3)Which industry(SOC\_NAME) has the most number of Data Scientist positions?

package h1b;

import java.io.IOException;

import java.util.TreeMap;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class question3 {

public static class Mymapper extends Mapper<LongWritable,Text,Text,IntWritable>{

public void map(LongWritable key,Text value,Context context) throws IOException, InterruptedException{

String Line =value.toString();

String[] arr =Line.split("\t");

int count=0;

if(arr[4].toString().contains("DATA SCIENTIST"))

{

count++;

context.write(new Text(arr[3]), new IntWritable(count));

}

}

}

public static class Myreducer extends Reducer<Text,IntWritable,Text,IntWritable>

{

int max\_job =0;

String kk="";

public void reduce(Text key,Iterable<IntWritable> values,Context context) throws IOException, InterruptedException{

//String k=key.toString();--

int count=0;

for(IntWritable a:values){

count+=a.get();

if (max\_job < count){

max\_job=count;

kk=key.toString();

}

//context.write(new Text(kk), new IntWritable(max\_job));

}

}

public void cleanup(Context context) throws IOException, InterruptedException

{

context.write(new Text(kk),new IntWritable(max\_job));

}

}

public static void main(String args[]) throws ClassNotFoundException, IOException, InterruptedException{

Configuration conf=new Configuration();

Job job=Job.getInstance(conf,"work");

job.setJarByClass(question3.class);

job.setMapperClass(Mymapper.class);

job.setReducerClass(Myreducer.class);

//job.setCombinerClass(Myreducer.class);

//job.setMapOutputKeyClass(Text.class);

//job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileSystem.get(conf).delete(new Path(args[1]),true);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job,new Path(args[1]));

System.exit(job.waitForCompletion(true)?0:1);

}

}