Objetivo

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

Las Fuentes de Datos

A continuación, se presenta una descripción de las fuentes de datos que se utilizarán en el proyecto.

1. Mapa de la malla vial de Bogotá

    - Nodes_of_red_vial-wgs84_shp.txt
      Formato CSV:

      ```
      0,-74.08921298299998,4.582989396000016
      1,-74.08952746199998,4.582560966000017
      2,-74.093892202,4.576679366000008
      3,-74.09408026199998,4.576433936999991
      4,-74.09451399300002,4.57581709599998
      ..
      ```

      Cada línea tiene primero el id del nodo, seguido de la longitud y la latitud correspondiente.

2. Datos de Uber que se pueden descargar en https://movement.uber.com/?lang=es-CO.

- bogota-cadastral-2018-X-All-HourlyAggregate.csv
  Archivos CSV donde X indica el trimestre, el cual puede ser 1, 2, 3 o 4. Tiene los siguientes atributos:
    - `sourceid:` el id de la zona de origen
    - `dstid:` el id de la zona destino
    - `hod:` es la hora del dia, 0 es 12 de la noche , 1 es 1am, … , 23 es 11pm
    - `mean_travel_time:` el tiempo promedio de viaje entre las dos zonas
    - `standard_deviation_travel_time:` la desviación estándar de los tiempos de viaje

- bogota-cadastral-2018-X-All-MonthlyAggregate.csv

Archivos CSV donde X indica el trimestre, el cual puede ser 1, 2, 3 o 4. Tiene los siguientes atributos:

- `sourceid:` el id de la zona de origen
- `dstid:` el id de la zona destino
- `month:` es el mes del año. 1 es Enero,... y 12 es Diciembre.
- `mean_travel_time:` el tiempo promedio de viaje entre las dos zonas
- `standard_deviation_travel_time:` la desviación estándar de los tiempos de viaje

- bogota-cadastral-2018-X-WeeklyAggregate.csv
  Archivos CSV donde X indica el trimestre, el cual puede ser 1, 2, 3 o 4. Tiene los siguientes atributos:
  - `sourceid:` el id de la zona de origen
  - `dstid:` el id de la zona destino
  - `dow:` el dia de la semana, 1 es Domingo, 2 Lunes, ... y 7 es Sábado
  - `mean_travel_time:` el tiempo promedio de viaje entre las dos zonas
  - `standard_deviation_travel_time:` la desviación estándar de los tiempos de viaje

- bogota_cadastral.json
  Archivo JSON con todas las zonas que define Uber.
  Formato:

```
{
"type":"FeatureCollection",
"features":[
{
"type":"Feature",
"geometry":
       {"type":"MultiPolygon","coordinates":[[[[-74.200295,4.617249],[-
       74.200285,4.617248],[-74.200277,4.617248],[-74.200257,4.617246] …]]],
"properties":{
       "cartodb_id":12,
       "scacodigo":"004575",
       "scatipo":0,
       "scanombre":"LOS LAURELES",
       "shape_leng":0.02774133557,
       "shape_area":0.00003682838,
       "MOVEMENT_ID":"1",
       "DISPLAY_NAME":"LOS LAURELES, 004575 (1)"}
},{
...
},
...
]
}
```

En la colección de features, existe un atributo geometry con los puntos (cada uno representado como un arreglo con longitud y latitud) que forman la frontera de la zona, y un atributo properties con las propiedades de la zona. Cada zona tiene:

- MOVEMENT_ID: Id de la zona.
- scanombre: Nombre de la zona.
- shape_leng: perímetro de la zona. Si se multiplica por 100 da en kilómetros.
- shape_area: área de la zona. Si se multiplica por 10,000 da en kilómetros cuadrados.

IMPORTANTE: La propiedad MOVEMENT_ID de una zona corresponde a los atributos sourceid y dstid en los viajes.

Para la lectura de archivos JSON se recomienda usar el API (librería) GSON. Consultar https://github.com/google/gson Para descargas del API (librería extensión .jar) consulte: https://maven-badges.herokuapp.com/maven-central/com.google.code.gson/gson Para documentación del API consulte: http://www.javadoc.io/doc/com.google.code.gson/gson

Carga de Información

Para responder a los requerimientos presentados más adelante, usted deberá cargar todos los archivos que considere pertinentes mencionados en la sección sobre las fuentes de datos. Las preguntas son sobre los primeros 6 meses del año 2018.
Solo es permitido leer una vez la información de los archivos.
Al final de la carga hay que reportar:

- El número de viajes que se cargaron de cada archivo CSV
- El número de zonas que se cargaron del archivo JSON
- El número de nodos (esquinas) de la malla vial del archivo TXT

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.

Diseño

Para cada requerimiento se quiere usar la estructuras más convenientes para que se logre la mejor eficiencia en tiempo y en espacio. En cada uno de esos casos usted es libre de usar la estructura que considere apropiada pero debe justificar su elección.

Requerimientos - Parte A (estudiante 1 de cada grupo)

Restricciones para 1A, 2A y 3A:

- **Al menos 1 de los 3 requerimientos se debe realizar con una cola de prioridad**
- **Al menos 1 de los 3 requerimientos se debe realizar con una tabla de hash**
- **Al menos 1 de los 3 requerimientos se debe realizar con un árbol balanceado**

1A- Obtener las N letras más frecuentes por las que comienza el nombre de una zona (No diferenciar las mayúsculas de las minúsculas). N es un dato de entrada.
El resultado debe aparecer de mayor a menor. Para cada letra se debe imprimir la letra y el nombre de las zonas que comienzan por esa letra.

2A- Buscar los nodos que delimitan las zonas por Localización Geográfica (latitud, longitud).
Dadas una latitud y una longitud, se deben mostrar todos los nodos en la frontera de las zonas que tengan la misma latitud y longitud truncando a las primeras 3 cifras decimales. Por ejemplo:
(-4.1234, 74.9876) == (-4.1234, 74.9876)
(-4.1234, 74.9876) == (-4.1239, 74.9872)
(-4.1234, 74.9876) != (-4.1229, 74. 9876)
(-4.1234, 74.9876) != (-4. 1234, 74. 9866)
(-4.1, 74.9) != (-4, 74)

Los nodos son del archivo de zonas. Se debe mostrar el número de nodos retornados y de cada nodo su latitud, longitud y nombre de la zona a la que pertenece.

3A- Buscar los tiempos promedio de viaje que están en un rango y que son del primer trimestre del 2018.
Dado un rango de tiempos promedio de viaje en segundos [limite_bajo, limite_alto], retornar los viajes cuyo tiempo promedio mensual esté en ese rango.
Se debe mostrar únicamente N viajes ordenados por zona de origen y zona de destino. Por cada viaje se debe mostrar su zona de origen, zona de destino, mes y tiempo promedio mensual del viaje.

Parte B (estudiante 2 de cada grupo)

Restricción para 1B, 2B y 3B:

- **Al menos 1 de los 3 requerimientos se debe realizar con una cola de prioridad**
- **Al menos 1 de los 3 requerimientos se debe realizar con una tabla de hash**
- **Al menos 1 de los 3 requerimientos se debe realizar con un árbol balanceado**

1B- Buscar los N zonas que están más al norte. N es un valor de entrada. Una zona A esta más al Norte que una zona B si algún punto de la frontera de A está más al norte que todos los puntos de la frontera de B.
Mostrar las zonas ordenadas desde las que estén más al norte. De cada zona se debe imprimir el nombre y la (latitud, longitud) de su punto más al norte.
NOTA: La latitud define la localización geográfica hacia el norte o hacia el sur en un mapa.

2B- Buscar nodos de la malla vial por Localización Geográfica (latitud, longitud).
Dado una latitud y una longitud, se deben mostrar todos los nodos que tengan esas mismas latitud y longitud truncando a 2 cifras decimales. Por ejemplo:
(-4.1234, 74.9876) == (-4.1235, 74.9876)
(-4.1234, 74.9876) == (-4.1239, 74.9872)
(-4.1234, 74.9876) != (-4.1300, 74. 9876)
(-4.1234, 74.9876) != (-4. 1277, 74. 9299)
(-4.1, 74.9) != (-4, 74)

Los nodos son del archivo de la malla vial. Se debe mostrar el número de nodos retornados y de cada nodo su id, latitud y longitud.

3B- Buscar los tiempos de espera que tienen una desviación estándar en un rango dado y que son del primer trimestre del 2018.
Dado un rango de desviaciones estándares [limite_bajo, limite_alto] retornar los viajes cuya desviación estándar mensual este en ese rango.

Se debe mostrar únicamente N viajes ordenados por zona de origen y zona de destino. De cada viaje se debe mostrar la zona de origen, zona de destino, mes y la desviación estándar del viaje.

Parte C (trabajo en grupo)

1C- Retornar todos los tiempos de viaje promedio que salen de una zona dada y a una hora dada.
Dados el Id de una zona de salida y una hora que son ingresados por el usuario, retornar los tiempos de viaje promedio con esas características.
Se debe mostrar la zona de origen, zona de destino, hora y tiempo promedio de cada viaje.

2C- Retornar todos los tiempos de viaje que llegan de una zona dada y en un rango de horas.
Dado el Id de una zona de llegada y un rango de horas que ingresa el usuario, mostrar todos los tiempos de viaje promedio que cumplan esos criterios.

Se debe mostrar la zona de origen, zona de destino, hora y tiempo promedio de cada viaje.

3C – Obtener las N zonas priorizadas por la mayor cantidad de nodos que definen su frontera. El valor N es un dato de entrada. Por cada zona se debe mostrar el nombre de la zona y el número de nodos que definen su frontera.

4C – Gráfica ASCII - Porcentaje de datos faltantes para el primer semestre 2018. En los datos por horas se considera que cada zona de origen debe tener los tiempos de viaje hacia todas las otras zonas en todas las horas y en cada trimestre. Por ejemplo, si hubiera 2 zonas (1 y 2), se espera que hayan 96 viajes que salen de la zona 1 durante el primer semestre 2018:

sourceid	dstid	trimestre	hod (hora del dia)	mean_travel_time
1	1	1	00	45
1	1	1	01	56
1	1	1	...	...
1	1	1	23	40
1	1	2	00	20
1	1	2	01	22
1	1	2	...	...
1	1	2	23	25
1	2	1	00	45
1	2	1	01	56
1	2	1	...	...
1	2	1	23	40
1	2	2	00	20
1	2	2	01	22
1	2	2	...	...
1	2	2	23	25

Crear una gráfica que muestre por cada zona de origen que porcentaje de datos faltan (un dato faltante indica que no hubo ningún viaje desde la zona de origen a la zona destino a una hora dada en un trimestre dado) usando el siguiente formato ASCII:

Porcentaje de datos faltantes por zona
1 | **********
2 | **********
3 | **********
…

Cada * representa un 2% .

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: 8 de Octubre, 11:59 p.m.
- Repositorio Bitbucket/GitHub con el nombre de la forma Proyecto_2_201920_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:
    - 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 el semestre (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: 29 de Octubre, 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