# **Documentación de los requerimientos funcionales**

|  |  |
| --- | --- |
| **Nombre** | R1 – Cargar los comparendos |
| **Descripción** | Carga el archivo de comparendos en una lista enlazada simple, que contiene nodos, los cuales a su vez contendrán ítems (que serán los comparendos). Este método debe encargarse de extraer la información en formato JSON de cada uno de los comparendos y crear el respectivo nodo, que se agrega a la lista. |
| **Entradas** | |
| La ruta del archivo de comparendos como String. | |
| **Salidas** | |
| Se crea la lista enlazada simple de comparendos. El método retornará un booleano; *true* si la carga de los comparendos fue exitosa, *false* de lo contrario. | |
| **Estimación de la complejidad algorítmica** | . Pues solo se necesita recorrer una sola vez el archivo en formato JSON; es decir, basta con leer su información línea por línea, extraer los datos y paulatinamente ir creando el respectivo objeto (comparendo). |

|  |  |
| --- | --- |
| **Nombre** | R2 – Cargar estaciones de policía |
| **Descripción** | Carga el archivo de las estaciones de policía en un heap de prioridad. Este método debe encargarse de extraer la información en formato JSON para crear la respectiva estación de policía y agregarla al heap de prioridad. |
| **Entradas** | |
| La ruta del archivo de estaciones de policía como String. | |
| **Salidas** | |
| Se crea el heap de prioridad con las estaciones de policía. El método no retorna nada. | |
| **Estimación de la complejidad algorítmica** | . Ya que solo se necesita recorrer una vez el archivo en formato JSON; es decir, basta con leer su información línea por línea, extraer los datos y paulatinamente ir creando el respectivo objeto(Estación de policía).Se utiliza un heap de prioridad tipo MaxHeapCP, cuyo criterio de prioridad es la --------: , |

|  |  |
| --- | --- |
| **Nombre** | R3 – Cargar Grafo |
| **Descripción** | Realiza la carga del grafo a partir de las fuentes de datos bogota\_vertices.txt y bogota\_arcos.txt |
| **Entradas** | |
| La ruta del archivo de los vértices como string y la ruta del archivo de arcos de Bogotá como string. | |
| **Salidas** | |
| Una cadena de vértices y arcos creados como reporte de la carga de archivos. | |
| **Estimación de la complejidad algorítmica** | . Se hace una iteración para contar el número de vértices que necesita el grafo. Nuevamente, se hace otra para adjuntar la id y la información al respectivo grafo. Posteriormente, se hace un único recorrido por la ruta del archivo de arcos de Bogotá, se calcula el costo y se agrega con sus respectivos vertices. |

|  |  |
| --- | --- |
| **Nombre** | R4 – Requerimiento 1 |
| **Descripción** | Se encuentra el id del vértice más cercano dada una latitud y longitud, basada en la distancia haversiana. |
| **Entradas** | |
| La latitud y longitud como doubles. | |
| **Salidas** | |
| El id del vértice más cercano como string. | |
| **Estimación de la complejidad algorítmica** | . Se calcula la distancia haversiana entre cada uno de los vértices, de estos se obtiene el que tenga una distancia menor con los puntos dados, por lo que es el más cercano y solo se necesita hacer el recorrido una vez. |

|  |  |
| --- | --- |
| **Nombre** | R5 – Requerimiento 2 |
| **Descripcion** | Adicionar la información de los comparendos de 2018 a la malla vial por el vértice más cercanos. |
| **Entradas** | |
| La lista de comparendos | |
| **Salidas** | |
| No tiene salidas. | |
| **Estimación de la complejidad algorítmica** | . Se usa el requerimiento 1 para encontrar el id del vértice más cercano, basado en la coordenada del comparendo, guardando la respectiva información relevante del comparendo ahí. |

|  |  |
| --- | --- |
| **Nombre** | R6 – Requerimiento 3 |
| **Descripcion** | Agregar al grafo la información del costo de un arco. |
| **Entradas** | |
| Los dos vértices del arco. | |
| **Salidas** | |
| No tiene salida. | |
| **Estimación de la complejidad algorítmica** | . Se calcula la distancia haversiana entre los dos vértices y se agregan como el costo numero 1. Para el segundo costo, se cuenta el numero de comparendos almacenados en cada vértice y la suma total pasa a ser el segundo costo. |

|  |  |
| --- | --- |
| **Nombre** | R7 – Requerimiento 1A |
| **Descripcion** | Obtener el camino de costo mínimo entre dos ubicaciones geográficas por distancia. |
| **Entradas** | |
| La latitud y longitud de las dos ubicaciones geograficas. | |
| **Salidas** | |
| El total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor distancia haversiana) y la distancia estimada (sumatoria de distancias harvesianas en Km). Se muestra el camino resultante en google maps. | |
| **Estimación de la complejidad algorítmica** | . En base a las dos ubicaciones geográficas dadas, se encuentran los dos vértices más cercanos y se hace un dfs entre estos dos. Se calcula la distancia estimada sumando el costo de todos los vértices marcados después del dfs. |

|  |  |
| --- | --- |
| **Nombre** | R8 – Requerimiento 2A |
| **Descripcion** | 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. |
| **Entradas** | |
| El usuario ingresa un tamaño M de puntos para poner las cámaras de videos. | |
| **Salidas** | |
| Los vértices (identificadores) y los arcos incluidos, y el costo (monetario) total. Muestre en un mapa en Google Maps la red de comunicaciones propuesta. Resalte las M ubicaciones de las cámaras y los arcos de la red que las unen. | |
| **Estimación de la complejidad algorítmica** | . Se debe hacer un recorrido en el que por todos los vértices, se itere sobre los ítems de los comparendos y verifique si en este vértice hay uno de los comparendos de mayor gravedad. Se calcula el costo monetario en base al costo de los arcos multiplicado al costo por kilometro. |

|  |  |
| --- | --- |
| **Nombre** | R9 – Requerimiento 1B |
| **Descripción** | Obtener el camino de costo mínimo entre dos ubicaciones geográficas por número de comparendos. |
| **Entradas** | |
| No tiene | |
| **Salidas** | |
| El total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor cantidad de comparendos) y la distancia estimada (sumatoria de distancias harvesianas en Km). Se muestre el camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). | |
| **Estimación de la complejidad algorítmica** | . En base a las dos ubicaciones geográficas dadas, se encuentran los dos vértices más cercanos y se hace un dfs entre estos dos. Se calcula la distancia estimada sumando el costo de todos los ítems de los vértices después del dfs. |

|  |  |
| --- | --- |
| **Nombre** | R10 – Requerimiento 2B |
| **Descripcion** | 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. |
| **Entradas** | |
| Los M puntos a revisar | |
| **Salidas** | |
| El total de vértices en el componente, los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final) y el costo (monetario) total. Se muestre en un mapa en Google Maps la red de comunicaciones propuesta. Resalte las M ubicaciones de las cámaras y los arcos de la red que las unen. | |
| **Estimación de la complejidad algorítmica** | . Se debe hacer un recorrido en el que por todos los vértices, se itere sobre los ítems de los comparendos y se hace un algoritmo para encontrar el mayor basado en el número de comparendos que tenga el respectivo vértice. Se calcula el costo monetario en base al costo de los arcos multiplicado al costo por kilómetro. Estos datos se encuentran dentro de una tabla de hash. |

|  |  |
| --- | --- |
| **Nombre** | R11 – Requerimiento 1C |
| **Descripcion** | Obtener los caminos más cortos para que los policías puedan atender los M comparendos más graves |
| **Entradas** | |
| Los M puntos a atender. | |
| **Salidas** | |
| 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). Se muestra los caminos resultantes en Google Maps diferenciando la ubicación del comparendo (origen) y la ubicación de las estaciones de policía (destinos).+ | |
| **Estimación de la complejidad algorítmica** | . Se encuentran los vértices con M comparendos más graves , se debe hacer un recorrido en el que por todos los vértices, se itere sobre los ítems de los comparendos para saber si tiene los comparendos más graves. Luego se hace un dfs desde el vértice donde está almacenado la estación más cercana para marcar el camino .Se calcula el costo monetario en base al costo de los arcos multiplicado al costo por kilómetro. La secuencia de vértices son los vértices marcados y la secuencia de arcos, los respectivos arcos en el camino. |

|  |  |
| --- | --- |
| **Nombre** | R11 – Requerimiento 2C |
| **Descripcion** | Identificar las zonas de impacto de las estaciones de policía. |
| **Entradas** | |
| El número de vértices y arcos que tiene el grafo no dirigido, a cantidad de comparendos que atiende cada una de las estaciones de policía, para cada componente conexa se imprime el color, el ObjectId de las estaciones de policía dentro del componente, y el número de vértices incluidos. | |
| **Salidas** | |
| la consola de texto 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). Se muestra los caminos resultantes en Google Maps diferenciando la ubicación del comparendo (origen) y la ubicación de las estaciones de policía (destinos).  En Google maps,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 a la que pertenece.  Se generan arcos del color del componente conexo al cual pertenece el vértice del grafo donde se producen los M primeros comparendos de cada estación. | |
| **Estimación de la complejidad algorítmica** | S Se recorre una vez la lista de los comparendos, nuevamente, usando dfs encontramos el camino más corto según la posición geográfica del comparendo hasta la estación de policía más cercana y guardamos el comparendo en una tabla de hash que tiene como llave la estación de policía y sus valores son los comparendos más cercanos. Se usan los vértices asociados y los arcos para crear el grafo y se usa el método CC para encontrar todos los componentes conexos. |