**Requerimientos**

|  |  |
| --- | --- |
| Nombre | **0.**Cargar el Grafo No Dirigido (grafo más grande) de la malla vial de la ciudad completa de Washington D.C. |
| Descripción | 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. |
| Entradas | JSON con la información del grafo grande de la malla vial de la ciudad completa de Washington. |
| Salidas | Grafo cargado y la información del total de vértices y de arcos que definen el grafo cargado. |
| Estimación de complejidad temporal | O(V+E) |
| Estructuras utilizadas | Grafo No Dirigido para cargar la información y archivo JSON de donde se lee la información. |

|  |  |
| --- | --- |
| Nombre | **1.** Al grafo creado, se debe agregar la información de cada una de las infracciones de todos los meses del año (nuevos archivos \*.csv). |
| Descripción | 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. |
| Entradas | Archivo con las infracciones de todos los meses del año (nuevos archivos \*.csv). |
| Salidas | Informar el total de infracciones cargadas y el número de las infracciones de cada mes. |
| Estimación de complejidad temporal | O(N) |
| Estructuras utilizadas | Grafo No Dirigido para cargar la información y archivo JSON de donde se lee la información de las infracciones. |

|  |  |
| --- | --- |
| Nombre | **2.** 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. |
| Descripción | 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. |
| Entradas | Latitud y Longitud. |
| Salidas | **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). |
| Estimación de complejidad temporal | O(E(log V)) |
| Estructuras utilizadas | Grafo No Dirigido para sacar la información y poder medir las distancias. |

|  |  |
| --- | --- |
| Nombre | **3.** Determinar los *n* vértices con mayor número de infracciones en la ciudad de Washington D.C. |
| Descripción | Determinar los *n* vértices con mayor número de infracciones en la ciudad de Washington D.C. *n* es un dato de entrada dado por el usuario. 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. |
| Entradas | *n* es un dato de entrada dado por el usuario. |
| Salidas | **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. |
| Estimación de complejidad temporal | O(V+E) |
| Estructuras utilizadas | Grafo No Dirigido que guarda la información, archivo de texto para leer en el buscador (pintar) y Heap para organizar de manera descendente. |

|  |  |
| --- | --- |
| Nombre | **4.**Encontrar el camino más corto (menor número de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Descripción | Encontrar el camino más corto (menor número de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Entradas | Latitud y Longitud |
| Salidas | **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). |
| Estimación de complejidad temporal | O(E(log V)) |
| Estructuras utilizadas | Grafo No Dirigido para sacar la información y poder medir las distancias. |

|  |  |
| --- | --- |
| Nombre | **5.**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). |
| Descripción | 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. |
| Entradas | Coordenada de área de interés, N y M |
| Salidas | **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. |
| Estimación de complejidad temporal | O(V) |
| Estructuras utilizadas | Grafo No Dirigido para saber la longitud y latitud de los vertices. |

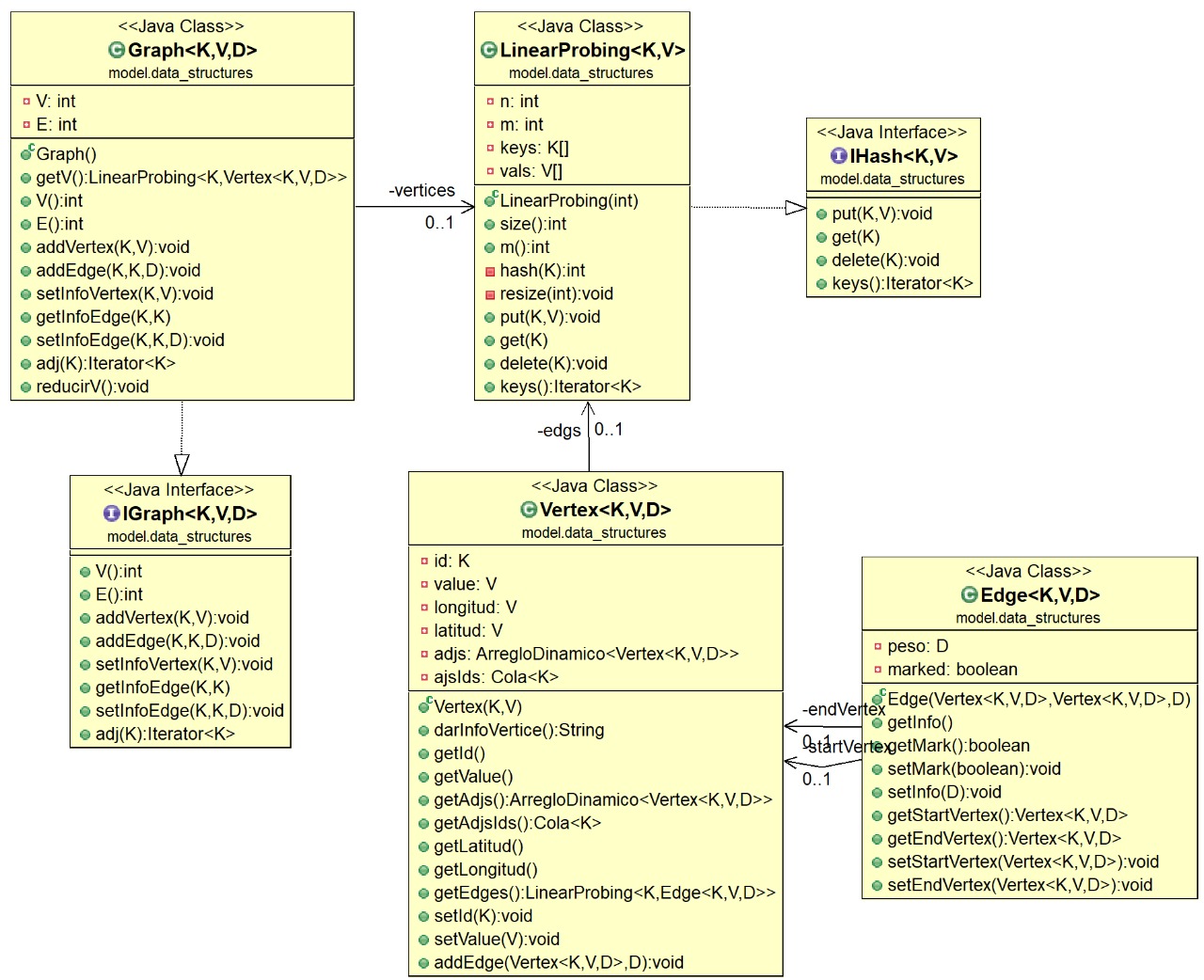
|  |  |
| --- | --- |
| Nombre | **6.** Calcular un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Kruskal, aplicado a la componente conectada (subgrafo) más grande encontrada en el punto 3. |
| Descripción | Calcular un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Kruskal, aplicado a la componente conectada (subgrafo) más grande encontrada en el punto 3. |
| Entradas | Subgrafo de punto 3 |
| Salidas | **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. |
| Estimación de complejidad temporal | O(E (log E)) |
| Estructuras utilizadas | Grafo No Dirigido para sacar los arcos que unen los vertices y Cola De Prioridad para ordenarlos de menor a mayor. |

|  |  |
| --- | --- |
| Nombre | **7.**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 punto 3. |
| Descripción | 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 punto 3. |
| Entradas | Subgrafo de punto 3 |
| Salidas | **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. |
| Estimación de complejidad temporal | O(E log( E )) |
| Estructuras utilizadas | Grafo No Dirigido para sacar los arcos que unen los vertices y Cola De Prioridad para ordenarlos de menor a mayor. |

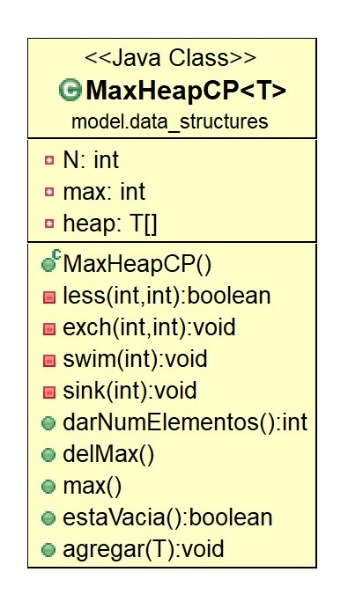
|  |  |
| --- | --- |
| Nombre | **8.**Calcular los caminos de costo mínimo (algoritmo de Dijkstra) con criterio distancia que conecten los vértices resultado de la aproximación de las ubicaciones de la cuadricula N x M encontrados en el punto 5. Respuesta en consola: Muestre en la consola de texto 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. |
| Descripción | 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 punto 5. |
| Entradas | Vértices resultado de punto 5. |
| Salidas | **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. |
| Estimación de complejidad temporal | O( E log (V )) |
| Estructuras utilizadas | Grafo No Dirigido para sacar los arcos que unen los vertices y Cola De Prioridad para ordenarlos de menor a mayor. |

|  |  |
| --- | --- |
| Nombre | **9.**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. |
| Descripción | 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. |
| Entradas | Latitud, longitud. |
| Salidas | **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. |
| Estimación de complejidad temporal | O( E log (V )) |
| Estructuras utilizadas | Grafo No Dirigido para sacar los arcos que unen los vertices y Cola De Prioridad para ordenarlos de menor a mayor. |

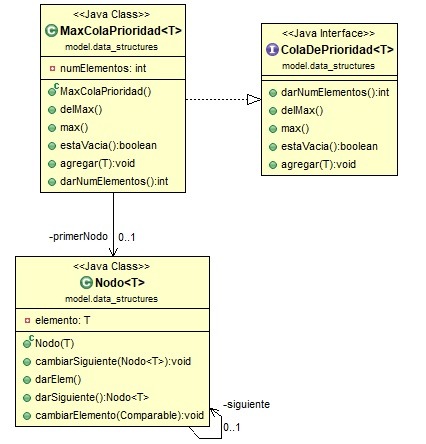
**UML Grafo:**



**UML Heap:**



**UML Cola Prioridad:**



**UML Solución:**

