Lista de Requerimientos

|  |
| --- |
| 0- Cargar el Grafo No Dirigido de la malla vial de la ciudad completa de Washington D.C. (formato JSON). |
| Entradas |
| 0 |
| Resultados |
| Grafo cargado. |

|  |
| --- |
| 1.2- Agregar la información al grafo de cada una de las infracciones de todos los meses del año (nuevos archivos \*.csv). |
| Entradas |
| nuevos archivos \*.csv (infracciones) |
| Resultados |
| Información cargada. |

|  |
| --- |
| 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. |
| Entradas |
| Latitud & Longitud |
| Resultados |
| En consola se muestra el camino a seguir, informando sus vértices(Id, Ubicación Geográfica), el costo mínimo (menor cantidad de infracciones), y la distancia estimada (en Km). En mapa se ve el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |

|  |
| --- |
| 3.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. |
| Entradas |
| n. |
| Resultados |
| 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 informa los identificadores de los vértices que la componen. Visualización mapa: marca la localización de los vértices resultantes en un mapa en Google Maps usando un color. Destaca su componente conectada más grande (con más vértices) usando un color y muestra sus vértices y sus arcos. |

|  |
| --- |
| 4.Encuentra el camino más corto (menor número de vértices) para un viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Entradas |
| Longitud & Latitud |
| Resultados |
| Muestra 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). Y muestra el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino) |

|  |
| --- |
| 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).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). |
| Entradas |
| (LonMin, LatMin) y (LonMax, LatMax) y dos valores enteros (N >= 2 y M >= 2) |
| Resultados |
| Muestra 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.  Y marca las ubicaciones de los vértices resultantes de la aproximación de la cuadrícula en Google maps |

|  |
| --- |
| 6.Calcular un árbol de expansión mínima(MST)con criterio distancia, utilizando el algoritmo de Kruskal, aplicado a la componente conectada (subgrafo) más grande encontradaen el punto 3. 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 generadoresultanteen Google Maps: sus vértices y sus arcos. |
| Entradas |
| Tupla (Xcoord, Ycoord) |
| Resultados |
| Realiza el ordenamiento de las infracciones en un Árbol Balanceado por localización geográfica. |

|  |
| --- |
| 7.Calcular un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Prim, aplicado a la componente conectada (subgrafo) más grande encontradaen el punto 3. 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 |
| [US$ valor inicial, US$ valor final] |
| Resultados |
| Para cada fecha en el semestre seleccionado, se calcula el valor acumulado (US$) de las infracciones para cada franja de hora completa. |

|  |
| --- |
| 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 |
| Entradas |
| Localización dada (AddressID) |
| Resultados |
| El total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, el valor total a pagar por las infracciones, y el Id de la calle (StreetSegId). |

|  |
| --- |
| 9.Encontrar el camino más corto (con criterio menor número de infracciones en la vía y menor cantidad de vértices) paraun viaje entre dos ubicaciones geográficas (latitud, longitud), escogidas aleatoriamente al interior del grafo. |
| Entradas |
| Rango de horas [HH:MM:SS inicial, HH:MM:SS final]. |
| Resultados |
| Información general: el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, y el valor total a pagar por las infracciones. Información por código (ViolationCode): Para las infracciones resultantes son agruparlas por código. Informar el detalle de código y el total de infracciones. |

|  |
| --- |
| Restricciones |
| • Los datos contenidos en los archivos sólo se pueden leer una vez |
| • Se deberá trabajar en Java 8 |
| • El proyecto se debe implementar en Eclipse |
| • La entrada/salida de información adicionales se debe realizar por consola |
| • No usar las colecciones del API Java. |