**Proyecto 3: Requerimientos Funcionales**

1:

|  |  |
| --- | --- |
| **Nombre** | ClosestVertex |
| **Resumen** | Dado un punto identificable por su latitud y longitud, se busca el vértice más cercano, según sui distancia harvesiana. |
| **Entradas** | Latitud 1, longitud1 |
| **Resultados** | Un objeto de la clase vertice que es el vertice con la menor distancia harvesiana al punto pasado por parametro |
| **Complejidad** | O (Log N) en el caso esperado y O(N) en el peor caso |
| **Estructuras a utilizar** | Se utilizará un kd-tree en este caso con k = 2 |

2:

|  |  |
| --- | --- |
| **Nombre** | AddInfraction |
| **Resumen** | Dado una infracción con su latitud y longitud, se busca el vértice de la malla vial más cercano y a este se agrega la información del comparendo |
| **Entradas** | Comparendo |
| **Resultados** | Se agrega la información del comparendo al vértice de la malla vial más cercano. |
| **Complejidad** | O (Log N) en el caso esperado y O(N) en el peor caso |
| **Estructuras a utilizar** | Se utilizará un kd-tree y un arreglo dinámico donde se tendrán todos los vértices de la malla vial |

3A:

|  |  |
| --- | --- |
| **Nombre** | addCostInfo |
| **Resumen** | Dada la información de costo de un arco se agrega al grafo |
| **Entradas** | La distancia haversiana, el total de comparendos |
| **Resultados** | Se agrega la información al grafo |
| **Complejidad** | O(1) |
| **Estructuras a utilizar** | El grafo de la malla vial. |

4:

|  |  |
| --- | --- |
| **Nombre** | AddPoliceStationInfo |
| **Resumen** | Dado una estación de policia con su latitud y longitud, se busca el vértice de la malla vial más cercano y a este se agrega la información de la estación. |
| **Entradas** | Estación de policía. |
| **Resultados** | Se adiciona la información de la estación de policía al grafo |
| **Complejidad** | O (Log N) en el caso esperado y O(N) en el peor caso |
| **Estructuras a utilizar** | Se utilizará un kd-tree y un arreglo dinámico donde se tendrán todos los vértices de la malla vial |

1A:

|  |  |
| --- | --- |
| **Nombre** | MinimumDistanceJourney |
| **Resumen** | Dado dos puntos, uno de origen y otro de destino, se busca hallar la distancia más corta entre ambos teniendo en cuenta la distancia harvesiana. |
| **Entradas** | Latitud origen, Latitud destino, Longitud origen, longitud destino |
| **Resultados** | Total de nodos y de cada nodo su id, latitud y longitud, y la distancia total estimada. |
| **Complejidad** | O(E+V+logV) con Dijkstra |
| **Estructuras a utilizar** | Se utilizara el un minQueue, un arreglo dinámico y el grafo con los vértices y arcos. |

2A:

|  |  |
| --- | --- |
| **Nombre** | MSTCamerasImportance |
| **Resumen** | Dado un número M cámaras que se quieren instalar, devolver la red de comunicaciones que soporte su instalación, que tenga los M vértices con mayor importancia en términos de infracciones y que tenga el mínimo costo de construcción |
| **Entradas** | La cantidad de cámaras requeridas M. |
| **Resultados** | El MST requerido sin caminos innecesarios. |
| **Complejidad** | O(ElogV) |
| **Estructuras a utilizar** | Se utilizara el un minQueue, un arreglo dinámico y el grafo con los vértices y arcos. |

1B.

|  |  |
| --- | --- |
| **Nombre** | MinimumInfractionsJourney |
| **Resumen** | Dado dos puntos, uno de origen y otro de destino, se busca hallar la distancia más corta entre ambos teniendo en cuenta el numero de infracciones en el camino. |
| **Entradas** | Latitud origen, Latitud destino, Longitud origen, longitud destino |
| **Resultados** | Total de nodos y de cada nodo su id, latitud y longitud, y la distancia total estimada. |
| **Complejidad** | O(E+V+logV) con Dijkstra |
| **Estructuras a utilizar** | Se utilizara el un minQueue, un arreglo dinámico y el grafo con los vértices y arcos. |

2B:

|  |  |
| --- | --- |
| **Nombre** | MSTCamerasQuantity |
| **Resumen** | Dado un número M cámaras que se quieren instalar, devolver la red de comunicaciones que soporte su instalación, que pase por los M vértices con más comparendos y que tenga el mínimo costo de construcción |
| **Entradas** | La cantidad de cámaras requeridas M. |
| **Resultados** | El MST requerido sin caminos innecesarios. |
| **Complejidad** | O(ElogV) |
| **Estructuras a utilizar** | Se utilizara el un minQueue, un arreglo dinámico y el grafo con los vértices y arcos. |

1C

|  |  |
| --- | --- |
| **Nombre** | MSTPolice |
| **Resumen** | Dado un número M, devolver la red de comunicaciones conecte los M vértices con más comparendos y que la suma de sus caminos sea mínima. |
| **Entradas** | La cantidad M de puntos que se quieren visitar obligatoriamente. |
| **Resultados** | El MST requerido sin caminos innecesarios. |
| **Complejidad** | O(ElogV) |
| **Estructuras a utilizar** | Se utilizara el un minQueue, un arreglo dinámico y el grafo con los vértices y arcos. |

2C

|  |  |
| --- | --- |
| **Nombre** | CCInfractions |
| **Resumen** | Crear una visualización de todos los componentes conectados según las estaciones de policía y los comparendos que atienden. |
| **Entradas** | Ninguna. |
| **Resultados** | Una visualizacion de todos los components conectados. |
| **Complejidad** | O(V+E) |
| **Estructuras a utilizar** | un arreglo dinámico y el grafo con los vértices y arcos. |