

Recorrido de grafos

Recorrido de grafos

- Hay varias técnicas, entre ellas:
 - Búsqueda en anchura (Breadth First Search)
 - Búsqueda en profundidad (Depth First Search)

Búsqueda en anchura

Búsqueda en anchura

- Utiliza colas para encontrar el camino más corto.
- Se van recorriendo los nodos por nivel, recorriendo todos los nodos de un nivel antes de pasar al siguiente nivel.
- Construye el árbol de espacios nivel por nivel.
- Se considera lento comparado con la búsqueda en profundidad.

Búsqueda en anchura: definición y características

- Dado:
 - grafo $G = (V, E)$
 - vértice de inicio s
- La búsqueda primero-en-anchura explora sistemáticamente las aristas de G para “descubrir” cada vértice que sea alcanzable desde s .
- Calcula la distancia (la menor cantidad de aristas) desde s hacia cada vértice alcanzable.
- También produce un “árbol de primera-anchura” con raíz s , que contiene todos los vértices alcanzables.
- Para cualquier vértice v alcanzable desde s , el camino en el árbol de primera-anchura desde s a v corresponde a “el camino más corto” desde s a v en G , es decir, el camino que contiene el menor número de aristas.
- El algoritmo funciona tanto en grafos dirigidos como en grafos no dirigidos.
- Este algoritmo descubre todos los vértices a una distancia k de s , antes de descubrir cualquier vértice a una distancia $k+1$.

Búsqueda en anchura: definición y características

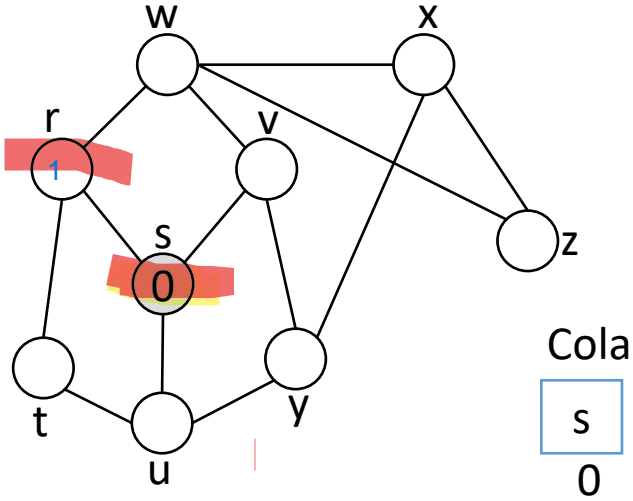
- Para registrar el progreso, la búsqueda primero-en-anchura colorea cada vértice blanco, gris, o negro.
- Todos los vértices empiezan en blanco, luego se vuelven grises y finalmente se vuelven negros.
- Un vértice es “descubierto” la primera vez que se le encuentra durante una búsqueda, momento en el cual se vuelve no-blanco.
- Los vértices grises y negros, han sido descubiertos, pero la búsqueda primero-en-anchura distingue entre ellos para asegurarse de que cada búsqueda procede de una manera primero-en-anchura.

Búsqueda en anchura: definición y características

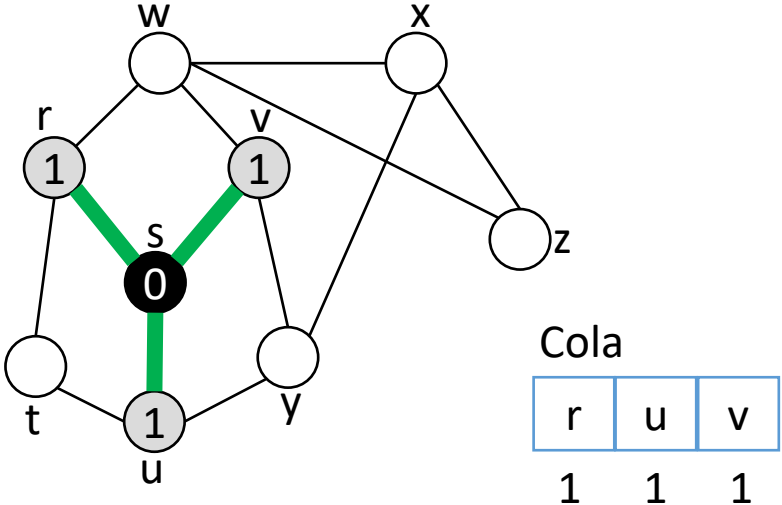
- si (u,v) pertenece a E y el vértice u es negro, entonces v es gris o negro, es decir que todos los vértices adyacentes a un vértice negro han sido descubiertos.
- Los vértices grises pueden tener algunos vértices blancos, estos vértices representan la frontera entre los vértices descubiertos y los no descubiertos.

lo que tiene el nodo +1 y así se obtiene el "peso" de los nodos vecinos

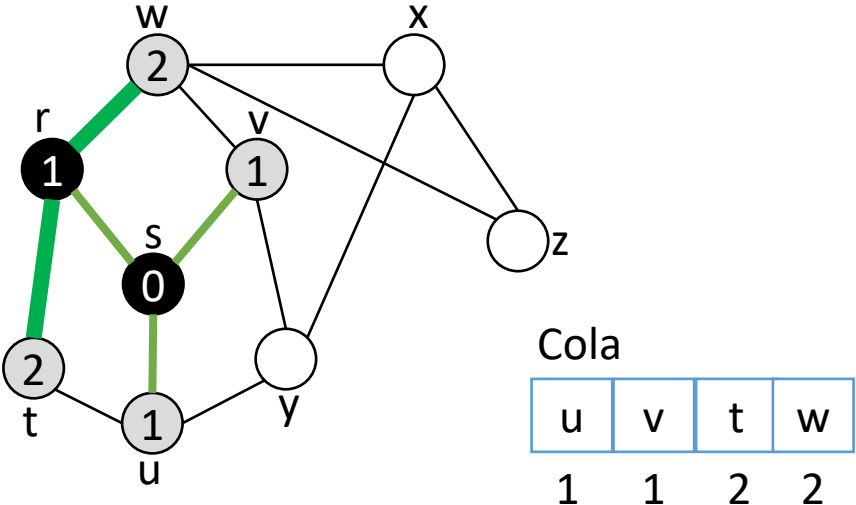
(a)



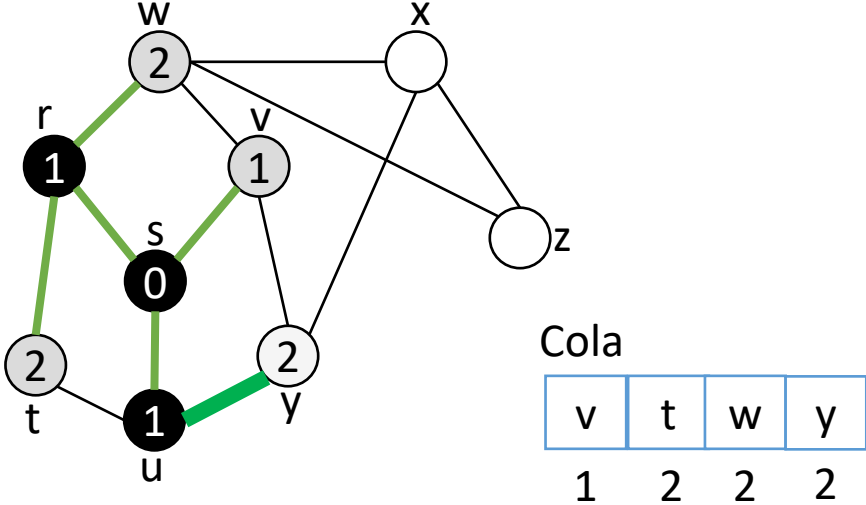
(b)



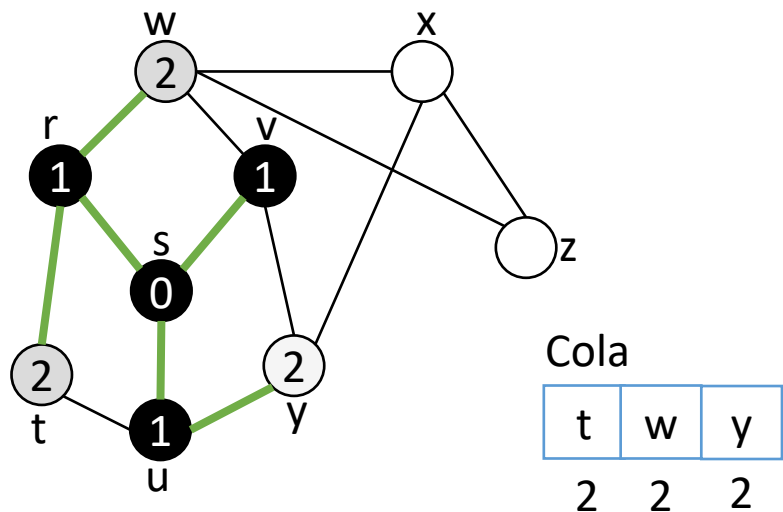
(c)



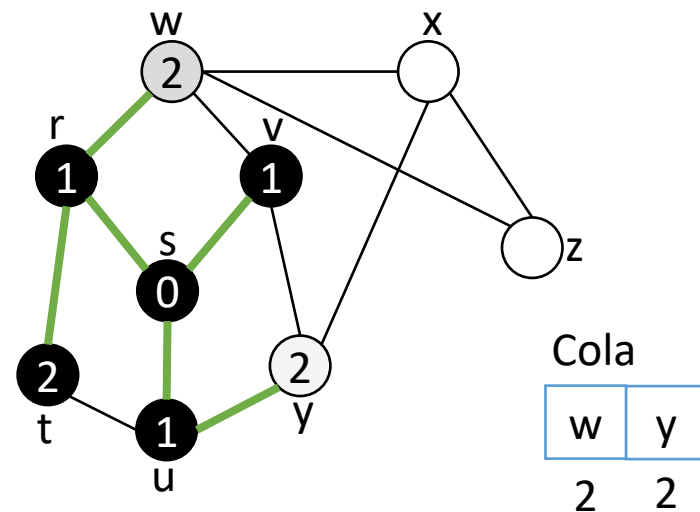
(d)



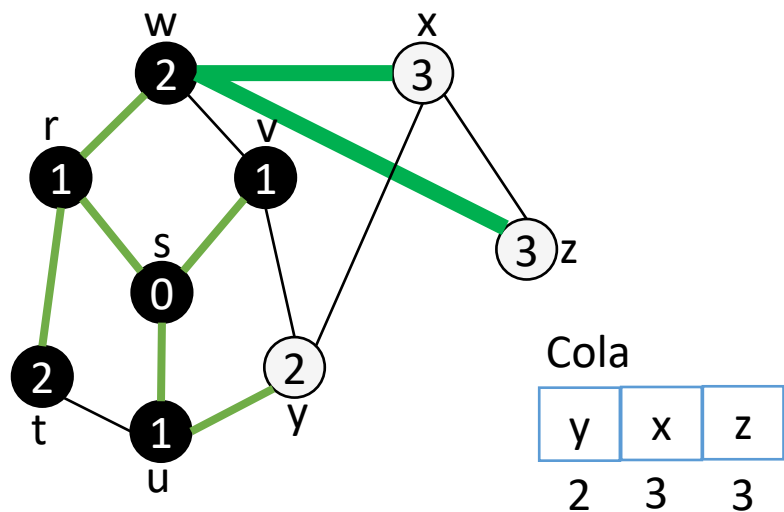
(e)



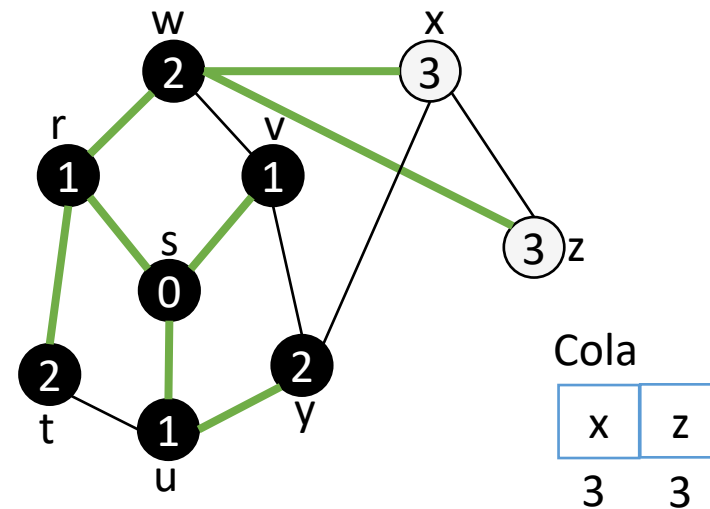
(f)



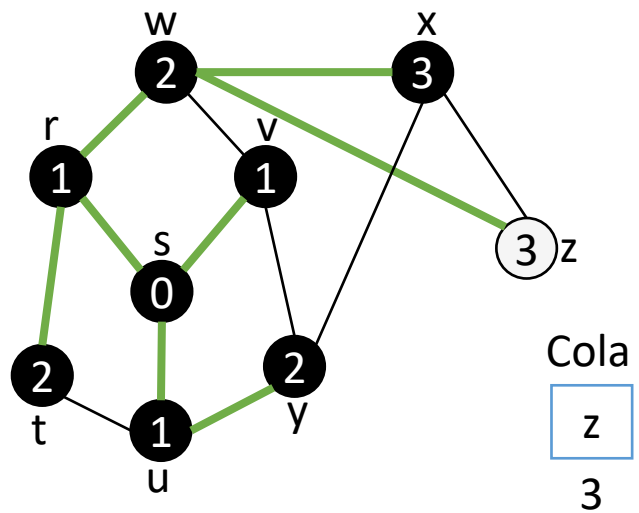
(g)



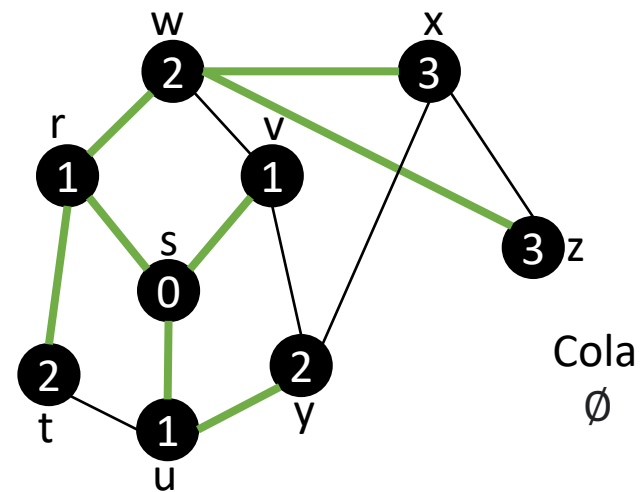
(h)



(i)



(j)



Búsqueda en anchura: aplicaciones

- Búsqueda del camino más corto y el árbol de estados menos costoso para grafos no dirigidos.
- Peer to peer networks (redes par a par) (BitTorrent, para encontrar todos los nodos vecinos).
- Rastreadores web (lo utilizan para construir un índice).
- Redes sociales (para encontrar personas a una distancia K).
- Sistemas de navegación GPS (para encontrar todos los lugares circundantes a un punto específico).
- Transmisión de datos en redes (los paquetes transmitidos siguen este algoritmo).
- Buscar un camino entre dos nodos.

Búsqueda en anchura: aplicaciones

- En árboles de juegos para encontrar la mejor jugada.
- Seguridad de redes (para encontrar todos los dispositivos conectados).

Búsqueda en anchura: ventajas y desventajas

- No se queda atrapado explorando un camino (ciclos).
- Si existe una solución, este algoritmo la encontrará.
- Si existe más de una solución, este algoritmo puede encontrar la que tenga una menor cantidad de pasos.
- Requiere una gran cantidad de memoria.

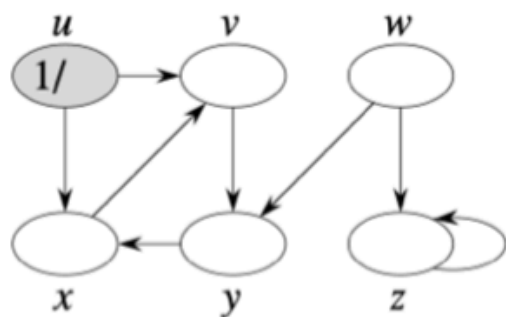
Búsqueda en profundidad

Búsqueda en profundidad

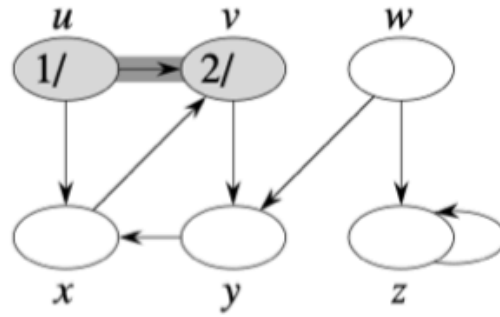
- Utiliza pilas en la búsqueda de la solución.
- El recorrido comienza en el nodo raíz y continúa explorando tan lejos como le sea posible hasta que se llegue a un nodo que no tenga vecinos sin explorar.
- Construye el árbol rama por rama.
- No es el mejor algoritmo para buscar el camino más corto.

Búsqueda en profundidad

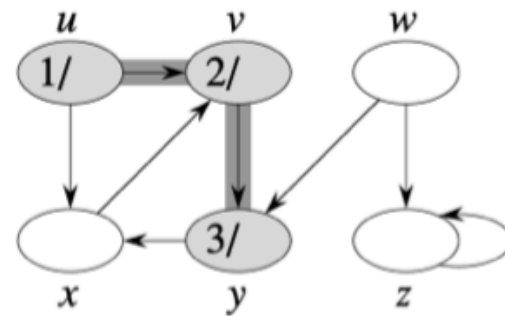
- Se exploran primero las aristas del último vértice v descubierto que aún tenga aristas que salen de él sin explorar.
- Cuando todas las aristas de v han sido exploradas, la búsqueda retrocede para explorar las aristas que salen del vértice a partir del cual v fue descubierto.
- El proceso continúa hasta que se descubren todos los vértices que son alcanzables desde el vértice de origen.
- Si algún vértice continúa sin ser descubierto, entonces uno de ellos es seleccionado como nuevo origen, y la búsqueda es repetida desde ese vértice.
- Todo el proceso se repite hasta que todos los vértices son descubiertos.
- Esta búsqueda genera un conjunto de “árboles de profundidad” disjuntos.



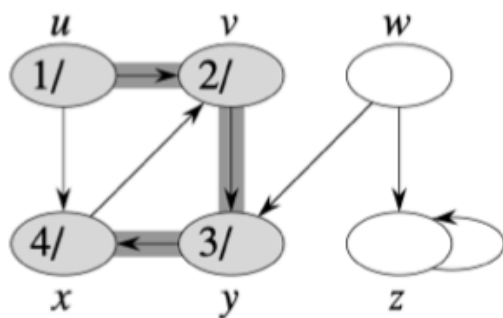
(a)



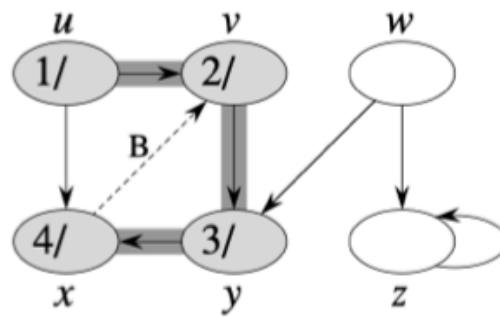
(b)



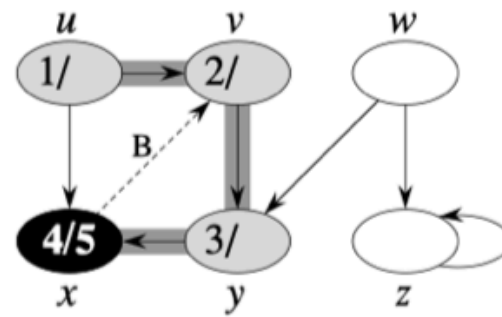
(c)



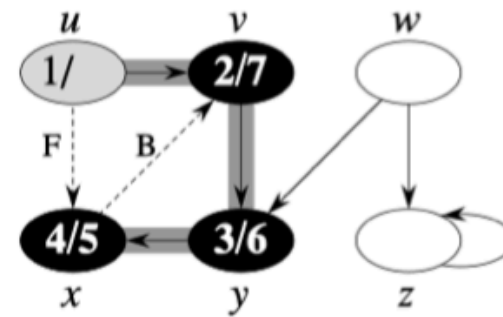
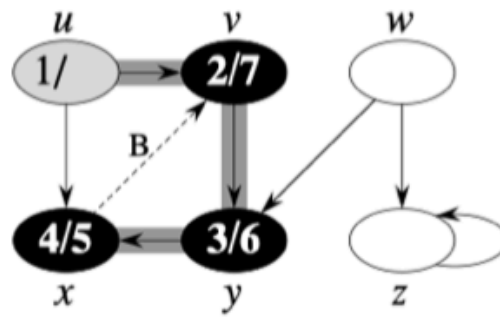
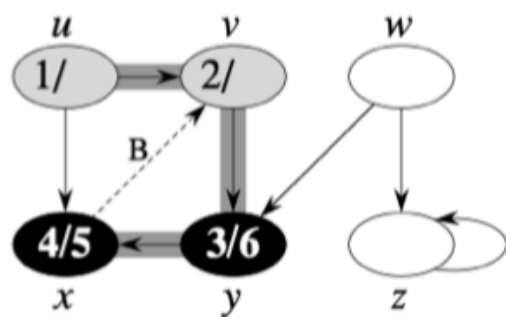
(d)

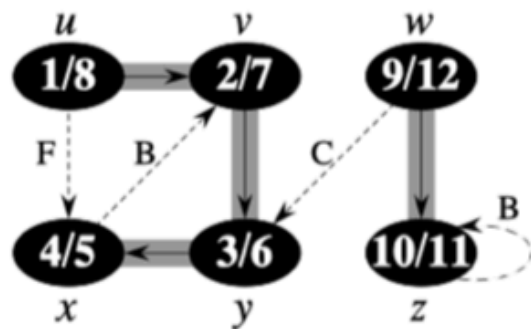
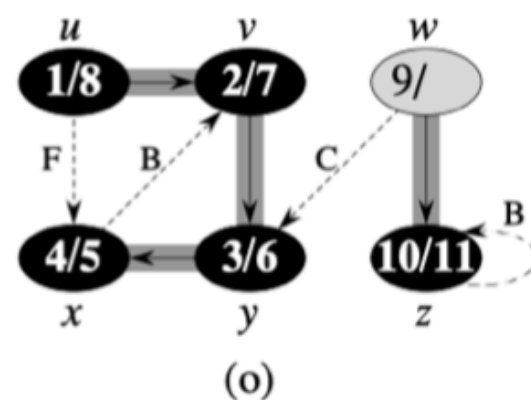
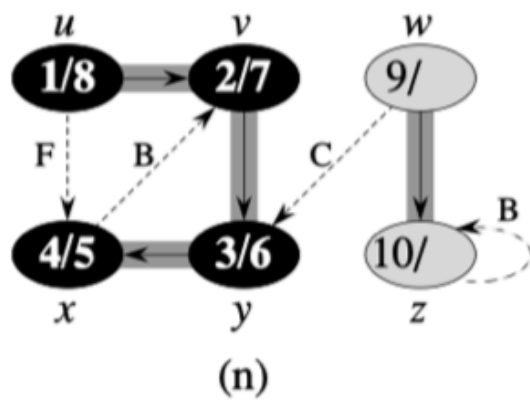
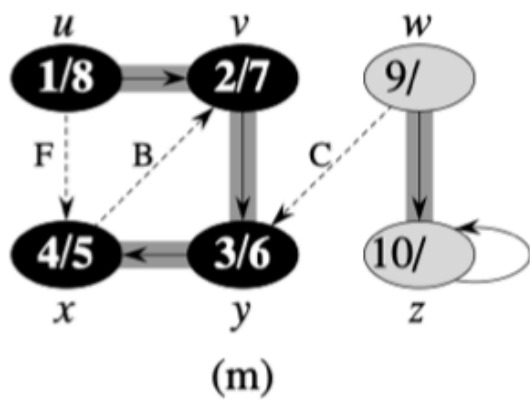
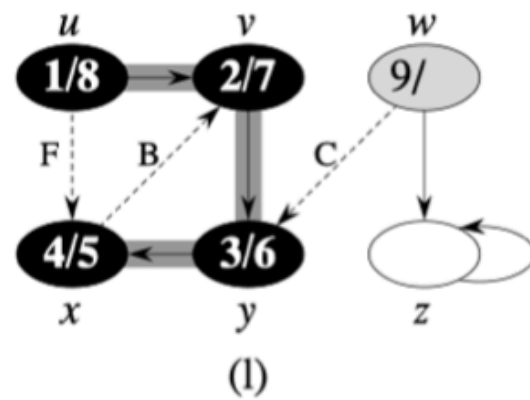
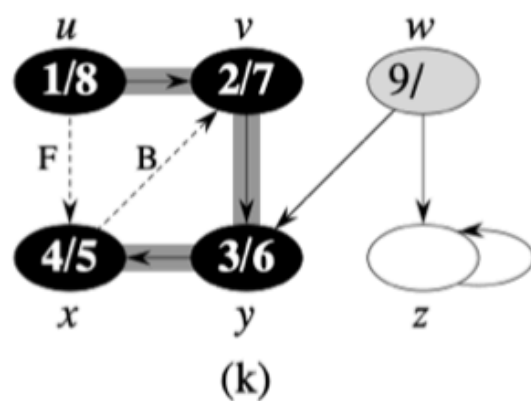
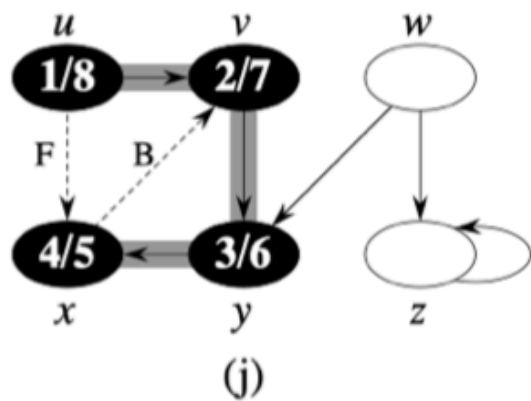


(e)



(f)





Búsqueda en profundidad: aplicaciones

- Búsqueda de caminos entre un par de vértices.
- Resolver acertijos con una única solución (como los laberintos).
- Rastreadores web.
- Generadores de laberintos.
- Checar que un modelo cumpla con un conjunto de criterios.

Búsqueda en profundidad: ventajas y desventajas

- Requiere relativamente poca memoria.
- Permite encontrar soluciones sin explorar demasiado los caminos.
- No garantiza encontrar la solución.
- No garantiza encontrar la solución con menor costo en los casos en los que hay más de una solución.