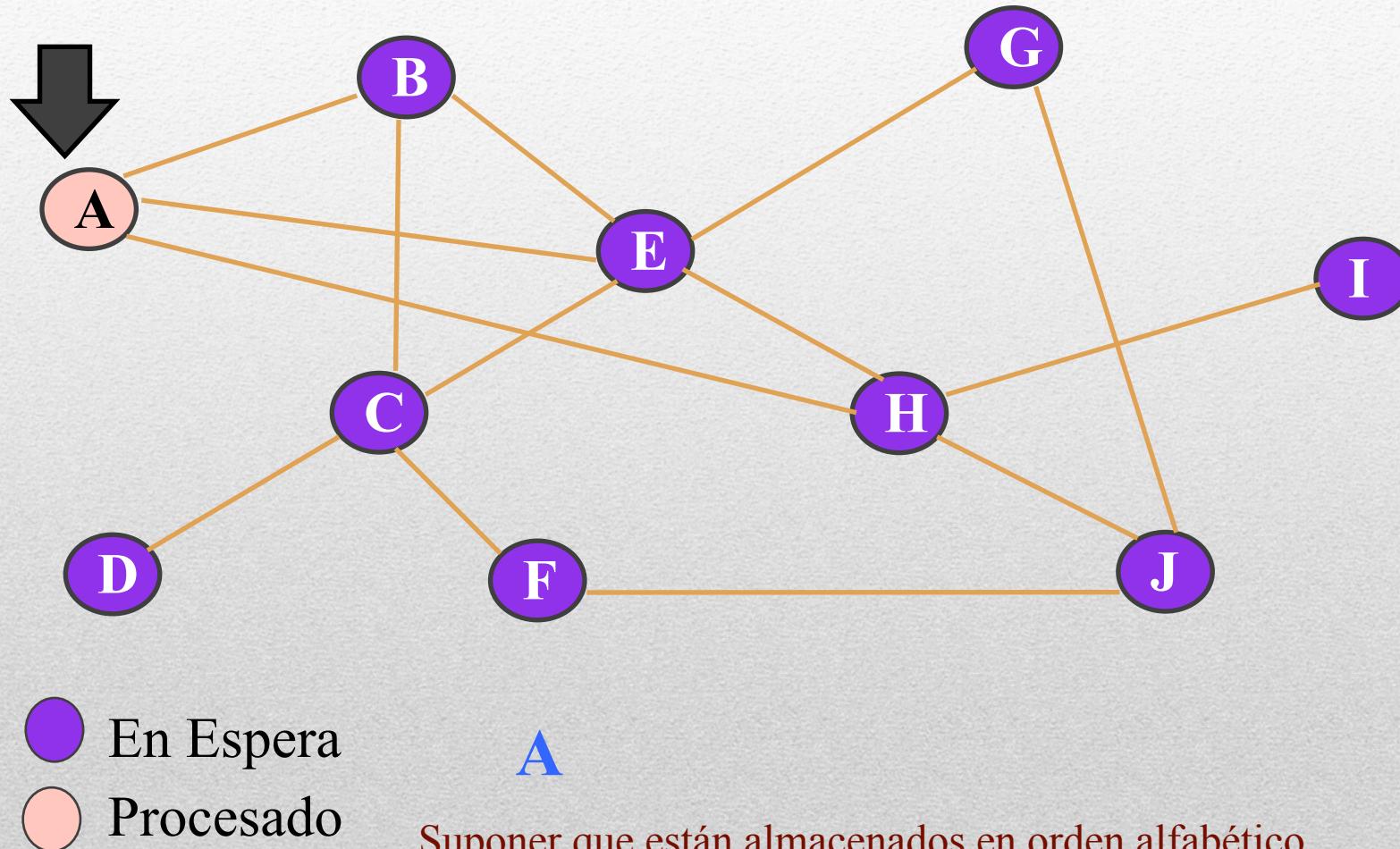
- En Espera
- Procesado

Suponer que están almacenados en orden alfabético.

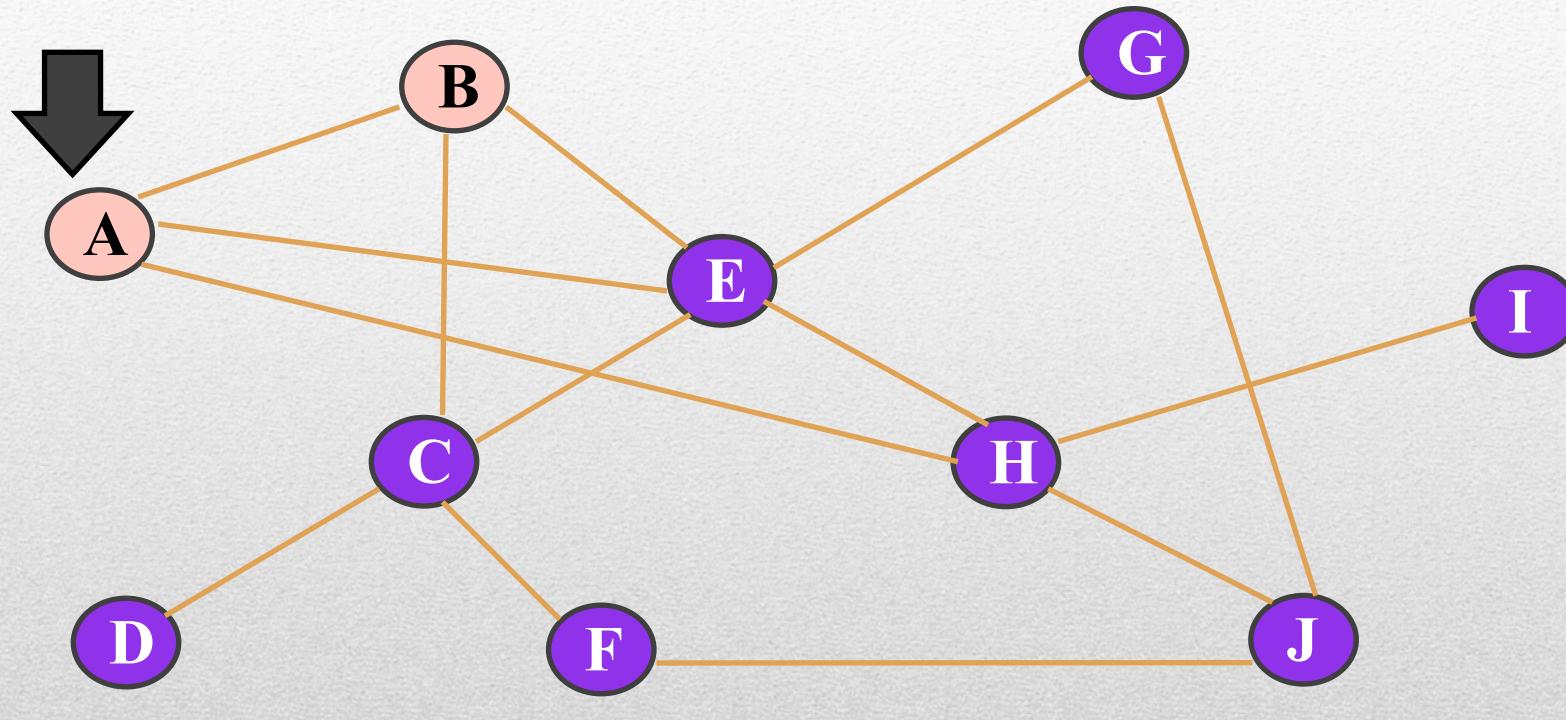
A

Stack

# Ejemplo... Depth First



# Ejemplo... Depth First



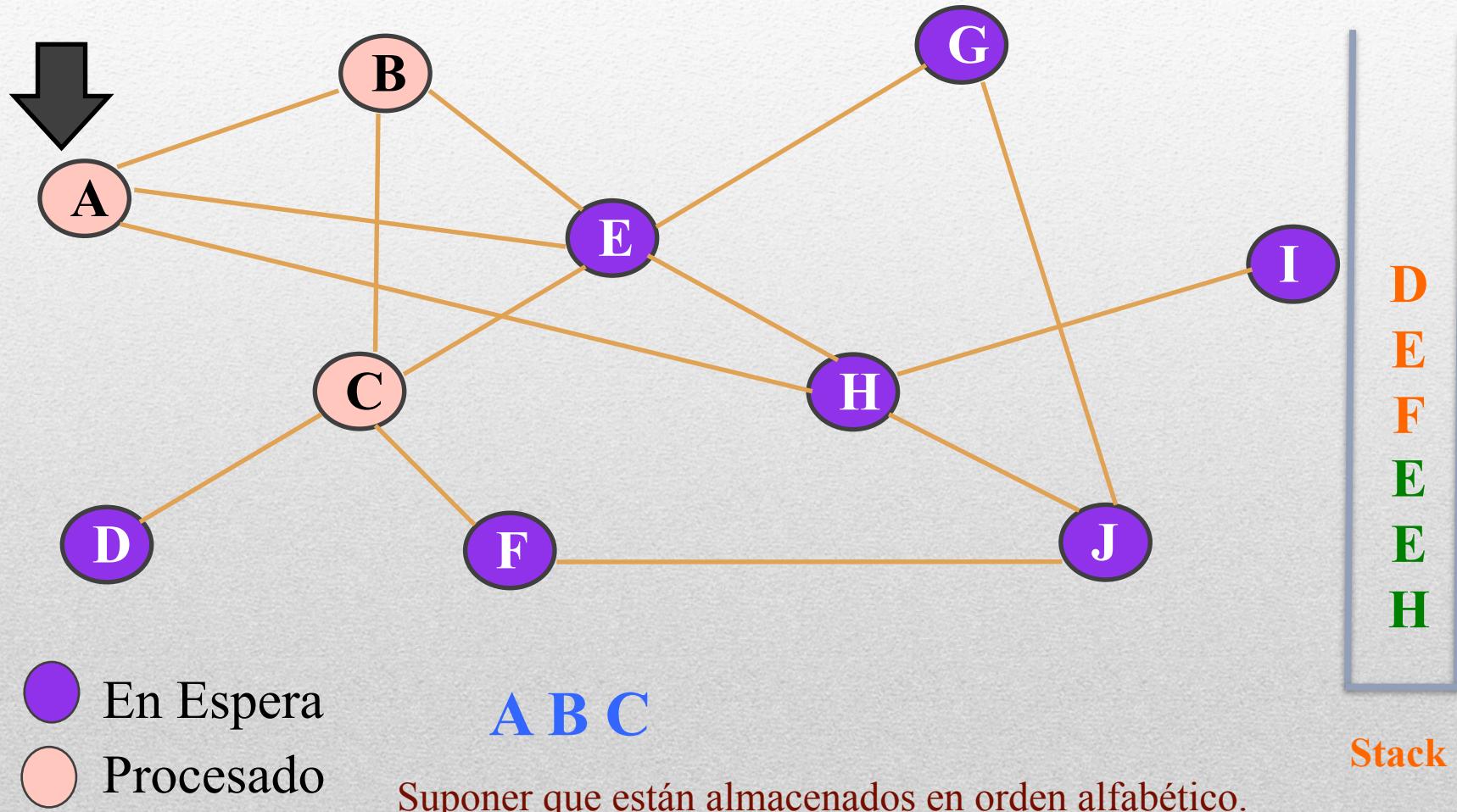
- En Espera
- Procesado

A B

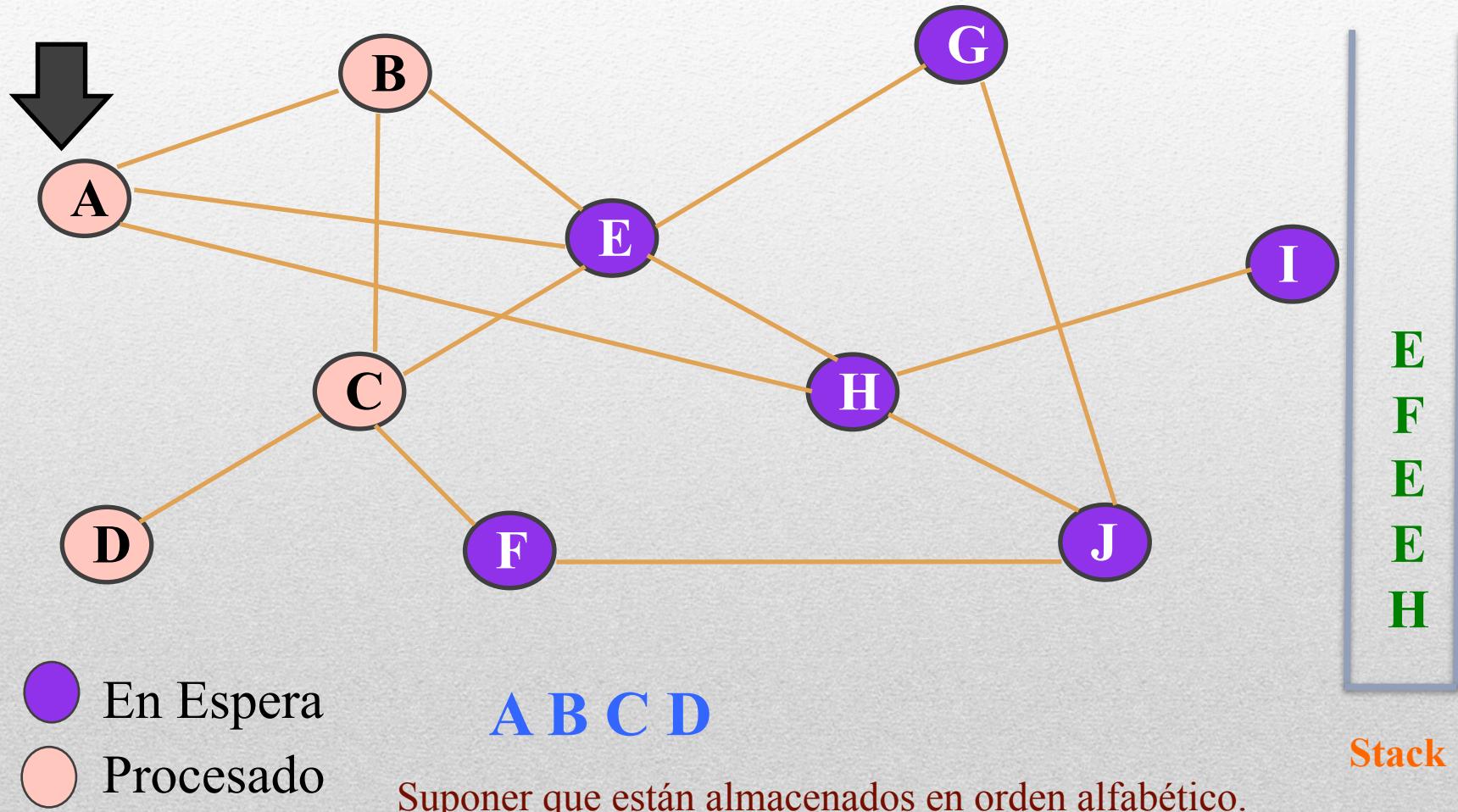
Suponer que están almacenados en orden alfabético.

Stack

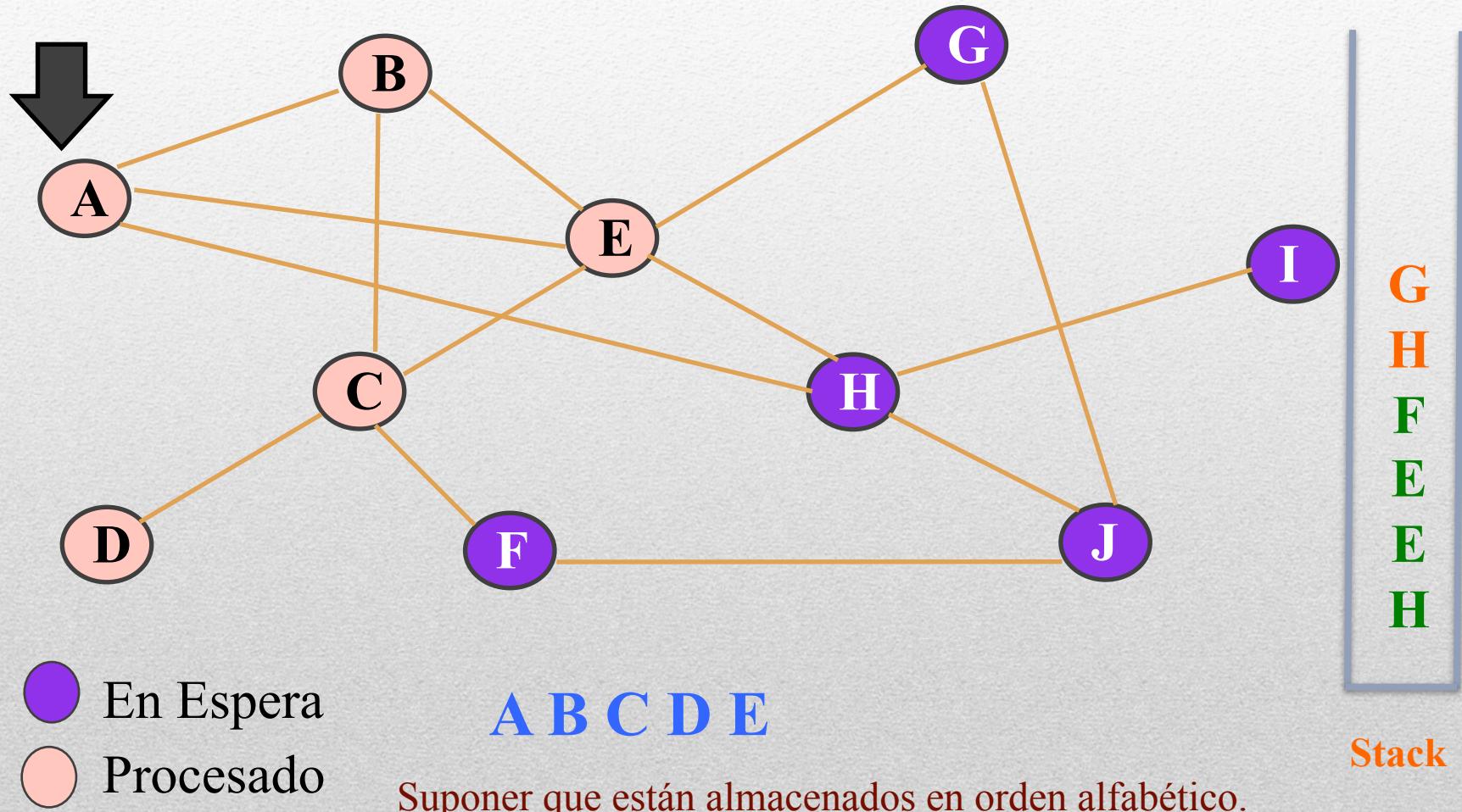
# Ejemplo... Depth First



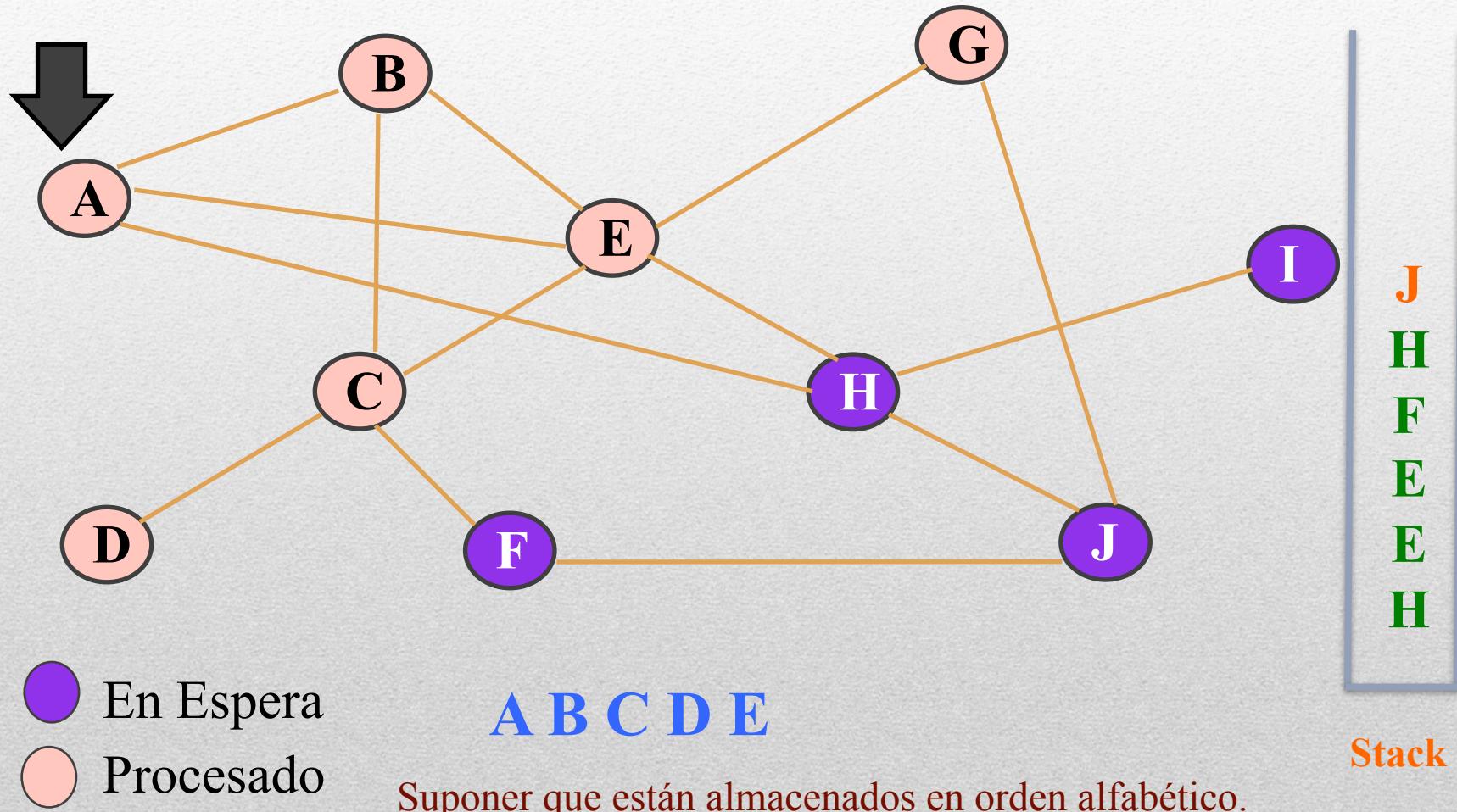
# Ejemplo... Depth First



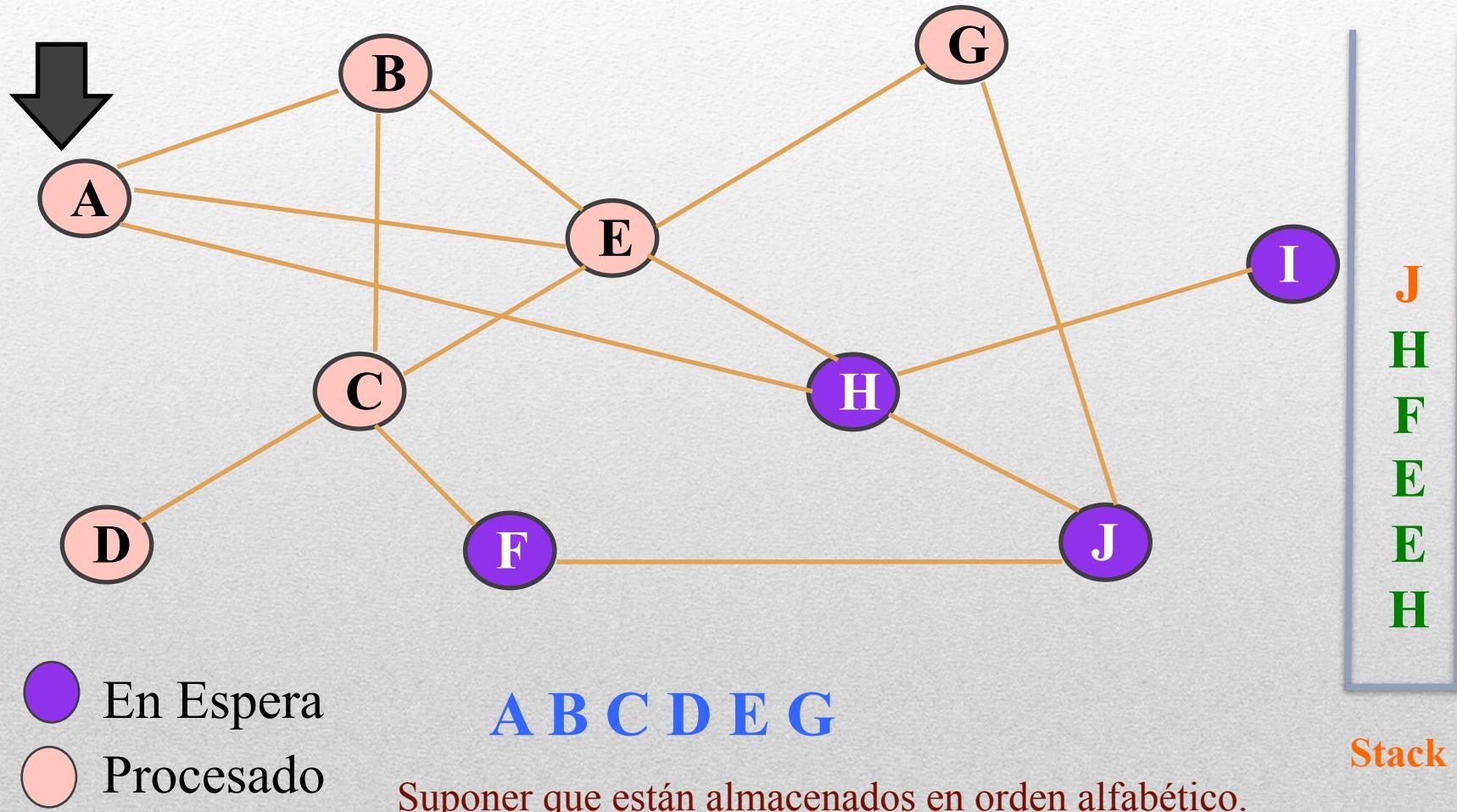
# Ejemplo... Depth First



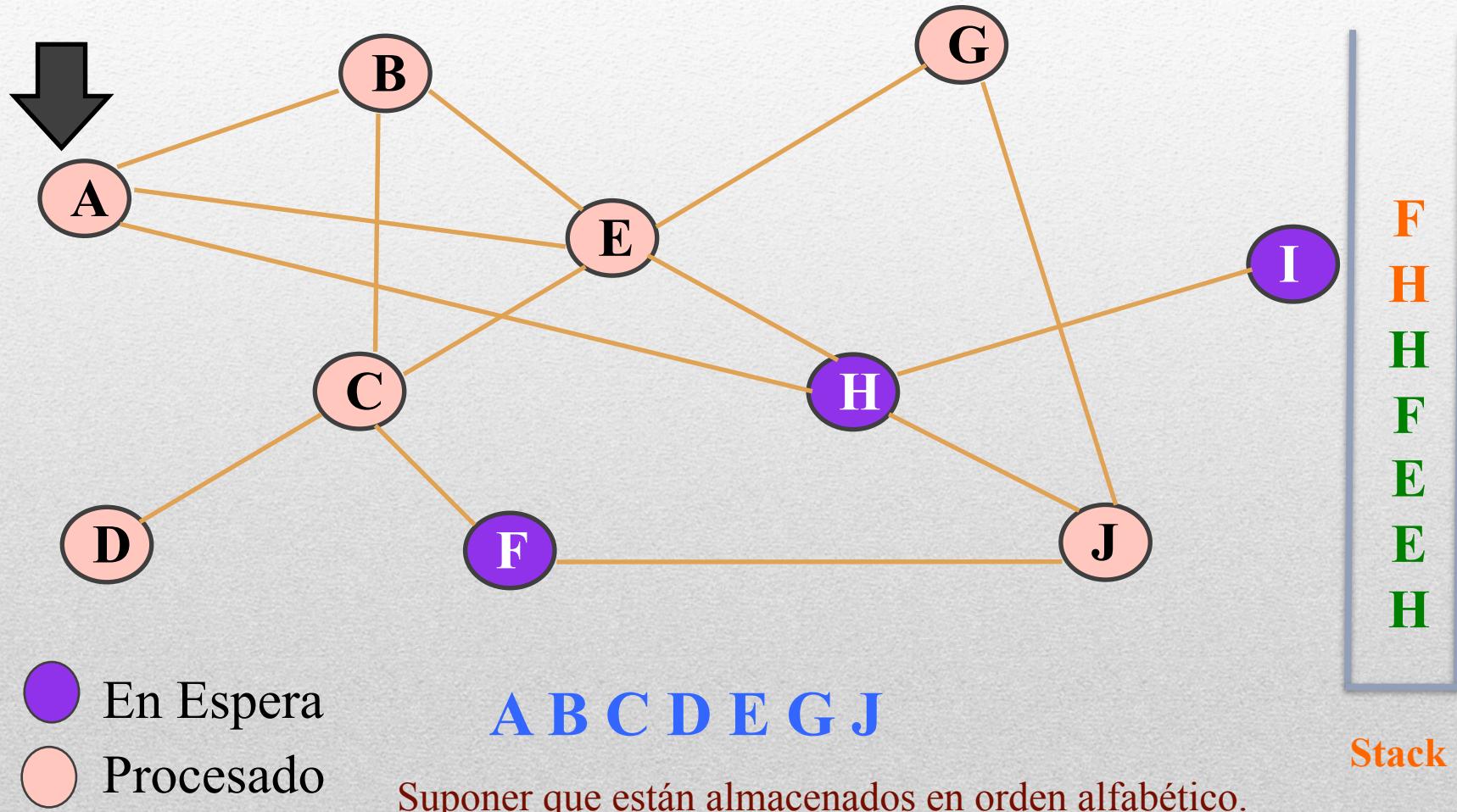
# Ejemplo... Depth First



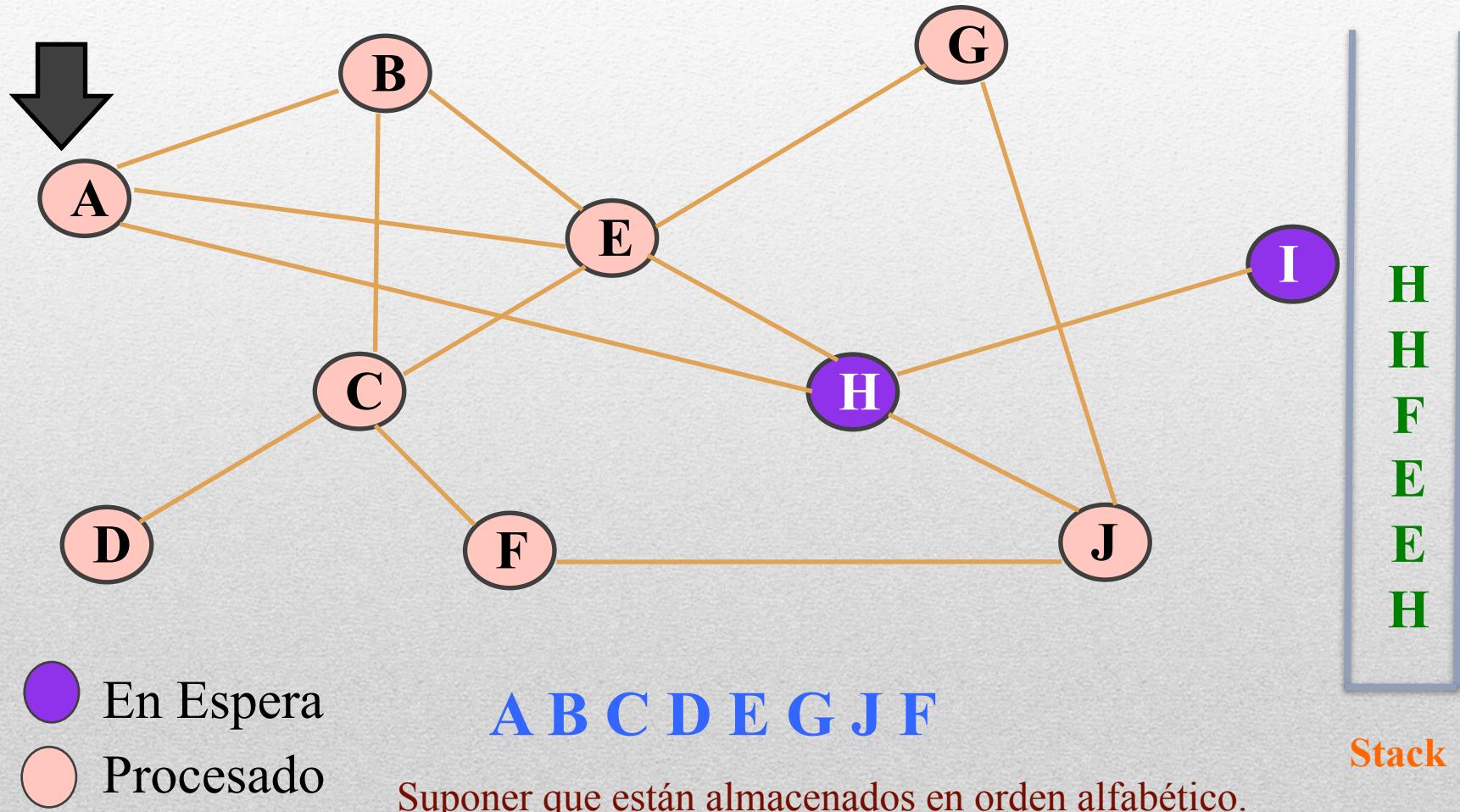
# Ejemplo... Depth First



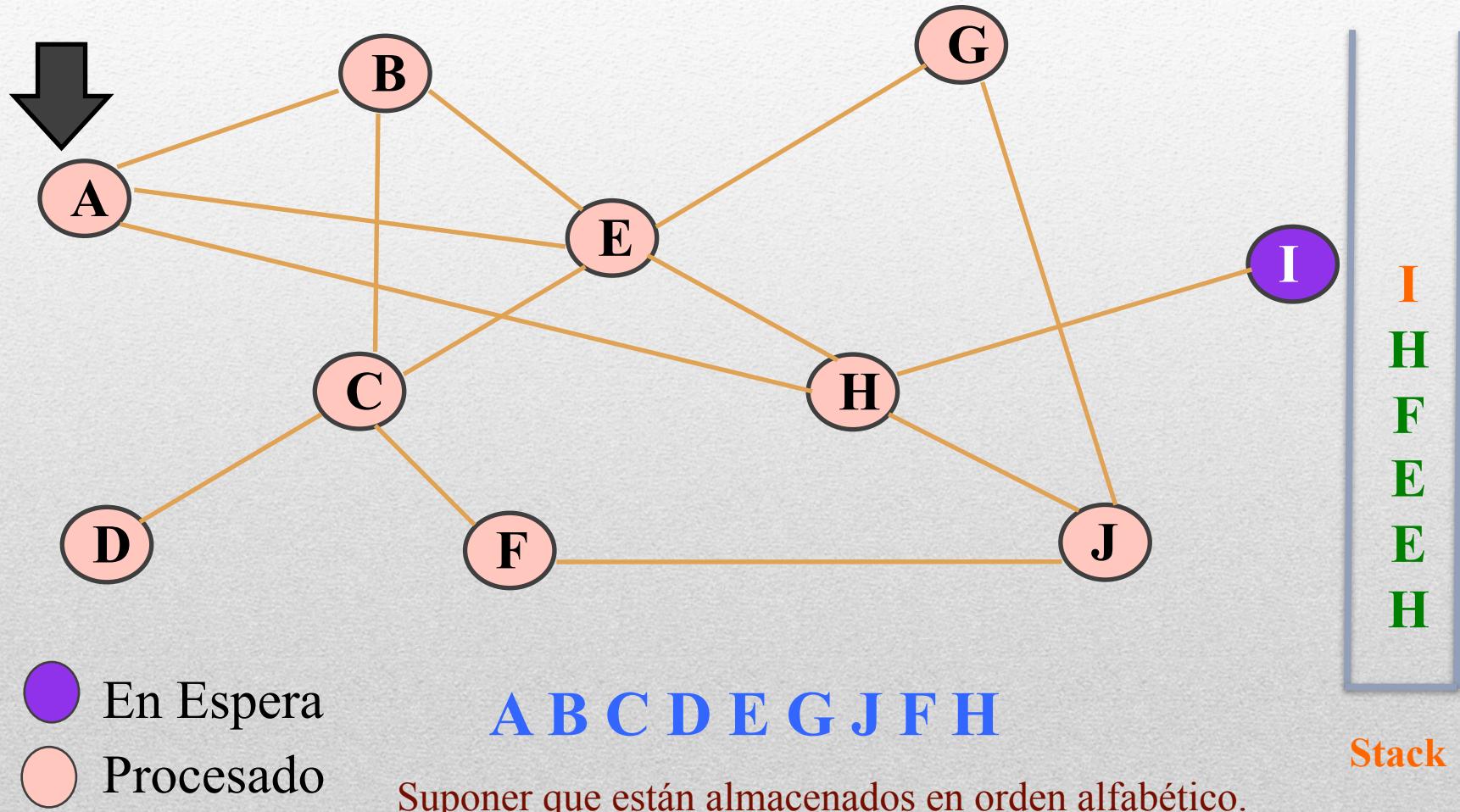
# Ejemplo... Depth First



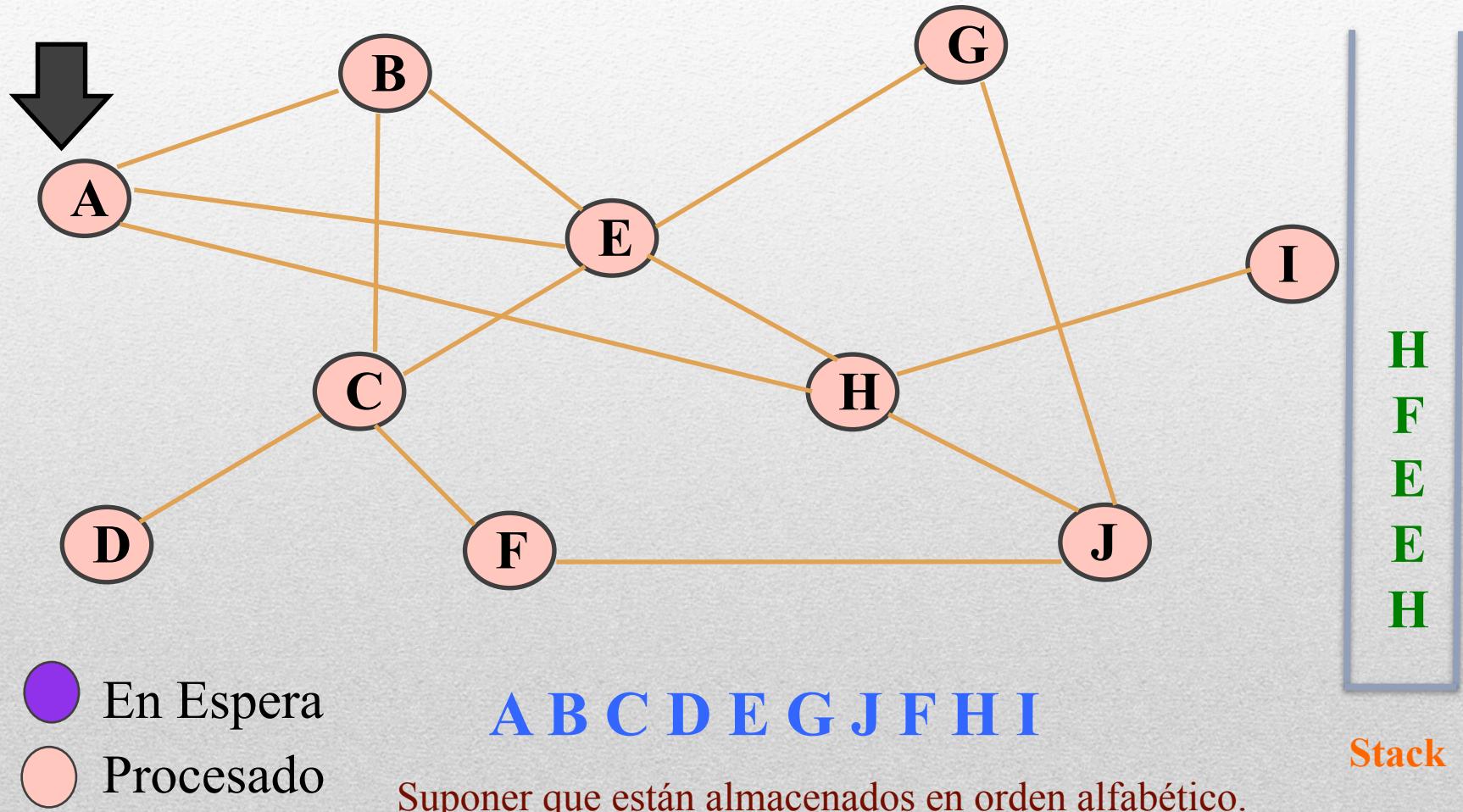
# Ejemplo... Depth First



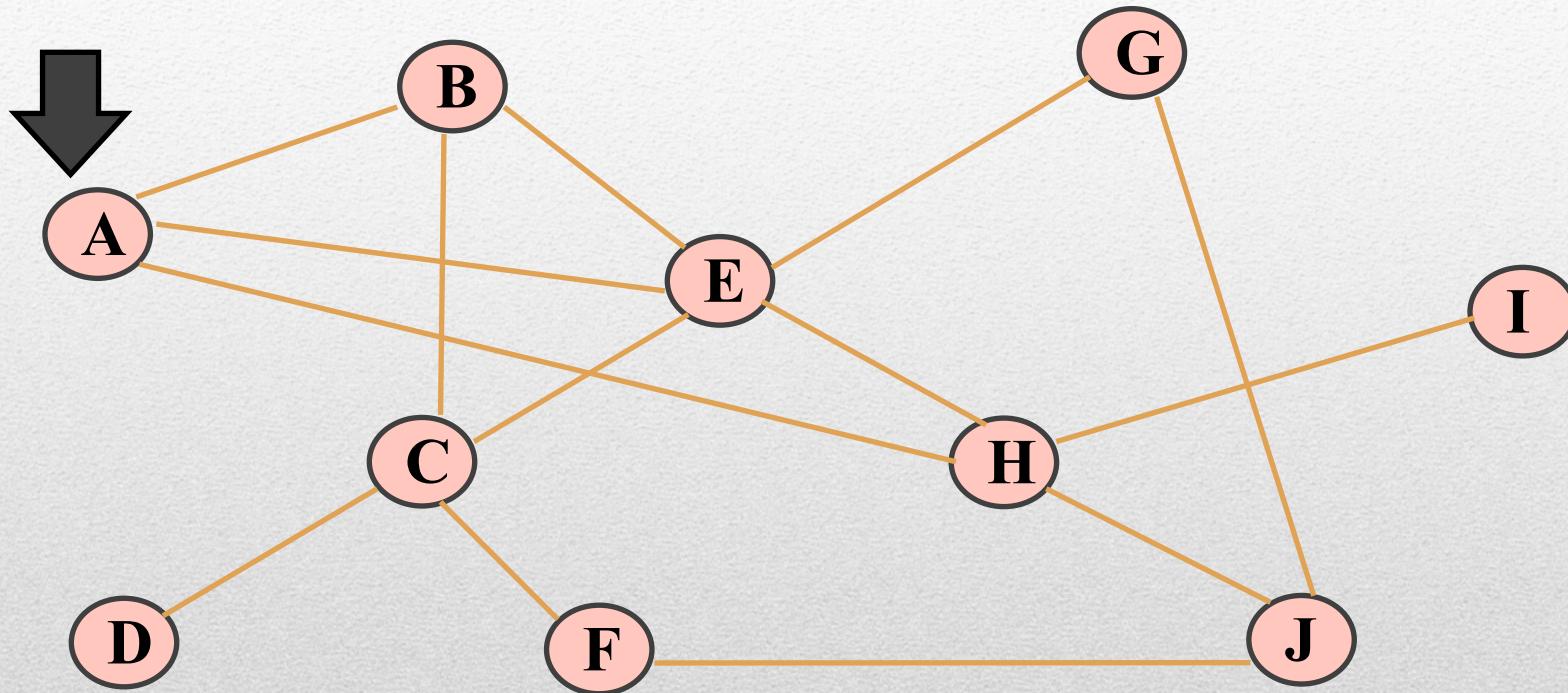
# Ejemplo... Depth First



# Ejemplo... Depth First



# Ejemplo... Depth First



En Espera



Procesado

A **B C D E G J F H I**

Suponer que están almacenados en orden alfabetico.

Stack