NAME: Xiang Xu

605.741.31
Module 12 Quiz

Suppose we have a Z-Order Curve with interspersed latitudes and longitudes such that a point with latitude ***33.333*** and longitude ***55.555*** is represented by rowID ***3535.353535.*** Let us also say that latitude 90S (south pole) is represented by **000.000** degrees, the equator is represented by **90.000** degrees, and 90N (the north pole) is represented by **180.000** degrees. Let us say that longitude 180W is represented by **000.000** degrees, 90W is represented by **90.000** degrees, 0 is represented by **180.000** degrees, 90E is represented by **270.000** degrees, and 179.999E is represented by **359.999** degrees. Suppose further that the Accumulo table represents the name of the tourist site at that latitude and longitude. A sample schema for the Accumulo table that represents the Eiffel Tower in France (**138.858**, **182.295**) and the La Moneda Palace in Chile (**56.557**, **109.346**) is:

rowID	columnFamily	columnQual	time	value
113882.825985	site	Eiffel Tower		
015069.535476	Site	La Moneda Palace		

Write the pseudocode for the Map and Reduce methods that find all points within a latitude-longitude bounding box. Remember that the bounds of the longitude may cross the 0 degree meridian. Part of the grade will be dependent on the efficiency of your MapReduce pseudocode. Remember that the rowID has an interleaved format for efficiency reasons.

1) Map pseudocode

```
/*
* key is [rowID],
* value is [column family, column qualifier, time, value]
*/

Map(key, value){
        //not cross 0 degree meridian
        if(even position in LowerBound < even position in HigherBound
        && odd position in key > odd position in LowerBound
        && odd position in key < odd position in HigherBound
        && even position in key > even position in LowerBound
        && even position in key < even position in HigherBound)
        Emit (key, value.columnQual)

        // cross 0 degree meridian
        if(even position in LowerBound > even position HigherBound
        && odd position in key > odd position in LowerBound
```

```
        && odd position in key < odd position in HigherBound
        && 359.999 > even position in key > even position in LowerBound
        && 0 < even position in key < even position in HigherBound)
        Emit (key, value.columnQual)

}
```

2) Reduce pseudocode

```
Reduce(key,value){
For each row j {
   Emit(value)
   }


}
```