|  |  |
| --- | --- |
| Nombre | R0 – Cargar el grafo. |
| Resumen | Carga el grafo de la malla vial de la ciudad completa de Bogotá |
| Entradas |  |
| Resultados | Informa el total de vértices y el total de arcos que definen el grafo cargado |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R1 – Agregar costo |
| Resumen | Carga el grafo de la malla vial de la ciudad completa de Bogotá |
| Entradas |  |
| Resultados | Se agregó respectivamente la información de costos |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R2 – Encontrar Id del vértice |
| Resumen | Dadas la longitud y la latitud encuentra el Id del Vértice de la malla vial más cercano por distancia Haversine |
| Entradas | Longitud y latitud |
| Resultados | Retorna el ID |
| Complejidad | O(n) |

PARTE A – Estudiante 1

|  |  |
| --- | --- |
| Nombre | R4 – Encontrar el camino de costo minimo |
| Resumen | Encuentra el camino de costo mínimo (menor tiempo promedio según Uber en la ruta) para un viaje entre dos localizaciones geográficas de la ciudad ((lat,long) origen, (lat, long) destino), ingresados por el usuario |
| Entradas | ((lat,long) origen, (lat, long) destino) |
| Resultados | Muestra en la consola de texto el camino a seguir, informando el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor tiempo promedio en segundos) y la distancia estimada (sumatoria de distancias Haversine en Km)  Muestra el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R5 – n vértices con menor velocidad promedio |
| Resumen | Determina los n vértices con menor velocidad promedio en la ciudad de Bogotá. Siendo la velocidad promedio de un vértice v, el promedio de las velocidades de todos sus arcos. |
| Entradas | N |
| Resultados | Muestra los n vértices resultantes en la consola de texto (su identificador, su ubicación (latitud, longitud), ordenados de menor a mayor por la velocidad promedio del vértice. Informar el número de componentes conectados (subgrafos) que se definen entre estos vértices en el grafo original. Por cada componente informar los identificadores de los vértices que la componen.  marca la localización de los n vértices resultantes en un mapa en Google Maps usando un color 1. Destaca la componente conectada más grande (con más vértices) usando un color 2. Para esta componente muestra sus vértices y sus arcos. |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R6 – árbol de expansión mínima |
| Resumen | Calcula un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Prim, aplicado al componente conectado (subgrafo) más grande de la malla vial de Bogotá. |
| Entradas |  |
| Resultados | Muestra en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la siguiente información del árbol generado: el total de vértices en el componente, los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final) y el costo total (distancia en Km) del árbol.  Muestre árbol generado resultante en Google Maps: sus vértices y sus arcos. |
| Complejidad | O(n) |

PARTE B – Estudiante 2

|  |  |
| --- | --- |
| Nombre | R7 – camino de menor costo |
| Resumen | Encuentra el camino de menor costo (menor distancia Haversine) para un viaje entre dos localizaciones geográficas de la ciudad ((lat,long) origen, (lat, long), destino), ingresadas por el usuario. Estas localizaciones deben aproximarse a las localizaciones más próximas en la malla vial. |
| Entradas | ((lat,long) origen, (lat, long), destino) |
| Resultados | Muestra en la consola de texto el camino a seguir, informando el total de vértices, sus vértices (Id, latitud, longitud), el tiempo estimado (la sumatoria de los tiempos de sus arcos) y la distancia Haversine estimada (sumatoria de distancias Haversine en Km).  Muestra el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R8 – Vértices alcanzables |
| Resumen | A partir de las coordenadas de una localización geográfica de la ciudad (lat, lon) de origen, indique cuáles vértices son alcanzables para un tiempo T (en segundos) dado por el usuario. La localización de origen debe aproximarse a la localización más próxima en la malla vial.. |
| Entradas | (lat, lon) |
| Resultados | Muestra en la consola los identificadores y la ubicación (lat, lon) de los vértices alcanzables en un tiempo T a partir de la localización de origen.  Marca la localización de origen en un color 1 y las localizaciones de los vértices alcanzables en un color 2 en Google Maps. |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R9 – árbol de expansión mínima |
| Resumen | Calcula un árbol de expansión mínima (MST) con criterio distancia, utilizando el algoritmo de Kruskal, aplicado al componente conectado (subgrafo) más grande de la malla vial de Bogotá. |
| Entradas |  |
| Resultados | Muestra en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la siguiente información del árbol generado: el total de vértices en la componente, los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final) y el costo total (distancia en Km) del árbol  Muestre el árbol generado resultante en Google Maps: sus vértices y sus arcos. |
| Complejidad | O(n) |

PARTE C

|  |  |
| --- | --- |
| Nombre | R10 – grafo simplificado |
| Resumen | Construye un nuevo grafo simplificado No dirigido de las zonas Uber, donde cada zona (MOVEMENT\_ID) es representada con un único vértice y los enlaces entre ellas representan su vecindad dentro de la malla vial. Es decir, si existe al menos un arco entre los vértices idA y idB en la malla vial y estos vértices pertenecen a la zonaX y la zonaY de Uber respectivamente, entonces las zonas zonaX y zonaY son vértices del grafo de zonas y estas zonas son adyacentes (vecinas) en este grafo. NO considerar los arcos que definan auto-ciclos entre una misma zona. Para cada zona seleccione una única localización (latitud, longitud) de referencia de manera que pueda luego visualizarse en el mapa. Adicionalmente, el costo del arco entre dos zonas vecinas corresponde al tiempo de viaje promedio reportado por Uber entre dichas zonas, teniendo en cuenta los tiempos promedio para los días (domingo, lunes, …, sábado) en que Uber haya reportado viajes entre dichas zonas. En caso de NO existir ningún tiempo de viajes en los datos Uber entre las dos zonas, asumir un valor predefinido de 200 segundos. Entre dos vértices (zonas) adyacentes solo debe crearse un arco, sin importar que pueda haber muchas intersecciones en común entre dichas zonas |
| Entradas |  |
| Resultados | Al final de la construcción del grafo de zonas, reporta la cantidad de vértices y arcos (cada arco debe contarse una única vez).  Muestre el grafo resultante en Google Maps: sus vértices (usando su localización de referencia) y sus arcos. Los arcos entre zonas conectan sus localizaciones de referencia. |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R11 – camino de costo mínimo |
| Resumen | Calcula el camino de costo mínimo (algoritmo de Dijkstra) basado en el tiempo promedio entre una zona de origen y una zona de destino sobre el grafo de zonas. |
| Entradas |  |
| Resultados | Muestra en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y del camino resultante: su secuencia de vértices/zonas (MOVEMENT\_ID) y su costo total (sumatoria de tiempos de los arcos en segundos). Adicionalmente, mostrar el tiempo promedio desde la zona origen hasta la zona destino reportado por el archivo de resolución semanal de Uber (teniendo en cuenta los tiempos promedio para los días (domingo, lunes, …, sábado) en que Uber haya reportado viajes entre dichas zonas). Esto con el propósito que el usuario pueda comparar el tiempo estimado en el grafo de zonas y el tiempo estimado con los datos de Uber entre la zona origen y la zona destino.  Muestra el camino de costo mínimo en Google Maps: sus vértices (usando su localización de referencia) y sus arcos. Los arcos entre zonas conectan sus localizaciones de referencia. |
| Complejidad | O(n) |

|  |  |
| --- | --- |
| Nombre | R12 – calcular los caminos de menor longitud |
| Resumen | . A partir de una zona origen, calcula los caminos de menor longitud (cantidad de arcos) a todas sus zonas alcanzables. De estos caminos,seleccionar el camino más largo (mayor cantidad de arcos); este será el camino desde la zona origen a su zona más distante (teniendo en cuenta el número de arcos) en el grafo de zonas. |
| Entradas |  |
| Resultados | Muestra en la consola de texto el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y del camino resultante más largo: su secuencia de vértices/zonas (MOVEMENT\_ID) y su número total arcos. Si hay múltiples caminos con la mayor longitud (número de arcos), mostrar entre estos caminos aquel que llegue al vértice destino con menor MOVEMENT\_ID  Muestre el camino más distante desde la zona de origen en Google Maps: sus vértices (usando su localización de referencia) y sus arcos. Los arcos entre zonas conectan sus localizaciones de referencia. |
| Complejidad | O(n) |