

Objetivo

El objetivo de este proyecto es poner en práctica los conceptos vistos en clase sobre *heaps*, tablas de hash y árboles binarios balanceados.

Contexto

“La accidentalidad vial en Colombia se ha convertido en la segunda manera de muerte violenta y según la Organización Mundial de la Salud (OMS), en el mundo cerca de seis millones de personas mueren por este evento, convirtiéndose así el tráfico automotor en la primera manera de muerte violenta en el nivel mundial.” Medicina Legal, Lesiones no Intencionales.

<http://www.medicinalegal.gov.co/documents/20143/49484/Muertes+Transito.pdf/ad2ae841-ed99-d524-66bf-1b70dec0b44a>

Para prevenir los accidentes viales los gobiernos usan policías de tránsito, cámaras, etc. los cuales producen mucha información. Para que las políticas del gobierno sean más efectivas esta información debe ser tomada en cuenta en la toma de decisiones. Sin embargo el problema es que la información no siempre es accesible al encargado de tomar las decisiones.

En este proyecto se va a procesar la información de infracciones viales de Washington D.C. para producir información en un formato más apropiado para tomar decisiones.

Las Fuentes de Datos

A continuación, se presenta una descripción de las fuentes de datos que se utilizarán en el proyecto. Los datos relacionados se pueden descargar de la página Web: <http://opendata.dc.gov/datasets?q=moving%20violations>.

Específicamente los datos para cada mes del año 2018 están en:

- <http://opendata.dc.gov/datasets/moving-violations-issued-in-january-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-february-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-march-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-april-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-may-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-june-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-july-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-august-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-september-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-october-2018>

- <http://opendata.dc.gov/datasets/moving-violations-issued-in-november-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-december-2018>

Los datos se pueden descargar de cada URL en formato CSV (*Comma separated values*) (sección Datos >> opción Descargar >> Hoja de cálculo). Un archivo CSV es un archivo de texto donde cada línea tiene sus campos de información separados por coma (,). A continuación se muestra un ejemplo de uno de los archivos:

```
OBJECTID,ROW_,LOCATION,ADDRESS_ID,STREETSEGID,XCOORD,YCOORD,TICKETTYPE,FINEAMT,TOTALPAID,PENALTY1,PENALTY2,ACCIDENTINDICATOR,AGENCYID,TICKETISSUEDATE,VIOLATIONCODE,VIOLATIONDESC,ROW_ID
15304300,,DC295      SW      .7      MILES      S/O      EXIT      1
S/B,810381,6993,398406.31,127737.13,Moving,100,0,0,,No,25,2018-12-01T08:25:00.000Z,T119,SPEED 11-15 MPH OVER THE SPEED LIMIT,
15304301,,NEW      YORK      AVE      W/B      @      NEW      JERSEY
NW,808095,6936,398718.3825,137500.5723,Moving,150,0,0,,No,25,2018-12-01T21:24:00.000Z,T113,FAIL TO STOP PER REGULATIONS FACING RED SIGNAL,
15304302,,NEW      YORK      AVE      W/B      @      NEW      JERSEY
NW,808095,6936,398718.3825,137500.5723,Moving,150,0,0,,No,25,2018-12-02T11:15:00.000Z,T113,FAIL TO STOP PER REGULATIONS FACING RED SIGNAL,
15304303,,BLOCK      100      NEW      YORK      AVE      NW
WESTBOUND,814461,12462,398891.13,137577.22,Moving,30,0,0,,No,2,2018-12-27T00:00:00.000Z,T333,FAIL TO DISPLAY PROOF OF VEHICLE INSURANCE,
```

Los atributos son:

- OBJECTID: Identificador único de la infracción.
- ROW_:
- LOCATION: Dirección en formato de texto.
- ADDRESS_ID: ID de la dirección.
- STREETSEGID: ID del segmento de la calle.
- XCOORD: Coordenada X donde ocurrió (No corresponde a longitud geográfica).
- YCOORD: Coordenada Y donde ocurrió (No corresponde a latitud geográfica).
- TICKETTYPE:
- FINEAMT: Cantidad a pagar por la infracción USD.
- TOTALPAID: Cuanto dinero efectivamente pagó el que recibió la infracción en USD.
- PENALTY1: Dinero extra que debe pagar el conductor.
- PENALTY2: Dinero extra que debe pagar el conductor.
- ACCIDENTINDICATOR: Si hubo un accidente o no.
- AGENCYID:
- TICKETISSUEDATE: Fecha cuando se puso la infracción.
- VIOLATIONCODE: código de la infracción.
- VIOLATIONDESC: descripción textual de la infracción.
- ROW_ID:

Carga de Información

Para responder a los requerimientos presentados más adelante, usted deberá cargar la información de todos los archivos .CSV correspondientes a uno de los semestres del año 2018.

El semestre lo podrá seleccionar el usuario.

Solo es permitido leer una vez la información de los archivos.

Al final de la carga hay que reportar:

- El número de infracciones de cada mes cargado,
- El total de infracciones del semestre,
- La zona geográfica Minimax que contiene las infracciones. El Minimax define la zona rectangular con las coordenadas (Xmin, Ymin) y (Xmax, Ymax) que contiene todas las infracciones. Xmin es el valor mínimo en X para las infracciones; Xmax es el valor máximo en X para las infracciones; Ymin es el valor mínimo en Y para las infracciones; Ymax es el valor máximo en Y para las infracciones.

Impresión en consola de muchos datos

En algunos de los requerimientos el conjunto de datos retornado puede ser muy grande, por lo tanto se imprimirán solo los primeros N datos, donde N es una constante que inicialmente se puede igualar a 20.

Requerimientos - Parte A (estudiante 1 de cada grupo)

1A- Obtener el ranking de las N franjas horarias que tengan más infracciones. El valor N es un dato de entrada. Se define las franjas horarias válidas:

00:00:00 - 00:59:59

01:00:00 - 01:59:59

02:00:00 - 02:59:59

...

21:00:00 - 21:59:59

22:00:00 - 22:59:59

23:00:00 - 23:59:59

Caso Especial: 24:00:00 equivale a 00:00:00

Para cada franja horaria hay que dar el intervalo que cubre la franja, el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente y el valor total a pagar por las infracciones.

Se requiere usar una Cola de Prioridad orientada a mayor por el número de infracciones en cada franja horaria.

2A- Realizar el ordenamiento de las infracciones por Localización Geográfica (Xcoord, Ycoord).

La llave de búsqueda es la tupla (Xcoord, Ycoord): Primero se tienen en cuenta el valor Xcoord. Si hay valores de Xcoord iguales, el criterio de ordenamiento lo define el valor Ycoord. Todas las infracciones con la llave (Xcoord, Ycoord) deben agruparse bajo un mismo objeto.

Realizar el ordenamiento de las infracciones en una Tabla de Hash por localización geográfica.

Consulta: Buscar las infracciones por localización geográfica (Xcoord, Ycoord) en la Tabla de Hash. Para la localización geográfica de consulta, se requiere informar el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, el valor total a pagar por las infracciones, el location, el Id de la dirección (AddressID) y el Id de la calle (StreetSegId).

3A- Buscar las infracciones por rango de fechas [Fecha Inicial (Año/Mes/Día), Fecha Final (Año/Mes/Día)].

Representar las infracciones por fecha (Año/Mes/Día) en un Árbol Balanceado por Fecha (TicketIssueDate). Para cada posible fecha en el rango definido con al menos una infracción, se requiere informar la fecha (Año/Mes/Día), el total de infracciones, su porcentaje de infracciones sin accidente, su porcentaje de infracciones con accidente y el valor total a pagar por las infracciones. Las fechas deben informarse en orden ascendente (cronológicamente).

Parte B (estudiante 2 de cada grupo)

1B- Obtener el ranking de las N tipos de infracción (ViolationCode) que tengan más infracciones. El valor N es un dato de entrada.

En el ranking deben aparecer los tipos de infracción de mayor a menor número de infracciones. Para cada tipo de infracción debe mostrarse su código, total de infracciones, su porcentaje de infracciones sin accidente, su porcentaje de infracciones con accidente y el valor total a pagar por las infracciones. Se requiere usar una Cola de Prioridad orientada a mayor por el número de infracciones de cada tipo de infracción.

2B- Realizar el ordenamiento de las infracciones por Localización Geográfica (Xcoord, Ycoord).

La llave de búsqueda es la tupla (Xcoord, Ycoord): Primero se tienen en cuenta el valor Xcoord. Si hay valores de Xcoord iguales, el criterio de ordenamiento lo define el valor

Ycoord. Todas las infracciones con la llave (Xcoord, Ycoord) deben agruparse bajo un mismo objeto.

Realizar el ordenamiento de las infracciones en un Árbol Balanceado por localización geográfica.

Consulta: Buscar las infracciones por localización geográfica (Xcoord, Ycoord) en el Arbol Balanceado. Para la localización geográfica de consulta, se requiere informar el total de infracciones, su porcentaje de infracciones sin accidente, su porcentaje de infracciones con accidente, el valor total a pagar por las infracciones, el location, el Id de la dirección (AddressID) y el Id de la calle (StreetSegId).

3B- Buscar las franjas de fecha-hora donde se tiene un valor acumulado de infracciones en un rango dado [US\$ valor inicial, US\$ valor final].

Usar una representación de Árbol Balanceado por el *valor acumulado de infracciones* para cada fecha-hora. Es decir, para cada fecha en el semestre seleccionado, se debe calcular el valor acumulado (US\$) de las infracciones para cada franja de hora completa (00:00:00 – 00:59:59, 01:00:00 – 01:59:59, ..., 23:00:00 – 23:59:59).

Como resultado de este requerimiento se debe mostrar las fechas – hora que tienen un valor acumulado de infracciones (US\$) en el rango de consulta suministrado. Es importante resaltar que el resultado debe mostrarse en orden ascendente por valor acumulado de infracciones y que puede haber mas de una fecha-hora que tenga un mismo valor acumulado de infracciones. En el resultado de cada fecha-hora solución se debe mostrar: su valor acumulado de infracciones (US\$), el total de infracciones, su porcentaje de infracciones sin accidente y su porcentaje de infracciones con accidente.

Parte C (trabajo en grupo)

Para esta parte defina las estructuras más convenientes para cada punto de forma que se logre la mejor eficiencia en tiempo y en espacio. Justifique su elección.

1C- Obtener la información de una localización dada (AddressID). Para la localización de consulta (AddressID), se requiere informar el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, el valor total a pagar por las infracciones, y el Id de la calle (StreetSegId).

2C- Obtener las infracciones en un rango de horas [HH:MM:SS inicial, HH:MM:SS final]. Para el rango de horas de consulta, se requiere dar:

Información general: el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, y el valor total a pagar por las infracciones.
Información por código (ViolationCode): Para las infracciones resultantes, hay que agruparlas por código. Informar el detalle de código y el total de infracciones.

3C – Obtener el ranking de las N localizaciones geográficas (Xcoord, Ycoord) con la mayor cantidad de infracciones. El valor N es un dato de entrada. Por cada localización (Xcoord, Ycoord) hay que reportar el total de infracciones, su porcentaje de infracciones sin accidente, su porcentaje de infracciones con accidente, el location, el Id de la dirección (AddressID) y el Id de la calle (StreetSegId).

4C – Mostrar una gráfica ASCII con la información de las infracciones por código (ViolationCode). La gráfica debe mostrar los tipos de infracción ordenados de mayor a menor por su número de infracciones. Se espera mostrar una representación ASCII de la forma:

Reporte Agregado de Infracciones por Código 2018

TXXX | *****

TYYY | *****

TZZZ | *****

...

Cada * representa un porcentaje K% (a definir). En la gráfica debe indicarse el porcentaje (%) al que corresponde cada *.

Restricciones

- Los datos contenidos en los archivos sólo se pueden leer una vez
- Se deberá trabajar en Java 8
- El proyecto se debe implementar en Eclipse
- La entrada/salida de información adicionales se debe realizar por consola
- **No usar las colecciones del API Java.**

Entrega de Diseño (33% Nota del Proyecto)

- Fecha/Hora límite de entrega: 24 Marzo, 11:59 p.m.
- Repositorio Bitbucket/GitHub con el nombre de la forma Proyecto_2_201910_sec_Y_team_Z (reemplazar Y por el número de la sección de su curso y Z su número de grupo)
- Verificar que el repositorio tiene configurado como usuarios a los monitores y al profesor con acceso de lectura.
- Entregables:
 - Documento con:
 - Documentación de los requerimientos funcionales (incluyendo los datos de entrada, la descripción, los datos de salida y estimación de complejidad temporal de cada requerimiento funcional).

Nota: Mencionar y justificar las estructuras de datos utilizadas para los requerimientos de la parte C (trabajo en grupo).

- Diseño de las Estructuras de Datos a utilizar (diagrama de clases UML, imagen)
- Diseño de la Solución al proyecto (diagrama de clases UML, imagen)
- Proyecto Eclipse/Java con:
 - Implementación y Pruebas Unitarias Automáticas de las Estructuras de Datos (Eclipse/Java)
 - Lectura/carga de los datos para un semestre dado por el usuario (mostrar la evidencia que los datos fueron leídos/cargados)
- Hacer el desarrollo del proyecto en la rama (*branch*) master del repositorio. Como parte final de esta entrega, crear la rama (*branch*) entrega-diseNo. Verificar que en esta rama queda la copia de los entregables del proyecto. Los entregables en la rama entrega-diseNo No deben actualizarse después de la fecha/hora límite de esta entrega.

Entrega Final (67% Nota del Proyecto)

- Fecha/Hora límite de entrega: 14 Abril, 11:59 p.m.
- Repositorio Bitbucket/GitHub Proyecto_2_201910_sec_Y_team_Z (reemplazar Y por el número de la sección de su curso y Z su número de grupo) creado para la entrega de diseño. El proyecto completo (documentación e implementación) debe estar accesible en la rama (*branch*) master.
- Entregables:
 - Proyecto Eclipse/Java con:
 - Implementación completa de los requerimientos funcionales
- Los entregables en la rama master No pueden tener fecha de actualización posterior a la fecha/hora límite de esta entrega