1. En un grafo, un(a) \_\_\_\_\_\_\_ conecta dos \_\_\_\_\_\_\_\_\_.
2. ¿Cómo se puede saber, observando su matriz de adyacencia, cuántas

aristas hay en un grafo no dirigido?

1. En una simulación de juego que utiliza un grafo, un(a) \_\_\_\_\_\_\_ corresponde a un

estado del tablero de juego y un(a) \_\_\_\_\_\_\_ corresponde al movimiento de un jugador.

1. Un grafo dirigido es aquel en el que

a. se debe seguir el árbol de expansión mínimo.

b. se debe ir entre vértices en orden topológico.

c. se puede ir en una sola dirección desde un vértice dado a otro.

d. se puede ir en una sola dirección en cualquier camino válido.

1. Si la matriz de adyacencia de un grafo tiene filas [0,1,0,0], [1,0,1,1],

[0,1,0,0] y [0,1,1,0], ¿cuál es la lista de adyacencia correspondiente para

los vértices A, B, C y D?

1. Un árbol de expansión mínima es un grafo en el que

a. el número de aristas que conecta todos los vértices es tan pequeño

como sea posible.

b. el número de aristas es igual al número de vértices.

c. se han eliminado todos los vértices innecesarios.

d. cada combinación de dos vértices está conectada por el mínimo

número de aristas.

1. ¿Cuántos árboles de expansión mínima diferentes hay en un

grafo no dirigido de tres vértices y tres aristas?

1. Elija la forma más rápida de verificar si existe una ruta desde el vértice

A al vértice Z en un grafo dirigido entre estas opciones.

a. Obtenga el minimumSpanningTree(A) y luego encuentre la ruta desde

la raíz de ese árbol hasta Z.

b. Recorra los vértices devueltos llamando depthFirst(A) hasta que

aparezca Z.

c. Recorra los vértices devueltos llamando breadthFirst(Z) hasta que

aparezca A.

d. Aplique el algoritmo de Warshall al gráfico y luego verifique la

matriz de conectividad para ver si A puede alcanzar Z.

1. Un gráfico no dirigido debe tener un ciclo si

a. se puede alcanzar cualquier vértice desde algún otro vértice.

b. el número de componentes conectados es mayor que uno.

c. el número de aristas es igual al número de vértices.

d. el número de rutas es menor que el número de aristas.

1. Un(a) \_\_\_\_\_\_ es un gráfico sin ciclos.
2. El grado de un vértice

a. es el número de aristas en la ruta que lo une a un vértice inicial.

b. es el número de aristas que lo conectan a otros vértices.

c. es el número de vértices en su componente conectado del

grafo.

d. es la mitad del número de aristas en su fila de la matriz de adyacencia.

1. ¿Puede un árbol de expansión mínimo para un grafo no dirigido tener

ciclos?

1. Verdadero o falso: puede haber muchas ordenaciones topológicas correctas para un

grafo dirigido dado.

1. La ordenación topológica da como resultado

a. aristas dirigidas, de modo que los vértices están en orden ascendente.

b. vértices enumerados en orden de número creciente de aristas desde el

vértice inicial.

c. vértices dispuestos en orden ascendente, de modo que F precede a G, que

precede a H, y así sucesivamente.

d. vértices enumerados de modo que los que están más adelante en la lista están aguas abajo de los que están antes.

1. Si la matriz de adyacencia de un grafo tiene filas [0,1,0,0], [0,0,0,1],

[1,0,0,0] y [1,0,0,0] con vértices A, B, C y D, ¿podría

pasarse como argumento a sortVertsTopologically()? Si es así, ¿cuál

sería el resultado?

1. ¿Qué es un DAG?
2. El algoritmo de Warshall

a. encuentra el ciclo más grande en un grafo, si lo hay.

b. cambia la matriz de adyacencia a una matriz de conectividad.

c. ordena los vértices en orden ascendente pero no topológicamente.

d. encuentra la menor cantidad de aristas necesarias para realizar el cierre.

1. ¿Bajo qué condiciones tiene sentido realizar una ordenación topológica en un grafo no dirigido?
2. Si el grafo G1 tiene 100 vértices y G2 tiene 10 vértices, ¿cuál es la

razón de tiempo de cálculo entre la llamada a sortVertsTopologically() en

G1 y G2?

1. ¿Qué algoritmo resuelve el problema de los puentes de Königsberg?

a. El algoritmo de Warshall

b. El algoritmo de árbol de expansión mínimo

c. El algoritmo de ordenación topológica

d. El algoritmo de recorrido en amplitud