

Requerimientos Funcionales

Autores Julián Padilla y Pablo Pastrana

Fecha 12 – 05 -2020

Listado de Requerimientos

Nombre	1A – Obtener el camino de costo mínimo entre dos ubicaciones geográficas por distancia
Resumen	<p>Para encontrar el camino con el costo mínimo se va a tomar la distancia haversiana en cada arco como medida de base. Teniendo este dato, el usuario ingresa el punto de origen y el punto de llegada ingresando la latitud y la longitud, estos datos se deben encontrar dentro de los límites y aproximarse a los vértices de la malla vial de Bogotá. Después, de haber ingresado los datos el programa retorna al usuario el número de vértices, el costo mínimo de distancia y la distancia estimada.</p> <p>Complejidad: O(V+E)</p> <p>Para este requerimiento se utilizará un BFS que nos permitirá encontrar el camino más corto entre dos vértices establecidos, que en este caso son los datos ingresados por el usuario del punto de origen y el punto de partida, esta búsqueda por anchura se ejecutará sobre el grafo no dirigido (Malla vial de Bogotá), al final nos permite devolver la ruta más corta en distancia al usuario.</p>
Entradas	
<ul style="list-style-type: none">➤ Punto de origen – Latitud y Longitud➤ Punto de Llegada – Latitud y Longitud	
Resultados	
<ul style="list-style-type: none">➤ Respuesta por Consola Se muestra al usuario el total de vértices que debe recorrer, sus vértices (Id, latitud, longitud), el costo mínimo (menor distancia haversiana) y la distancia estimada (sumatoria de distancias haversianas en KM).➤ Visualización Mapa Se muestra el camino resultante en Google Maps (Incluyendo la ubicación de inicio y la ubicación de destino)	

Nombre	2A – Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los M puntos donde se presentan los comparendos de mayor gravedad
Resumen	<p>Primero el usuario ingresa el número de M de comparendos que se requieren consultar. Por otro lado, el distrito quiere instalar una red de cámaras de video en M sitios; para esto se requiere que la red tenga el menor costo de implementación el cual es de US\$10000 por cada kilómetro. Para esto se tomaron los M vértices donde más se presentan los comparendos de mayor gravedad. Para saber cuáles comparendos son de mayor gravedad, se sigue el siguiente criterio, el cual consiste en primero revisar que tipo de servicio (Criterio: Público>Oficial>Particular), segundo revisar el código de infracción en el caso de que dos comparendos tienen el mismo tipo de servicio y se compara el código de infracción usando el orden lexicográfico.</p> <p>Complejidad: O(n(V+E))</p> <p>Para este requerimiento se utilizará primero una tabla de hash SeparatingChainingST para almacenar los comparendos según el criterio de mayor gravedad mencionada anteriormente, después de tener los comparendos organizados se asignarán a los vértices del grafo que se acerquen a sus valores de latitud y longitud, teniendo esto en cuenta se revisará cuáles de los vértices tienen más comparendos de mayor gravedad y serán conectados entre sí, por último se utilizará un BFS para encontrar la ruta más corta que conecta a todos los vértices con las condiciones establecidas.</p>
Entradas	
<ul style="list-style-type: none">➤ M – Número de vértices que se requieren	
Resultados	
<ul style="list-style-type: none">➤ Respuesta en Consola Se muestra al usuario el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la información de la red propuesta la cual consta de los vértices (identificadores) y los arcos incluidos Id vértice inicial e ID vértice final), y el costo monetario total.➤ Visualización Mapa Se muestra en un mapa en Google Maps la red de comunicaciones propuesta. Asimismo, se resalta las M ubicaciones de las cámaras y los arcos de la red que las unen.	



Listado de Requerimientos

Nombre	1B – Obtener el camino de costo mínimo entre dos ubicaciones geográficas por número de comparendos
Resumen	<p>Para encontrar el camino con el costo mínimo se va a tomar la cantidad de comparendos en cada arco como medida de base. Teniendo este dato, el usuario ingresa el punto de origen y el punto de llegada ingresando la latitud y la longitud, estos datos se deben encontrar dentro de los límites y aproximarse a los vértices de la malla vial de Bogotá. Después, de haber ingresado los datos el programa retorna al usuario el número de vértices, la información de cada vértice, el costo mínimo de comparendos y la distancia estimada.</p> <p>Complejidad: $O(V+E)$</p> <p>Para este requerimiento se utilizará un BFS que nos permitirá encontrar el camino más corto entre dos vértices establecidos teniendo en cuenta la cantidad de comparendos en cada arco, para esto el usuario proporciona el punto de origen y el punto de partida, esta búsqueda por anchura se ejecutará sobre el grafo no dirigido (Malla vial de Bogotá), al final nos permite devolver la ruta mas corta con la menor cantidad de comparendos al usuario.</p>
Entradas	
<ul style="list-style-type: none">➤ Punto de origen – Latitud y Longitud➤ Punto de Llegada – Latitud y Longitud	
Resultados	
<ul style="list-style-type: none">➤ Resultado en Consola Se muestra al usuario el camino a seguir, informando el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor cantidad de comparendos) y la distancia estimada (sumatoria de distancias haversianas en KM).➤ Visualización Mapa Se muestra el camino resultante en Google Maps (Incluyendo la ubicación de inicio y la ubicación de destino)	

Nombre	2B – Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los M puntos donde se presenta el mayor número de comparendos en la ciudad
Resumen	<p>Primero el usuario ingresa el número de M de comparendos que se requieren consultar. Por otro lado, el distrito quiere instalar una red de cámaras de video en M sitios; para esto se requiere que la red tenga el menor costo de implementación el cual es de U\$10000 por cada kilometro. Para esto se tomaron los M vértices donde se presentan el mayor número de comparendos en la ciudad.</p> <p>Complejidad: $O(n(V+E))$</p> <p>Para este requerimiento se utilizará primero una tabla de hash SeparatingChainingST para almacenar por cada vértice el número de comparendos que se encuentra en un rango de latitud y longitud, donde la llave será el Id Vértice y el valor corresponderá al número de comparendos, después de tener los comparendos contabilizados por cada vértice, se incluirá esta información en el grafo, teniendo esto realizado se buscarán cuales vértices presentan mas cantidad de comparendos y serán conectados entre si, por ultimo se utilizará un BFS para encontrar la ruta mas corta que conecta a todos los vértices con las condiciones establecidas.</p>
Entradas	
<ul style="list-style-type: none">➤ M – Número de vértices que se requieren	
Resultados	
<ul style="list-style-type: none">➤ Respuesta en Consola Se muestra al usuario el tiempo que toma el algoritmo en encontrar la solución (en milisegundos), y la información de la red propuesta la cual consta de los vértices (identificadores) y los arcos incluidos (Id vértice inicial e ID vértice final), y el costo monetario total.➤ Visualización Mapa Se muestra en un mapa en Google Maps la red de comunicaciones propuesta. Asimismo, se resalta las M ubicaciones de las cámaras y los arcos de la red que las unen.	

Listado de Requerimientos

Nombre	1C – Obtener los caminos más cortos para que los policías puedan atender los M comparendos más graves
Resumen	<p>Para este requerimiento, se espera que cada ubicación de un comparendo grave debe ser atendida por la estación de policía mas cercana, es decir, la mínima distancia de desplazamiento usando la malla vial de Bogotá. Se asumirá que cualquier policía en una estación puede atender un comparendo y que todos los policías se encuentran en sus estaciones.</p> <p>Complejidad: $O(n(V+E))$</p> <p>Para este requerimiento se utilizará primero una Tabla de Hash (SeparateChaining) que nos permita almacenar en cada vértice los comparendos que se encuentran cerca, teniendo esto, estableceremos los vértices mas cercanos a cada estación teniendo en cuenta la distancia haversiana, con el numero de comparendos ingresado por el usuario estableceremos los mas cercanos a la estación, a partir de esto se aplicara un DFS el cual nos permitirá recorrer todos los adyacentes al vértice especificado y se utilizando JxMaps se pintarán las rutas correspondientes con colores diferentes para mostrar cuales rutas son mas eficientes.</p>
Entradas	
<p>➤ M – Número de comparendos que se quieren atender</p>	
Resultados	
<p>➤ Respuesta en Consola</p> <p>Se muestra al usuario el tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y los caminos resultantes para cada uno de los M comparendos: su secuencia de vértices y arcos, así como su costo total (sumatoria de distancia de los arcos en kilómetros).</p> <p>➤ Visualización Mapa</p> <p>Se muestra los caminos resultantes en Google Maps diferenciando la ubicación del comparendo (origen) y la ubicación de las estaciones de policía (destinos). Asigne un color diferente para graficar las rutas más eficientes de cada uno de los M comparendos.</p>	

Nombre	2C – Identificar las zonas de impacto de las estaciones de policía
Resumen	<p>Se ingresa todos los comparendos registrados en el año 2018 para ser manipulados en el requerimiento. Después de esta etapa, se va a identificar las zonas de impacto de cada un de las estaciones de policía y se asignara a cada estación los comparendos que puede atender en el menor tiempo posible; es decir, la estación que tenga la ruta más corta al lugar del comparendo.</p> <p>Tomando los caminos identificados anteriormente se creará un grafo tomando únicamente los vértices y arcos involucrados por dichos caminos. Asimismo, se definirá un esquema JSON para persistir la información del grafo. Después de esta etapa, se calculará los componentes conexos presentes en el grafo construido anteriormente, a cada componente se le asignará un color único.</p> <p>Complejidad: $O(n(V+E))$</p> <p>Para este requerimiento se utilizará primero una Tabla de Hash (SeparateChaining) que nos permita almacenar en cada estación de policía los comparendos con la ruta mas corta en donde fue tomado el comparendo, es decir el vértice punto de partido con sus conexiones a los lugares de la toma del comparendo, teniendo estos caminos establecido se creará un nuevo grafo no dirigido y se creará un archivo JSON, después de tener el grafo se reconocerá con la clase CC los componentes conectados a los cuales se les asignará un color para poder ser pintado con la clase Mapa que utiliza la librería JxMaps.</p>
Entradas	
<p>➤ No hay entrada por parte del usuario</p>	
Resultados	
<p>➤ Respuesta en Consola</p> <p>Se muestra al usuario la siguiente información:</p> <ul style="list-style-type: none">• El tiempo que toma el algoritmo en encontrar la solución (en milisegundos).• El número de vértices y arcos que tiene el grafo no dirigido.• La cantidad de comparendos que atiende cada una de las estaciones de policía.• Para cada componente conexas se imprime el color, el ObjectId de las estaciones de policía dentro del componente y el numero de vértices incluidos. <p>➤ Visualización Mapa</p> <p>A partir del grafo construido se pintará sobre el mapa de la malla vial de Bogotá utilizando Google Maps los siguientes componentes:</p> <ul style="list-style-type: none">• Un circulo en la posición de cada estación de policía. El área del circulo es proporcional al porcentaje de comparendos atendidos por dicha estación. El color del circulo es el mismo que el del componente conexo al que pertenece.• Cada arco del color del componente conexo al cual pertenece el vértice del grafo donde se producen los M comparendos de cada estación.	