Parcial 2 Estructura de Datos II

Sebastian J. Racedo V.

March 23, 2023

1 Punto 1 2.0

Hay n ciudades numeradas del 0 al $n-1$. Dado el array **edge** donde $edge[i] = [from_i, to_i, weight_i]$ representa un borde bidireccional y ponderado entre las ciudades $from_i$ y to_i, y dado el entero **MaxDistance**.

Devuelva la ciudad con el menor número de ciudades a las que se pueda llegar a través de alguna ruta y cuya distancia es a lo mas **MaxDistance**. Si hay varias ciudades de este tipo, devuelva la ciudad con el menor número. Observe que la distancia de un camino que conecta las ciudades i y j es igual a la suma de los pesos de los bordes a lo largo de ese camino.

1.1 Example 1:

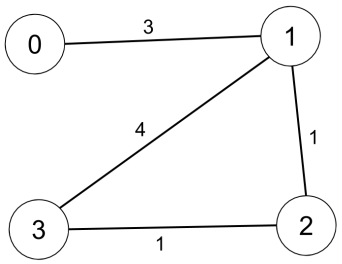

Figure 1: Example 1.

Entrada: n = 4, edges = [[0,1,3],[1,2,1],[1,3,4],[2,3,1]], MaxDistance = 4
Salida esperada: 0
Explicación: La figura anterior describe el gráfico. Las ciudades vecinas a una MaxDistance = 4 para cada ciudad son:

$$City0 \rightarrow [City1, City2]$$
$$City1 \rightarrow [City0, City2, City3]$$
$$City2 \rightarrow [City0, City1, City3]$$
$$City3 \rightarrow [City1, City2]$$

Las ciudades 0 y 3 tienen 2 ciudades vecinas a una MaxDistance = 4, pero tenemos que devolver la ciudad 0 ya que tiene el menor número.

1.2 Example 2:

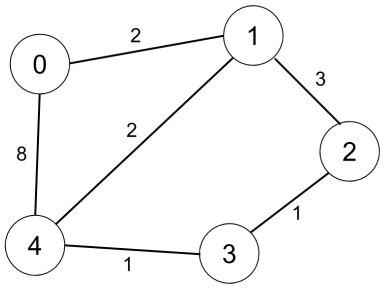

Figure 2: Example 2.

Entrada: n = 5, edges = [[0,1,2],[0,4,8],[1,2,3],[1,4,2],[2,3,1],[3,4,1]], MaxDistance = 2
Salida esperada: 0
Explicación: La figura anterior describe el gráfico. Las ciudades vecinas a una MaxDistance = 2 para cada ciudad son:

$$City0 \rightarrow [City1]$$

$$City1 \rightarrow [City0, City4]$$

$$City2 \rightarrow [City3, City4]$$

$$City3 \rightarrow [City2, City4]$$

$$City4 \rightarrow [City1, City2, City3]$$

La ciudad 0 tiene 1 ciudad vecina a una MaxDistance = 2.

2 Punto 2 2.0

Dado el siguiente grafo encuentre el árbol de expansión mínima.

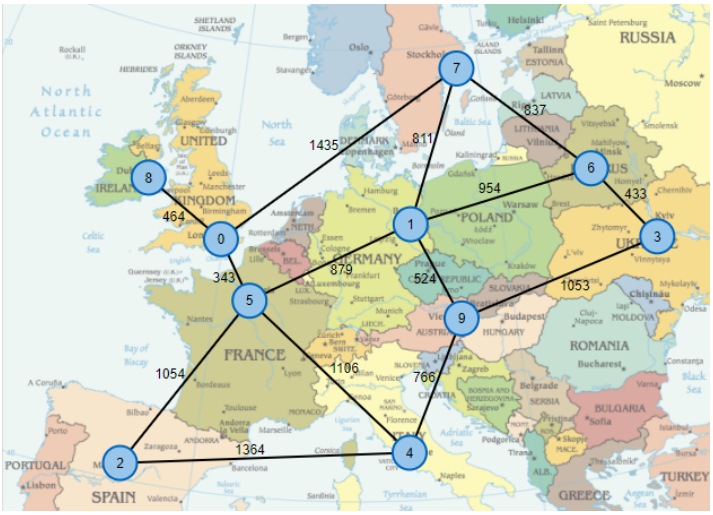

Figure 3: Punto 2.

3 Punto 3 1.0

Usando el grafo del segundo punto, debe crear direcciones entre los posibles destinos (usted decide como direccionar cada par de destinos). Una vez lo haga debe codificar (en java o python) los procedimientos/funciones necesarias para decir si este es fuertemente o débilmente conexo.

4 Notas:

La solución (análisis) debe ser original.
Cualquier intento de fraude y/o plagio será castigado de acuerdo al reglamento estudiantil.
NO se recibirán soluciones después de la hora límite de entrega.
Puede usar , Python, Java o Pseudocodigo para resolver este parcial.
Debe documentar todo su código (la documentación debe ser de autoría propia), esto será evaluado!
Para mandar su solución se espera un comprimido que tenga la forma usualmente usada Recuerde que es .ZIP