//访问最多的网站 //map给reduce传入的Key应该是网站  
  
  
package //linux文件夹位置;  
  
import java.io.IOException;  
import java.util.\*;  
import java.net.URL;  
  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.IntWritable;  
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;  
import org.apache.hadoop.util.GenericOptionsParser;  
  
public class hadoopop {  
 public static class Map extends Mapper<Object, Text, Text, Text>{  
  
 public void map(Object key, Text value, Context context  
 ) throws IOException, InterruptedException {  
 String line = value.toString();  
 String weburl = new String();  
 String another = new String();  
 int leng = line.length();  
 int len = (leng-1);  
 while(line.charAt(len)!=' '){  
 len--;  
 }  
 String[] values = {line.substring(0,len),line.substring(len+1)};  
 if(values[0].compareTo("\*填上\*")!=0){ //if not first line in  
 weburl = values[0];  
 another = values[1];  
 }  
 context.write(new Text(weburl),new Text(another)); // put to reduce  
 }  
 }  
  
 public static class IntSumReducer  
 extends Reducer<Text,Text,Text,Text> {  
  
 public void reduce(Text key, Iterable<Text> values,  
 Context context  
 ) throws IOException, InterruptedException {  
   
 int time = 0;  
 if(time == 0){  
 context.write(new Text("网站"),new Text("访问数量"));  
 time++;  
 }  
  
 int x=0;  
   
 for(Text t:values){  
 x++; //count  
 }  
   
 context.write(key,new Text(x));  
  
 }  
 }  
  
 public static void main(String[] args) throws Exception {  
 Configuration conf = new Configuration();  
 String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();  
 if (otherArgs.length != 2) {  
 System.*err*.println("Usage: wordcount <in> <out>");  
 System.*exit*(2);  
 }  
  
 Job job = new Job(conf, "count");  
 job.setJarByClass(gradecount.class);  
  
 // 设置Map和Reduce处理类  
 job.setMapperClass(Map.class);  
 job.setReducerClass(IntSumReducer.class);  
  
 // 设置输出类型  
 job.setOutputKeyClass(Text.class);  
 job.setOutputValueClass(Text.class);  
  
 // 设置输入和输出目录  
 FileInputFormat.addInputPath(job, new Path(otherArgs[0]));  
 FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));  
 System.*exit*(job.waitForCompletion(true) ? 0 : 1);  
 }  
}