Proyecto\_3\_201910\_sec\_3\_team\_7

# Listado de Requerimientos

|  |  |
| --- | --- |
| Nombre | 0 -Cargar el Grafo No Dirigido de la malla vial de la ciudad completa de Washington D.C. creado en el taller 8. |
| Resumen | Cargar el Grafo No Dirigido (grafo más grande) de la malla vial de la ciudad completa de Washington D.C. (formato JSON), creado en el taller 8. Informar el total de vértices y el total de arcos que definen el grafo cargado. Solo es permitido leer una vez la información de los archivos. |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Se carga la malla vial en un grafo no dirigido. | |
| Complejidad | |
| La complejidad es O(N2) en el peor de los casos en el cual hay que hacer rehash para poder seguir agregando vértices al grafo. | |

|  |  |
| --- | --- |
| Nombre | 1 –Agregar al grafo la información de las infracciones de todos los meses del año. |
| Resumen | Al grafo creado, se debe agregar la información de cada una de las infracciones de todos los meses del año (nuevos archivos \*.csv). Para este fin, ubique el vértice más cercano a la ubicación geográfica de la infracción y sobre este vértice almacene la información de la infracción que considere relevante. Utilice la distancia harvesiana(en kilómetros) entre dos ubicaciones geográficas para calcular la distancia entre la ubicación de un vértice del grafo y la ubicación de una infracción. Informar el total de infracciones cargadas y el número de las infracciones de cada mes. |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Se agrega al grafo la información de las infracciones. | |
| Complejidad | |
| La complejidad es de O(N\*M) N es el número de vértices del grafo y M es el número de infracciones. | |
|  | |
| Nombre | 1A - Encontrar el camino de costo mínimo (menor cantidad de infracciones en la ruta) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Resumen | Encontrar el camino de costo mínimo (menor cantidad de infracciones en la ruta) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo.  Respuesta en consola: Muestre en la consola de texto el camino a seguir, informando sus vértices (Id, Ubicación Geográfica), el costo mínimo (menor cantidad de infracciones), y la distancia estimada (en Km). Visualización mapa: Muestre el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Se muestra en consola el camino a seguir, informando sus vértices (Id, Ubicación Geográfica), el costo mínimo (menor cantidad de infracciones), y la distancia estimada (en Km); además, se muestra el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). | |
| Complejidad | |
| La complejidad es de O(2V \* E) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

|  |  |
| --- | --- |
| Nombre | 2A - Determinar los *n* vértices con mayor número de infracciones en la ciudad de Washington D.C. Adicionalmente identificar las componentes conectadas que se definan únicamente entre estos *n* vértices. |
| Resumen | Determinar los *n* vértices con mayor número de infracciones en la ciudad de Washington D.C. Adicionalmente identificar las componentes conectadas (subgrafos) que se definan únicamente entre estos *n* vértices. Las componentes conectadas (subgrafos) solo pueden usar arcos del grafo original que conecten estos *n* vértices.  **Respuesta en consola:** Mostrar los *n* vértices resultantes en la consola de texto (su identificador, su ubicación (latitud, longitud), y el total de infracciones) ordenados de mayor a menor por el número de infracciones. Informar el número de componentes conectadas (subgrafos) que se definen entre estos vértices en el grafo original. Por cada componente informar los identificadores de los vértices que la componen.  **Visualización mapa**: marque la localización de los vértices resultantes en un mapa en Google Maps usando un color 1. Destaque la componente conectada más grande (con más vértices) usando un color 2. Para esta componente muestre sus vértices y sus arcos. |
| Entradas | |
| Los n vértices(int). | |
| Resultados | |
| Se muestra por consola los n vértices con su identificador, su ubicación (latitud, longitud), y el total de infracciones ordenados de mayor a menor por el número de infracciones; además, se marca la localización de los vértices resultantes en un mapa en Google Maps usando un color 1. Destaque la componente conectada más grande (con más vértices) usando un color 2. Para esta componente muestre sus vértices y sus arcos. | |
| Complejidad | |
| La complejidad es de O(2V \* E) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

|  |  |
| --- | --- |
| Nombre | 1B - Encontrar el camino más corto para un viaje entre dos ubicaciones geográficas, escogidas aleatoriamente al interior del grafo. |
| Resumen | Encontrar el camino más corto (menor número de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. **Respuesta en consola:** Muestre en la consola de texto el camino a seguir, informando el total de vértices, sus vértices (Id, Ubicación Geográfica) y la distancia estimada (en Km).  **Visualización mapa:** Muestre el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Se muestra por consola el camino, informando el total de vértices, sus vértices (Id, Ubicación Geográfica) y la distancia estimada (en Km); además, se muestra el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). | |
| Complejidad | |
| La complejidad es de O(E + 2V) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

|  |  |
| --- | --- |
| Nombre | 2B – Mostrar, a partir de las coordenadas de un área de interés de la ciudad y dos valores enteros (N >= 2 y M >= 2), los MxN vértices más cercanos en el grafo dentro del área. |
| Resumen | A partir de las coordenadas de un área de interés de la ciudad (LonMin, LatMin) y (LonMax, LatMax) y dos valores enteros (N >= 2 y M >= 2), definir una cuadricula regular de N columnas (incluyendo LonMin y LonMax) por M filas (incluyendo LatMin y LatMax). Las intersecciones de la cuadrícula así definida contienen N x M ubicaciones geográficas separadas de forma uniforme en el área de interés (incluyendo sus límites). Aproximar estas N x M ubicaciones a los vértices más cercanos en el grafo (usar la distancia harvesiana). El resultado de la aproximación va dar un conjunto de máximo N x M vértices en el grafo. **Respuesta en consola:** Muestre en la consola de texto el número de vértices en el grafo resultado de la aproximación. Mostar el identificador y la ubicación geográfica de cada uno de estos vértices.  Visualización mapa: Marque las ubicaciones de los vértices resultantes de la aproximación de la cuadrícula en Google Maps. |
| Entradas | |
| Las cooredenadas del área (LonMin, LatMin) y (LonMax, LatMax) y los dos valores enteros (N >= 2 y M >= 2). | |
| Resultados | |
| Se muestra en la consola el número de vértices, con sus identificadores y ubicaciones geográficas, en el grafo resultado de la aproximación; además, se marcan las ubicaciones de los vértices resultantes de la aproximación de la cuadrícula en Google Maps. | |
| Complejidad | |
| La complejidad es de O(E \* V) en el peor de los casos, E es el número de arcos y V es el número de vértices del grafo. | |

|  |  |
| --- | --- |
| Nombre | 1C - Calcular un árbol de expansión mínima con criterio distancia, utilizando el algoritmo de Kruskal, aplicado a la componente conectada más grande encontrada en el requerimiento 2A. |
| Resumen | Calcular un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Kruskal, aplicado a la componente conectada (subgrafo) más grande encontrada en el requerimiento 2A. **Respuesta en consola:** Muestre en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la siguiente información del árbol generado: los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final), y el costo total (distancia en Km) del árbol.  Visualización mapa: Muestre el árbol generado resultante en Google Maps: sus vértices y sus arcos. |
| Entradas | |
| La componente conectada encontrada en el requerimiento 2A. | |
| Resultados | |
| Se muestra en consola el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final), y el costo total (distancia en Km) del árbol; además, se muestran los vértices y arcos del árbol resultante en Google Maps. | |
| Complejidad | |
| La complejidad es de O(E \* V) en el peor de los casos, E es el número de arcos y V es el número de vértices del grafo. | |

|  |  |
| --- | --- |
| Nombre | 2C - Calcular un árbol de expansión mínima con criterio distancia, utilizando el algoritmo de Prim, aplicado a la componente conectada más grande encontrada en el requerimiento 2A. |
| Resumen | Calcular un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Prim, aplicado a la componente conectada (subgrafo) más grande encontrada en el requrimiento 2A. **Respuesta en consola:** Muestre en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la siguiente información del árbol generado: los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final), y el costo total (distancia en Km) del árbol.  Visualización mapa: Muestre árbol generado resultante en Google Maps: sus vértices y sus arcos. |
| Entradas | |
| La componente conectada encontrada en el requerimiento 2A. | |
| Resultados | |
| Se muestra en consola el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final), y el costo total (distancia en Km) del árbol; además, se muestran los vértices y arcos del árbol resultante en Google Maps. | |
| Complejidad | |
| La complejidad es de O(2V \* E) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

|  |  |
| --- | --- |
| Nombre | 3C - Calcular los caminos de costo mínimo (algoritmo de Dijkstra) con criterio distancia que conecten los vértices resultado de la aproximación de las ubicaciones de la cuadricula N x M encontrados en el requerimiento 2B. |
| Resumen | Calcular los caminos de costo mínimo (algoritmo de Dijkstra) con criterio distancia que conecten los vértices resultado de la aproximación de las ubicaciones de la cuadricula N x M encontrados en el requerimiento 2B. **Respuesta en consola:** Muestre en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y la siguiente información de cada camino resultante: su secuencia de vértices (identificadores) y su costo (distancia en Km).  **Visualización mapa:** Muestre los caminos de costo mínimo en Google Maps: sus vértices y sus arcos. Destaque el camino más largo (en distancia) usando un color diferente. |
| Entradas | |
| Las ubicaciones de la cuadricula N x M encontrados en el requerimiento 2B. | |
| Resultados | |
| Se muestra en la consola el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y de cada camino resultante la secuencia de vértices (identificadores) y su costo (distancia en Km); además, se muestran los vértices y sus arcos de los caminos de costo mínimo en Google Maps, destacando el camino más largo (en distancia) usando un color diferente. | |
| Complejidad | |
| La complejidad es de O(E + 2V) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

|  |  |
| --- | --- |
| Nombre | 4C - Encontrar el camino más corto (con criterio menor número de infracciones en la vía y menor cantidad de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Resumen | Encontrar el camino más corto (con criterio menor número de infracciones en la vía y menor cantidad de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. **Respuesta en consola:** Muestre en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y la siguiente información del camino resultante: su secuencia de vértices (identificadores), el total de infracciones y la distancia calculada (en Km).  **Visualización mapa:** Muestre el camino resultante en Google Maps: sus vértices y sus arcos. |
| Entradas | |
| Ninguna. | |
| Resultados | |
| Se muestra en la consola el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y la secuencia de vértices (identificadores), el total de infracciones y la distancia calculada (en Km) del camino resultante; además, se muestran los vértices y sus arcos del camino resultante en Google Maps. | |
| Complejidad | |
| La complejidad es de O(E + 2V) E es el número de arcos y V es el número de vértices en el peor de los casos. | |

Las estructuras de datos que se van a utilizar para el proyecto son grafo no dirigido, tabla de hash, arreglo dinámico, ColaPrioridadHeap

Estudiante B: JONATAN ESMISTS HERNANDEZ RUBIO

Estudiante A: Julián Eduardo Jaimes Castellanos