Proyecto de Estructuras de Datos

Requerimientos Funcionales

R1) Lectura de archivos:

Descripción: El código debe ser capaz de leer todos los archivos CSV de tránsito de un cuatrimestre seleccionado por el usuario de 2018 en Washington D.C y almacenarlos en estructuras de datos cola y fila.

Entradas: Selección del cuatrimestre escogido por el usuario.

Salidas: Estructuras de datos con infracciones de los archivos almacenados.

Orden Estimado: N (lectura de archivo CSV por cuatrimestre)

R2) Verificación de identificación única:

Descripción: El código debe ser verificar la independencia de los identificadores de todas las infracciones del cuatrimestre seleccionado por el identificador OBJECTID.

Entradas: -

Salidas: Estructura de datos con los objetos con más de una identificación.

Orden Estimado: N^2 (un recorrido por elemento)

R3) Consultar infracciones por fecha/hora inicial y fecha/hora final:

Descripción: El código debe ser capaz de leer y retornar todas las infracciones del cuatrimestre seleccionado por la fecha y hora inicial, y, la fecha y hora final especificadas.

Entradas: Fecha y hora, inicial y final de búsqueda.

Salidas: OBJECTID, TICKETISSUEDATE de las infracciones encontradas en forma de cola.

Orden Estimado: N

R4) Consultar FINEAMT promedio para accidentalidad y no accidentalidad:

Descripción: El código debe ser capaz calcular el FINEAMT promedio de todas las infracciones con VIOLATIONCODE especificado para accidentalidad y no accidentalidad.

Entradas: VIOLATIONCODE de infracciones.

Salidas: FINEAMT promedio por: infracciones con accidente, e infracciones sin accidente.

Orden Estimado: N

R5) Consultar infracciones en un rango de fechas en una dirección ordenadas por STREETSEGID:

Descripción: El código consulta las infracciones de una dirección y rango de fechas determinado. El OBJECTID, TICKETISSUEDAT, STREETSEGID y ADDRESS\_ID resultantes se ordenan descendientemente por STREETSEGID en una pila.

Entradas: ADRESS\_ID de las infracciones.

Rango de fechas a buscar.

Salidas: Pila ordenada descendientemente por STREETSEGID con OBJECTID, TICKETISSUEDAT, STREETSEGID y ADDRESS\_ID de las infracciones encontradas.

Orden Estimado: 2\*N

R6) Consultar FINEAMT promedio con OBJECTID en un rango determinado:

Descripción: El código debe ser capaz retornar una estructura de datos que contenga el VIOLATIONCODE y el FINEAMT promedio cuyo FINEAMT promedio se encuentre en un rango determinado.

Entradas: Rango de FINEAMT.

Salidas: FINEAMT promedio por VIOLATIONCODE dentro del rango en una pila.

Orden Estimado: N

R7) Obtener facturas por orden de facturación con TOTALPAID en un rango determinado:

Descripción: El código debe retornar una estructura de datos ordenada ascendente o descendentemente por fecha con: OBJECTID, TICKETISSUEDATE y el TOTALPAID, cuyo TOTALPAID se encuentre en un rango determinado.

Entradas: Rango de TOTALPAID.

Orden de la estructura de datos por fecha.

Salidas: OBJECTID, TICKETISSUEDATE, y TOTALPAID en una cola.

Orden Estimado: N + N^(3/2)

R8) Mostrar infracciones ordenadas ascendentemente por VIOLATIONDESC en un rango de hora inicial y final:

Descripción: El código ordena las infracciones ascendentemente dado un rango de tiempo en horas por parámetro. La estructura de datos entregada muestra OBJECTID, TICKETISSUEDATE y VIOLATIONDESC.

Entradas: Rango de hora inicial y final.

Salidas: OBJECTID, TICKETISSUEDATE, y VIOLATIONDESC en una pila.

Orden Estimado: N

R9) Mostrar FINEAMT promedio y desviación estándar para infracciones con un tipo de VIOLATIONDESC:

Descripción: Para una infracción de tipo de VIOLATIONDESC el método recorre toda la estructura de datos y calcula el FINEAMT promedio junto con su desviación estándar.

Entradas: VIOLATIONDESC de las infracciones a analizar.

Salidas: Double con el valor promedio de FINEAMT y Double con su desviación estándar.

Orden Estimado: 2\*N (recorrido doble)

R10) Retornar el número de infracciones en un rango de horas del día:

Descripción: Para un rango de horas especificado se retorna el número de infracciones que ocurrieron en ese intervalo de tiempo.

Entradas: Par de enteros como límites del rango de horas de las infracciones.

Salidas: Entero con número de infracciones.

Orden Estimado: N (recorrido único)

R11) Grafica ASCII porcentaje de infracciones por horas del día:

Descripción: Utilizando un símbolo X, se grafica en ASCII el porcentaje de infracciones por hora del día.

Entradas: -

Salidas: Gráfica ASCII.

Orden Estimado: N (recorrido único)

R12) La deuda total por infracciones que se dieron en un rango de fechas:

Descripción: Para un rango de horas especificado se retorna la deuda toal de infracciones que ocurrieron en ese intervalo de tiempo (TOTALPAID - FINEAMT - PENALTY1 - PENALTY2).

Entradas: Par de enteros como límites del rango de horas de las infracciones.

Salidas: Entero con deuda total de las infracciones.

Orden Estimado: N (recorrido único)

R13) Grafica ASCII deuda acumulada de infracciones por mes:

Descripción: Utilizando un símbolo X, se grafica en ASCII la cantidad de deuda acumulada de todas las infracciones por mes.

Entradas: -

Salidas: Gráfica ASCII.

Orden Estimado: N (recorrido único)