REQUERIMIENTOS FUNCIONALES

|  |  |
| --- | --- |
| **Nombre** | **1A- Obtener los M comparendos con mayor gravedad.** |
| Resumen | Este requerimiento imprime la información de M comparendo por la gravedad de este. Para saber si un comparendo es más grave que otro primero se mira el tipo de servicio. Público es más grave que Oficial y Oficial es más grave que Particular. Si dos comparendos tienen el mismo tipo de servicio se compara el código de la infracción (campo INFRACCION) usando el orden lexicográfico (forma de comparación de los Strings en Java, A12 es más grave que A11 y B10 es más grave que A10). |
| Entradas | El usuario ingresa el número M, el cual representa la cantidad de comparendos que desea ver. |
| Resultado | Se mostrará en consola los M comparendos de mayor a menor gravedad. De cada comparendo se mostrará el id, tipo de servicio, infracción, fecha-hora y clase de vehículo. MaxHeapCP<Comparendo> |
| Complejidad | O(n) |
| Estructuras que se usaran | En este método solo se usa una MaxHeapCP<Comparendo> la cual ingresa los Comparendos teniendo la gravedad descrita en el Resumen. Se usa esta estructura ya que es la forma mas eficiente de poder tomar M datos sin hacer varios recorridos. |

|  |  |
| --- | --- |
| **Nombre** | **2A- Buscar los comparendos por mes y día de la semana.** |
| Resumen | El usuario ingresa el número del mes (1-12) y el día de la semana (L, M, I, J, V, S, D), y Se mostrará en consola los N (Donde N inicialmente es 20) comparendos que cumplan los criterios de búsqueda. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora y clase de vehículo. |
| Entradas | El usuario ingresa el número del mes (1-12) y el día de la semana (L, M, I, J, V, S, D). |
| Resultado | Se mostrará en consola los N (Donde N inicialmente es 20) comparendos que cumplan los criterios de búsqueda. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora y clase de vehículo. MaxHeapCP<Comparendo> |
| Complejidad | O(n) |
| Estructuras que se usaran | Las estructuras usadas serán, primero una SeparateChaining<LLaves2A,MaxHeapCP<Comparendo>> donde se ingresa una Llaves2A que esta compuesta por el día y el mes de cada comparendo y agrega ese comparendo a la MaxHeapCP<Comparendo>> correspondiente, luego se retorna la MaxHeapCP<Comparendo>> que se da después de usar el get(…) con la llave que se crea con la entradas del método. |

|  |  |
| --- | --- |
| **Nombre** | **3A- Buscar los comparendos que tienen una fecha-hora en un rango y que son de una localidad dada.** |
| Resumen | El usuario ingresa un rango de fecha-hora [limite\_bajo, limite\_alto] en formato “YYYY/MM/DD-HH:MM:ss” y una localidad. Se mostrará en consola los N (Donde N inicialmente es 20) comparendos que cumplan los criterios de búsqueda. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora y clase de vehículo. |
| Entradas | El usuario ingresa un rango de fecha-hora [limite\_bajo, limite\_alto] en formato “YYYY/MM/DD-HH:MM:ss” y una localidad. |
| Resultado | Se mostrará en consola los N (Donde N inicialmente es 20) comparendos que cumplan los criterios de búsqueda. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora y clase de vehículo. |
| Complejidad | O() |
| Estructuras que se usaran | Se usarán MaxHeapCP<Comparendo>> y ArbolRojoNegro. Ya que hace el procedimiento mas optimo. |

|  |  |
| --- | --- |
| **Nombre** | **1B- Buscar los M comparendos más cercanos a la estación de policía.** |
| Resumen | Para los oficiales de la estación de policía del Campin (Ak 30 #57a8, Bogotá, latitud: 4.647586, longitud: - 74.078122) los comparendos más importantes son los que están más cercanos a su estación. Un oficial debe poder ingresar un número M y se deben mostrar en consola los M comparendos más cercanos. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora, clase de vehículo, longitud y latitud geográficas. |
| Entradas | El usuario ingresa el número M, el cual representa la cantidad de comparendos que desea ver. |
| Resultado | mostrar en consola los M comparendos más cercanos. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora, clase de vehículo, longitud y latitud geográficas. |
| Complejidad | O(n) |
| Estructuras que se usaran | En este método solo se usa una MaxHeapCP<Comparendo> la cual ingresa los Comparendos teniendo la gravedad descrita en el Resumen. Se usa esta estructura ya que es la forma más eficiente de poder tomar M datos sin hacer varios recorridos. |

|  |  |
| --- | --- |
| **Nombre** | **2B- Buscar los comparendos por medio de detección, clase de vehículo, tipo de servicio y localidad.** |
| Resumen | El usuario ingresa el medio de detección, clase de vehículo, tipo de servicio y localidad. Se debe mostrar en consola los N comparendos que cumplan los criterios de búsqueda ordenados por fecha. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora, clase de vehículo y localidad. |
| Entradas | El usuario ingresa el medio de detección, clase de vehículo, tipo de servicio y localidad. |
| Resultado | Se debe mostrar en consola los N comparendos que cumplan los criterios de búsqueda ordenados por fecha. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora, clase de vehículo y localidad. |
| Complejidad | O(n) |
| Estructuras que se usaran | Las estructuras usadas serán, primero una SeparateChaining<LLaves2B,MaxHeapCP<Comparendo>> donde se ingresa una Llaves2B que está compuesta por el medio de detección, clase de vehículo, tipo de servicio y localidad de cada comparendo y agrega ese comparendo a la MaxHeapCP<Comparendo>> correspondiente, luego se retorna la MaxHeapCP<Comparendo>> que se da después de usar el get(…) con la llave que se crea con la entradas del método. |

|  |  |
| --- | --- |
| **Nombre** | **3B- Buscar los comparendos que tienen una latitud en un rango dado y que involucraron un tipo de vehículo particular** |
| Resumen |  |
| Entradas | El usuario ingresa un rango de latitudes [limite\_bajo, limite\_alto] y una clase de vehículo (automovil, camioneta, moto, etc.) |
| Resultado | Se debe mostrar en consola los N comparendos que cumplan los criterios de búsqueda. De cada comparendo se debe mostrar el id, tipo de servicio, infracción, fecha-hora, clase de vehículo y latitud geográfica. |
| Complejidad | O() |
| Estructuras que se usaran | Se usarán MaxHeapCP,y ArbolRojoNegro. Ya que hace que el procedimiento sea más óptimo. |

|  |  |
| --- | --- |
| **Nombre** | **1C- Visualizar Datos en una Tabla ASCII** |
| Resumen | Se quiere mostrar una tabla como la anterior para poder entender los datos. El usuario debe ingresar un número de días D y se debe mostrar una tabla dividida por rangos de tamaño D mostrando el número de comparendos en ese rango (En la tabla anterior D es 7). El valor de cada “\*” lo determina el programa para que la tabla quepa en la consola. |
| Entradas | El usuario debe ingresar un número de días D. |
| Resultado | mostrar una tabla dividida por rangos de tamaño D mostrando el número de comparendos en ese rango (En la tabla anterior D es 7). El valor de cada “\*” lo determina el programa para que la tabla quepa en la consola. |
| Complejidad | O() |
| Estructuras que se usaran | Se usarán MaxHeapCP, SeparateChaining y ArbolRojoNegro. Ya que hace que el procedimiento sea más óptimo. |

|  |  |
| --- | --- |
| **Nombre** | **2C- El costo de los tiempos de espera hoy en día (cola)** |
| Resumen | Da información para comprender el sistema antiguo que maneja la policía. |
| Entradas | Ninguna |
| Resultado | Mostrar en consola los siguientes datos:  • El costo total que generan las penalizaciones en 2018  • El número de días en promedio que debe esperar un comparendo  • Una histograma ASCII con el número comparendos procesados por día y el número de comparendos que están esperando. Recuerde que por día se pueden procesar manualmente máximo 1500 comparendos.  • Para los comparendos que tienen un costo diario de $400 indicar cual fue el tiempo mínimo que estuvieron en espera, el tiempo promedio y el tiempo máximo. Hacer lo mismo para los comparendo con un valor de $40 y de $4. Mostrar la información anterior en una tabla |
| Complejidad | O() |
| Estructuras que se usaran | Se usarán MaxHeapCP, SeparateChaining y ArbolRojoNegro. Ya que hace que el procedimiento sea más óptimo. |

|  |  |
| --- | --- |
| **Nombre** | **3C- El costo de los tiempos de espera usando el nuevo sistema** |
| Resumen | Da información para comprender el sistema nuevo que maneja la policía. |
| Entradas | Ninguna |
| Resultado | Mostrar en consola los siguientes datos:  • El costo total que generan las penalizaciones en 2018  • El número de días en promedio que debe esperar un comparendo  • Una histograma ASCII con el número comparendos procesados por día y el número de comparendos que están esperando. Recuerde que por día se pueden procesar manualmente máximo 1500 comparendos.  • Para los comparendos que tienen un costo diario de $400 indicar cual fue el tiempo mínimo que estuvieron en espera, el tiempo promedio y el tiempo máximo. Hacer lo mismo para los comparendo con un valor de $40 y de $4. Mostrar la información anterior en una tabla |
| Complejidad | O() |
| Estructuras que se usaran | Se usarán MaxHeapCP, SeparateChaining y ArbolRojoNegro. Ya que hace que el procedimiento sea más óptimo. |

|  |  |
| --- | --- |
| **Nombre** | **4C- Conclusión Final** |
| Resumen | Este documento se encuentra en la carpeta docs del proyecto. |