Red de transporte

ESTRUCTURA DATOS Y ALGORITMOS II

LUCÍA MIELGO Y MARTINA GARCÍA

En este documento, explicamos brevemente las funciones de cada archivo de nuestro proyecto para facilitar su comprensión. Hemos creado una clase grafo que recoge las ciudades y sus conexiones, e implementado el algoritmo de Dijkstra para encontrar la ruta más corta entre dos ciudades. Además, utilizamos un árbol binario para organizar las distancias entre ciudades Por último, hemos programado un árbol de recubrimiento mínimo para analizar la estructura de conexión entre las ciudades.

CLASE GRAFO

Clase nodo:

-Cada nodo representa una ciudad con sus atributos nombre y una lista de caminos desde esta.

- Nodo_previo, guarda el nodo anterior en el camino para utilizarlo en la ruta más corta.

-Insertar_id:se utiliza para construir la matriz de adyacencia.Este asigna un id a cada nodo.

Clase arista:

-Representa el camino entre dos ciudades(nodos).

- __repr__ sirve para representar e imprimir estas conexiones.

Clase grafo:

-Atributos principales:número de nodos y nodos.

-insertar_nodo:sirve para añadir un nuevo nodo al grafo.

-insertar_arista:crea una nueva arista y la añade a los caminos de los nodos.

-mostrar_grafo:imprime la matriz de adyacencia que representa las distancias y conexiones entre las ciudades.

mostrar_distancias_ordenadas:utiliza el árbol binario para mostrar las distancias y conexiones entre ciudades de menor a mayor distancia.

BST

-**Clase NodoArbol**:cada nodo guarda una distancia y una lista de ciudades asociadas a estas últimas.Por otro lado,estos apuntan a sus dos nodos hijos,derecha e izquierda.

-**Clase Arbol**:árbol binario que utilizaremos para recoger las distancias y almacenarlas de forma ordenada.Dentro de esta clase, tenemos el método insertar para añadir un nuevo nodo.Asimismo, utilizamos la función insertar_recur que se trata de un método recursivo para insertar el nodo en el árbol en función de la distancia.Por último tenemos el método in order,que imprime la la distancia y las ciudades de cada nodo siguiendo el orden árbol izquierdo,raíz y árbol derecha.

RUTA MÁS CORTA

Para encontrar la ruta más corta utilizamos algoritmo Dijkstra.

- **elegir_nodo(origen)**:esta función selecciona el nodo vecino con la menor distancia desde el nodo origen.De tal manera que recorre todos los caminos posibles desde el origen y elige el nodo que tenga la distancia más corta al final del camino.

-**ruta_mas_corta(origen,destino,grafo)**:inicializamos todos los nodos del árbol con el estado de visitado=False. Establecemos la distancia del nodo origen a 0, que se trata de la ciudad desde la que salimos. Por otro lado,el bucle while se ejecuta hasta que el nodo actual sea igual al destino.Así pues ,para cada camino que sale del nodo actual,se calcula la nueva distancia sumando la distancia del camino y la acumulada hasta el nodo.Si esta distancia es menor que la distancia acumulada al final del camino se actualiza esta y se modifica el nodo actual.Al terminar el bucle, se devolverá una tupla que contiene la distancia desde el nodo origen hasta el destino y la lista de nodos que forman la ruta más corta.

MST

El objetivo del MST es conectar todos los nodos del grafo con la menor distancia total posible, sin formar ciclos

calcular_mst(grafo: Grafo).

En primer lugar se inicializan las variables :

-visitados: lista para llevar un registro de los nodos que han sido visitados.
-aristas_mst: lista para almacenar las aristas que forman parte del MST.
-nodo_actual: el primer nodo del grafo.
-distancia_total: almacena la distancia total del MST.

-Bucle:
El bucle while se ejecuta hasta que todos los nodos (excepto uno) estén en la lista visitados.En cada vuelta se busca la arista con la menor distancia que conecta un nodo visitado con uno no visitado.Así pues,la arista encontrada se añade a aristas_mst, y su distancia se suma a distancia total.

Esta función nos devuelve un diccionario con la distancia total del mst y las aristas .Estas últimas se representan como tuplas que guardan los nombres de los nodos conectados y la distancia entre estos.