Proyecto 3 de Estructuras de Datos

Requerimientos Funcionales

**R1) Carga de archivos:**

Descripción: El código debe cargar la malla vial de Washington D.C de un archivo JSON y almacenarlo en grafo.

Entradas: Archivo JSON de malla vial. Archivo JSON de infracciones de todos los meses.

Salidas: Número de Vértices y Arcos de malla vial, y número de infracciones totales cargadas y totales por mes. Se resalta el camino en el mapa generado.

Orden Estimado: N (lectura de archivo JSON por año y conteo durante la carga)

**R2) Coste mínimo entre vértices:**

Descripción: Calcula el camino de coste mínimo entre dos vértices escogidos aleatoriamente. El método informa las estadísticas del camino escogido.

Entradas: - (Par de coordenadas (Lat1, Lon1) y (Lat2, Lon2) aleatorias).

Salidas: Distancia en km del camino de menor costo. Número de infracciones en el recorrido (costo). Vértices del camino realizado (Id, lat,lon). Camino resaltado en el mapa generado.

Orden Estimado: E\*log(N) (Algoritmo de Dijkstra)

**R3) Ranking de N vértices:**

Descripción: El código debe retornar los N vértices (ubicación, infracciones) ordenados de mayor a menor, mostrando las componentes conexas de los mismos y el identificador de la CC de los vértices. Adicionalmente este debe resaltar los vértices en el mapa de Washington y la componente conexa más grande obtenida.

Entradas: N infracciones.

Salidas: Estadísticas de N vértices con mayor número de infracciones (Lat1, Lon, Identificador). Mapa actualizado con vértices obtenidos y CC de mayor tamaño.

Orden Estimado: N (Máximo son realizados dos recorridos, el primero con el ordenamiento de las infracciones por n infracciones en un MaxColaPrioridad y el segundo en la identificación de componentes conexas).

**R4) Distancia mínima entre vértices:**

Descripción: Calcula el camino de distancia mínima entre dos vértices escogidos aleatoriamente. El método informa las estadísticas del camino escogido

Entradas: - ( Par de coordenadas (Lat1, Lon1) y (Lat2, Lon2) aleatorias).

Salidas: Distancia en km del camino de menor costo. Número de infracciones en el recorrido (costo). Vértices del camino realizado (Id, lat, lon). Camino resaltado en el mapa generado por el código.

Orden Estimado: E\*log(N) (Algoritmo de Dijkstra)

**R5) Vértices en cuadrícula:**

Descripción: Encontrar vértices más cercanos a cuadrícula recibida por parámetro. Los vértices retornados corresponden a la aproximación de estos a cada uno de los puntos de la cuadrícula.

Entradas: Número de columnas y filas de la cuadrícula (N >= 2 y M >= 2). Par de coordenadas mínimas y máximas de la cuadrícula (Lat1, Lon1) y (Lat2, Lon2).

Salidas: Vértices de cuadrícula aproximada (Id, lat, lon). Vértices resaltados en el mapa generado.

Orden Estimado: N\*M (búsqueda de nodos más cercanos en cantidad)

**R6) Árbol de expansión mínima para CC máxima Kruskal:**

Descripción: Se calcula el árbol de expansión mínima para la CC máxima obtenida en R3) por medio del algoritmo de Kruskal.

Entradas: CC máxima (se ejecuta el método asociado a R3)).

Salidas: MST asociado (id, E (Id inicial e Id final), y el costo total (distancia en Km)). Árbol resaltado en mapa generado. Tiempo en ms de implementación.

Orden Estimado: E\*log(N) (algoritmo de Kruskal)

**R7) Árbol de expansión mínima para CC máxima Prim:**

Descripción: Se calcula el árbol de expansión mínima para la CC máxima obtenida en R3) por medio del algoritmo de Prim.

Entradas: CC máxima (se ejecuta el método asociado a R3)).

Salidas: MST asociado (id, E (Id inicial e Id final), y el costo total (distancia en Km)). Árbol resaltado en mapa generado. Tiempo en ms de implementación.

Orden Estimado: E\*log(N) (algoritmo de Kruskal)

**R8) Caminos de orden mínimo en cuadrícula estimada:**

Descripción: El código encuentra el camino con menor distancia entre los vértices obtenidos en la cuadrícula de R5) por medio del algoritmo de Dijkstra.

Entradas: - (vértices obtenidos con R5))

Salidas: Vértices (id) y arcos incluidos (distancia en km). Tiempo en ms de implementación. Mapa resaltado con los caminos encontrados por el método y camino más largo de diferente color.

Orden Estimado: E\*log(N) (Algoritmo de Dijkstra ejecutado un número constante de veces)

**R9) Distancia/coste mínimo entre vértices:**

Descripción: Calcula el camino de distancia y coste mínima entre dos vértices cuyas coordenadas son escogidas aleatoriamente. El método informa las estadísticas del camino escogido

Entradas: - ( Par de coordenadas (Lat1, Lon1) y (Lat2, Lon2) aleatorias).

Salidas: Distancia en km del camino. Número de infracciones en el recorrido (costo). Vértices del camino realizado (Id). Camino resaltado en el mapa generado por el código.

Orden Estimado: E\*log(N)