Assignment 2

Map Reduce in Spark

Question 2: Kth Percentile

In this first I have used two broadcast variables

1. N (number of elements in the list)
2. Hasher (floor of (1 % of the sum of all elements))

The Hasher will divide the list in 100 lists such that list number i will contain elements which are smaller than the elements in the list number j. if i<j

For this I have used 2 sets of mapper and reducer

Mapper 1: flatMapToPair

This will use the hasher value to create keys according to the input value and add a count to the value of the keys which are greater than the current key .

For eg. for value = 5 and hasher =2

Since, 5/2 => 2

A tuple will be created (2,(5,0))

And An array of tuples will also be created (3,(-1,1)),(4,(-1,1)) and so on

The output format of this phase is:

```
JavaPairRDD<Integer,Tuple2<Integer,Integer>> mappedOne
```

Reducer 1: GroupedByKey + MapValues

Now, this will filter the tuples based on the -1 part in the value section

```
for (Tuple2<Integer, Integer> tuple2 : tuple2s) {
   count+=tuple2._2;
   if (tuple2._1!=-1)
   list.add(tuple2._1);
}
```

The output format is :

```
JavaPairRDD<Integer,Tuple2<Iterable<Integer>,Integer>>
```

Mapper 2 : FlatMapToPair

This, will first find the positions of 25th ,50th and 75th percentile and if there are two positions then we will find the two positions and map them to the same key.

Now,

```
for (int i = 0; i < 6; i++) {
   if (k[i] < count + integerTuple2Tuple2._2._2 && k[i] >
integerTuple2Tuple2._2._2 && k[i] != -1) {
      num =
(double)Select.quickselect(integerTuple2Tuple2._2._1,
(int)k[i]-integerTuple2Tuple2._2._2);
      Integer key= (int) floor(k[i]/2);
      list.add(new Tuple2<Integer,Double>(key,  num));
}
}
```

With the help of above code I will those 6 lists which will contain the given positions and apply quick select on those lists to find the element

The format of this output is :

```
JavaPairRDD<Integer,Double> mappedTwo
```

Reducer 2: ReduceByKey

This will just average the values on same keys

```
public Double call(Double aDouble, Double aDouble2) throws
Exception {
    return (aDouble+aDouble2)/2;
}
```

The format of the output of this phase is :

```
JavaPairRDD<Integer,Double>
```

The rest of the map reduce is just to create the output in given format