

```

/*****
/*          ASU CSE 512 Project Phase 1 Fall 2017          */
/*          Group: ASU DataDevils                          */
/*
/*          Members                                         */
/*          Nishi Shah                                     */
/*          Kruthika Surineni                             */
/*          Richard Tuznik                                 */
/*          Vamsi Krishna Godavarthi                      */
*****/

```

```

/*****

```

Youtube video link: <https://youtu.be/G2sKrPhtyFs>

```

/*****
/*****
*          Systems:                                         */
/*          Master: 10.152.93.149                          */
/*          Worker1: 10.144.221.111                       */
/*          Worker2: 10.143.3.226                         */
*****/

```

```

/*****
/*          Code - Command to Load data to HDFS          */
*****/

```

1) ./hadoop fs -copyFromLocal -f /home/ubuntu/Downloads/datasets/arealm.csv
hdfs://master:54310/arealm2.csv

2) ./hadoop fs -copyFromLocal -f /home/ubuntu/Downloads/datasets/zcta510.csv
hdfs://master:543102) /zcta510_2.csv

```

/*****

```

```
/*-----*/
/*                               Code - Queries to be executed                               */
/*-----*/
```

```
import org.apache.spark.SparkContext
import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel;
val conf = new SparkConf().setAppName("Simple
Application").setMaster("spark://10.152.93.149:7077");
val sc=getOrCreate();
val points="hdfs://master:54310/arealm2.csv"
val rectangle="hdfs://master:54310/zcta510_2.csv"
```

//Question 2(a)

```
/*----- Start an example Spatial Range Query without Index -----*/
```

```
import org.datasyslab.geospark.spatialOperator.RangeQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.Envelope;
import org.datasyslab.geospark.enums.FileDataSplitter;

val queryEnvelope=new Envelope (-113.79,-109.73,32.99,35.08);
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
val resultSize = RangeQuery.SpatialRangeQuery(objectRDD, queryEnvelope, false, false).count();

/*----- End an example Spatial Range Query without Index -----*/
```

//Question 2(b)

```
/*----- Start an example Spatial Range Query with Index -----*/
```

```
import org.datasyslab.geospark.spatialOperator.RangeQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.Envelope;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.IndexType;

val queryEnvelope=new Envelope (-113.79,-109.73,32.99,35.08);
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
objectRDD.buildIndex(IndexType.RTREE,false);
val resultSize = RangeQuery.SpatialRangeQuery(objectRDD, queryEnvelope, false, true).count();

/*----- End an example Spatial Range Query with Index -----*/
```

//Question 3(a)

```
/*----- Start an example Spatial KNN Query without Index -----*/
import org.datasyslab.geospark.spatialOperator.KNNQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.GeometryFactory;
import com.vividsolutions.jts.geom.Point;
import com.vividsolutions.jts.geom.Coordinate;
import org.datasyslab.geospark.enums.FileDataSplitter;

val fact=new GeometryFactory();
val queryPoint=fact.createPoint(new Coordinate(35.08,-113.79));
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
val resultSize = KNNQuery.SpatialKnnQuery(objectRDD, queryPoint, 5,false).size();

/*----- End an example Spatial KNN Query without Index -----*/
```

//Question 3(b)

```
/*----- Start an example Spatial KNN Query with Index -----*/
import org.datasyslab.geospark.spatialOperator.KNNQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.GeometryFactory;
import com.vividsolutions.jts.geom.Point;
import com.vividsolutions.jts.geom.Coordinate;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.IndexType;

val fact=new GeometryFactory();
val queryPoint=fact.createPoint(new Coordinate(35.08,-113.79));
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
objectRDD.buildIndex(IndexType.RTREE,false);
val resultSize = KNNQuery.SpatialKnnQuery(objectRDD, queryPoint, 5,true).size();

/*----- End an example Spatial KNN Query with Index -----*/
```

//Question 4(a)

```
/*----- Start an example Spatial Join Query without Index -----*/
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;

val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
objectRDD.spatialPartitioning(GridType.EQUALGRID);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery(objectRDD,rectangleRDD,false,false).count();

/*----- End an example Spatial Join Query without Index -----*/
```

//Question 4(b)

```
/*----- Start an example Spatial Join Query with Index -----*/
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;
import org.datasyslab.geospark.enums.IndexType;

val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false);
objectRDD.spatialPartitioning(GridType.EQUALGRID);
objectRDD.buildIndex(IndexType.RTREE,true);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery(objectRDD,rectangleRDD,true, false).count();

/*----- End an example Spatial Join Query with Index -----*/
```

//Question 4(c)

```
/*----- Start an example Spatial Join Query without Index -----*/
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;

val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
objectRDD.spatialPartitioning(GridType.RTREE);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery(objectRDD,rectangleRDD,false,false).count();
/*----- End an example Spatial Join Query without Index -----*/

/*****/
```