**DOCUMENTACIÓN REQUERIMIENTOS FUNCIONALES**

**ESTUDIANTE A:**

|  |  |
| --- | --- |
| 1. Obtener los M comparendos con mayor gravedad. | |
| Descripción: | Dado el tipo de servicio y orden lexicográfico en el código de infracción se va a determinar un orden de prioridad el cual lo referenciaremos como gravedad. Se devolverá los M comparendos con mayor gravedad. |
| Datos de Entrada: | El número de comparendos que desea obtener. |
| Datos de Salida: | El número de comparendos solicitados con mayor gravedad descendentemente. |
| Estimación de complejidad: | La complejidad teniendo en cuenta que vamos a implementar una cola de prioridad es en el mejor caso de retorno del máximo 1 y en general es logN. |

|  |  |
| --- | --- |
| 2. Buscar los comparendos por mes y día de la semana. | |
| Descripción: | Con base a un mes y un día de la semana se devuelven los N comparendo que cumplan. |
| Datos de Entrada: | Se ingresa el número del mes (1-12) y el día de la semana (L, M, I, J, V, S, D). |
| Datos de Salida: | Se devuelve los N comparendos que cumplan. |
| Estimación de complejidad: | Dado que se va a realizar una búsqueda sobre una tabla de hash la complejidad es contante: 3-5\*. |

|  |  |
| --- | --- |
| 3. Buscar los comparendos que tienen una fecha-hora en un rango y que son de una  localidad dada | |
| Descripción: | Se ingresa un rango de fecha-hora [limite\_bajo, limite\_alto] en formato “YYYY/MM/DD-HH:MM:ss” y una localidad. |
| Datos de Entrada: | Se muestran los N comparendos que cumplan los criterios de búsqueda. |
| Datos de Salida: | Dado una localidad específica y un rango fecha-hora se debe mostrar los comparendos que cumplan con estos criterios. |
| Estimación de complejidad: | La complejidad de esto teniendo en cuenta que usaremos un árbol Rojo-Negro es 1.0logN |

**ESTUDIANTE B:**

|  |  |
| --- | --- |
| 1. Mostrar los M comparendos más cercanos a la estación de policía | |
| Descripción: | Se ingresa el número de comparendos que se quieren mostrar, y a partir de ese número se muestran los comparendos con menor distancia. Para ello, planteamos una cola de prioridad en la que el criterio es la distancia, llevada hacia el mínimo, es decir, tendremos una cola de prioridad donde el primer elemento es el de menor distancia a la estación de policía |
| Datos de Entrada: | El número M |
| Datos de Salida: | La cola con los comparendos |
| Estimación de complejidad: | Se estima complejidad lineal, ya que los comparendos ya están cargados en la cola, es hacerles el respectivo dequeue dependiendo de M |

|  |  |
| --- | --- |
| 2. Buscar los comparendos por medio de detección, clase de vehículo, tipo de servicio y localidad | |
| Descripción: | Dados los cuatro parámetros que se mencionan anteriormente, se desarrollará una tabla de hash por encadenamiento separado, en donde la llave será la concatenación de estos cuatro parámetros |
| Datos de Entrada: | Medio de detección, clase de vehículo, tipo de servicio, localidad y los M primeros comparendos a mostrar |
| Datos de Salida: | La cola con los primeros M comparendos que cumplan con criterio de búsqueda, ordenados por fecha. Sin embargo, para poder hacer el ordenamiento puede que se devuelva un arreglo de comparable ya ordenado |
| Estimación de complejidad: | Se estima complejidad lineal, en el caso de muchas colisiones |

|  |  |
| --- | --- |
| 3. Buscar los comparendos que tienen una latitud entre un rango dado y que involucraron un tipo de vehículo particular | |
| Descripción: | Dados los límites del intervalo de latitud, y le tipo de vehículo, con un árbol 2-3 de criterio longitud, se evalúa el tipo de vehículo, para así terminar de filtrar en la búsqueda |
| Datos de Entrada: | Extremos del intervalo de latitud, el tipo de vehículo y el numero N de comparendos a mostrar |
| Datos de Salida: | Los datos de los comparendos que cumplen con esta condición |
| Estimación de complejidad: | Teniendo en cuenta que es un árbol rojo-negro, la complejidad es 1logN |

**PARTE C:**

|  |  |
| --- | --- |
| 1. Visualizar Datos en una Tabla ASCII | |
| Descripción: | Se busca generar una tabla ASCII con base a los datos precedentes que llegan en una estructura de cola donde se aclara la información para los días partidos en rangos, y una cantidad de “\*”, donde cada \* vale por un determinado número de comparendos. El tamaño de los rangos es D, el cual ingresa el usuario. |
| Datos de Entrada: | Se ingresa D que es la cantidad de días que va a tener cada rango individual de fechas a mostrar. |
| Datos de Salida: | Se imprime la tabla ASCII con los rangos y sus respectivos números de \*. |
| Estimación de complejidad: | La complejidad de esto teniendo en cuenta que se va a recorrer la cola a totalidad es N. |

|  |  |
| --- | --- |
| 2. El costo de los tiempos de espera hoy en día (cola) | |
| Descripción: | Con base a los datos suministrados, analizar estos de tal manera que por día se pueda recopilar cuantos comparendos son procesados por día (\* en la tabla ASCII) y el número de comparendos que están esperando (# en la tabla ASCII); a la par del número de días en promedio que debe esperar un comparendo y su costo total. Recopilando para casa costo diario, tiempo mínimo, máximo y promedio de días en espera. |
| Datos de Entrada: | El archivo de comparendos. |
| Datos de Salida: | • El costo total que generan las penalizaciones en 2018  • El número de días en promedio que debe esperar un comparendo  • Un 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.   |  |  |  |  | | --- | --- | --- | --- | | Costo diario del comparendo | Tiempo mínimo de espera (días) | Tiempo promedio de espera (días) | Tiempo máximo de espera (días) | | $400 |  |  |  | | $40 |  |  |  | | $4 |  |  |  | |
| Estimación de complejidad: | La complejidad de esto teniendo en cuenta que se va a recorrer la cola a totalidad es N. |

|  |  |
| --- | --- |
| 3. El costo de los tiempos de espera usando el nuevo sistema | |
| Descripción: |  |
| Datos de Entrada: | Los datos del modelo anterior funcionan como una cola, entonces lo que se busca es una tabla de hash, en donde la llave sea la fecha, y que en cada posición del arreglo de la tabla de Hash, estén los comparendos para esa fecha, y que en cada una de estas colas, haya una cola de prioridad, en donde el máximo será el comparendo que genere una mayor multa. Con eso, podremos procesar los comparendos que más están atrasados o los que podrían generarme la mayor multa, reduciendo los costos para la secretaría de movilidad |
| Datos de Salida: | El nuevo costo que tiene que asumir la secretaría de movilidad con el sistema implementado por nosotros |
| Estimación de complejidad: | Una estimación de complejidad inicial es N log N, teniendo en cuenta las búsquedas de la tabla de Hash por encadenamiento separado con los de la búsqueda de prioridad, y las veces que se tiene que ejecutar el código. |