

Trabajo práctico sobre Grafos

Instrucciones cumplir para que su práctico sea corregido:

- Crear un proyecto java usando Maven
 - En dicho proyecto implementar métodos como respuesta a cada uno de los siguientes ejercicios
 - El proyecto se debe almacenar en un repositorio en github (Para esto puede usar una cuenta que ya disponga o crearse una)
 - Incluir un programa en consola que haga la prueba de los métodos resueltos indicando mediante mensajes a que pregunta es respuesta cada método probado. De no incluir este punto en su práctico no se realizará la revisión.
 - En la plataforma copiar el enlace para descargarse el repositorio. Asegurarse que su repositorio tiene acceso de lectura al público. En caso de que su repositorio no se pueda descargar al momento de la revisión no se tomará en cuenta su trabajo.
-
1. Para un grafo no dirigido implementar los métodos insertarVertice, insertarArista, eliminarVertice, eliminarArista, cantidadDeVertices, cantidadDeArista, gradoDeVertice
 2. Para un grafo no dirigido implementar los métodos insertarVertice, insertarArista, eliminarVertice, eliminarArista, cantidadDeVertices, cantidadDeArista, gradoDeEntradaVertice, gradoDeSalidaDeVertice
 3. Para un grafo dirigido implementar método o clase para encontrar si hay ciclos sin usar matriz de caminos.
 4. Para un grafo dirigido implementar método o clase para encontrar si hay ciclos usando la matriz de caminos
 5. Para un grafo dirigido implementar un método o clase que sea capaz de retornar los componentes de las islas que existen en dicho digrafo.
 6. Para un grafo dirigido implementar un método o clase para encontrar la matriz de caminos de dicho grafo dirigido
 7. Para un grafo dirigido implementar un método o clase que permita determinar si el digrafo es débilmente conexo
 8. Para un grafo dirigido implementar un método o clase que permita determinar si el digrafo es fuertemente conexo
 9. Para un grafo no dirigido implementar un método o clase que permita encontrar si en dicho grafo hay ciclo.
 10. Para un grafo no dirigido implementar método o clase para encontrar los componentes de las diferentes islas que hay en dicho grafo
 11. Para un grafo dirigido implementar un algoritmo para encontrar el número de islas que hay en el grafo
 12. Para un grafo dirigido implementar el algoritmo de Wharsall, que luego muestre entre que vértices hay camino.
 13. Para un grafo dirigido usando la implementación del algoritmo de Floyd-Wharsall encontrar los caminos de costo mínimo entre un vértice v y el resto. Mostrar los costos y cuáles son los caminos
 14. Para un grafo dirigido implementar un algoritmo que retorne cuántas componentes fuertemente conexas hay en dicho grafo. Definimos formalmente un componente fuertemente conectado, C , de un grafo G , como el mayor subconjunto de vértices C (que es un subconjunto de los vértices del grafo G) tal que para cada pareja de vértices v, w pertenecen a C tenemos una ruta desde v hasta w y una ruta desde w hasta v .

15. Para un grafo dirigido pesado implementar el algoritmo de Dijkstra que muestre con que vértices hay caminos de costo mínimo partiendo desde un vértice v , con qué costo y cuáles son los caminos.
16. Para un grafo no dirigido pesado implementar el algoritmo de Kruskal que muestre cual es el grafo encontrado por el algoritmo
17. Para un grafo no dirigido pesado implementar el algoritmo de Prim que muestre cual es el grafo encontrado por el algoritmo
18. Para un grafo dirigido implementar al algoritmo de ordenamiento topológico. Debe mostrar cual es el orden de los vértices según este algoritmo.
19. Para un grafo dirigido pesado implementar el algoritmo de Ford-Fulkerson