CS5542 Big Data Apps and Analytics

LAB ASSIGNMENT #2

1. **Spark Programming:**

Write a spark program with an interesting use case using text data as the input and program should have at least Two Spark Transformations and Two Spark Actions.

Present your use case in map reduce paradigm as shown below (for word count).

Use Case:

The written project takes text data as input and generates a file with term frequencies with highest repeated word in 1st line and so on. It also prints the top 10 TF (Term Frequency) words in the text data supplied.

Input Text :

Every 8 seconds or so, a developer asks a question on Stack Overflow. This year, 56,033 coders in 173 countries answered the call.

We asked them 45 questions. Key highlights include the following:

Developers love Rust. Even back-end developers know JavaScript. Only 7% of developers identify as "rockstars". Most developers prefer dogs to cats. (But not developers in Germany.)

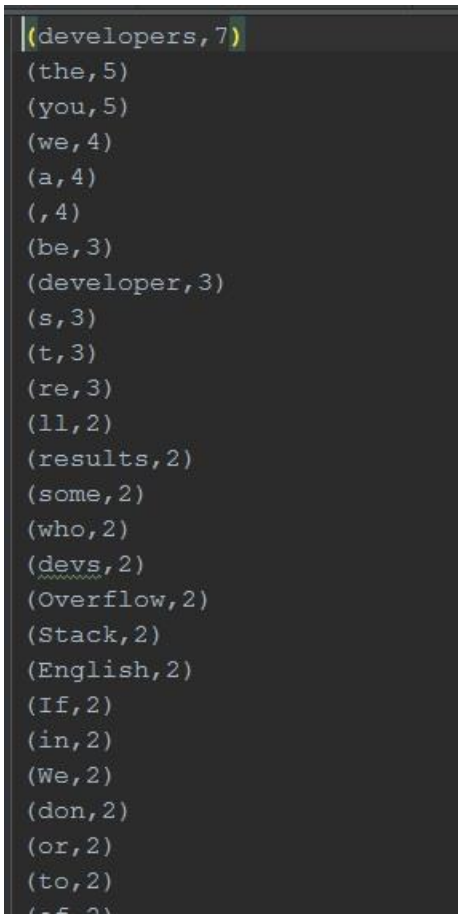
Surveys aren’t perfect. While our large sample size helps offset some biases, it’s still biased against devs who don't speak English, or who don't like taking English-language surveys. In some sections we've augmented the results with insights gleaned from the activity of Stack Overflow's 40 million monthly visitors. If you’re an employer, we’d be happy to help you reach those developers. If you're a developer (you're probably a developer), we hope you sign up.

Throughout these results we'll be using the terms "developers", "devs", and "respondents" interchangeably. We’ll also be keeping commas outside quotation marks, because that’s what developers do.

Code :

```
package com.example.spark.demo

import org.apache.spark.{SparkConf, SparkContext}

/**
  * Created by vikesh on 2/1/2017.
  */
object labassignment2 {

  def main(args: Array[String]) {

    val conf = new SparkConf().setAppName("wordCount") .set("spark.eventLog.enabled",
"true")
    System.setProperty("hadoop.home.dir", "C:/Users/Vikesh/Documents/UMKC
Subjects//PB/hadoopforspark/")

    val inputFile = "data/textfile.txt"

    val outputFile = "data/wordcount"

    val sc = new SparkContext(conf)

    val input =  sc.textFile(inputFile)

    val words = input.flatMap(line => line.split("\\W+"))
    words.foreach(f=>println(f))

    val counts = words.map(words => (words, 1)).reduceByKey{case (x, y) => x + y}

    val tf=counts.sortBy(f=>f._2,ascending = false)
    tf.foreach(f=>println(f))
    tf.saveAsTextFile(outputFile)

    tf.take(10).foreach(f=>println(f))
    sc.stop()

  }

}
```

Output :



Top TF words :

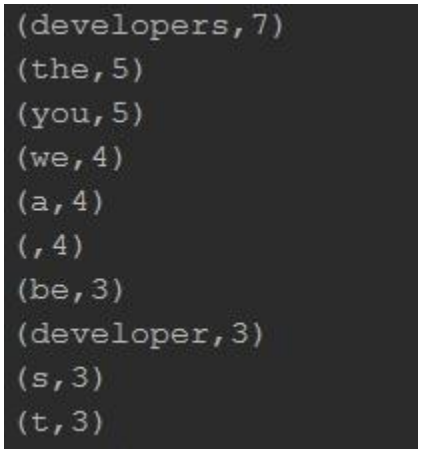
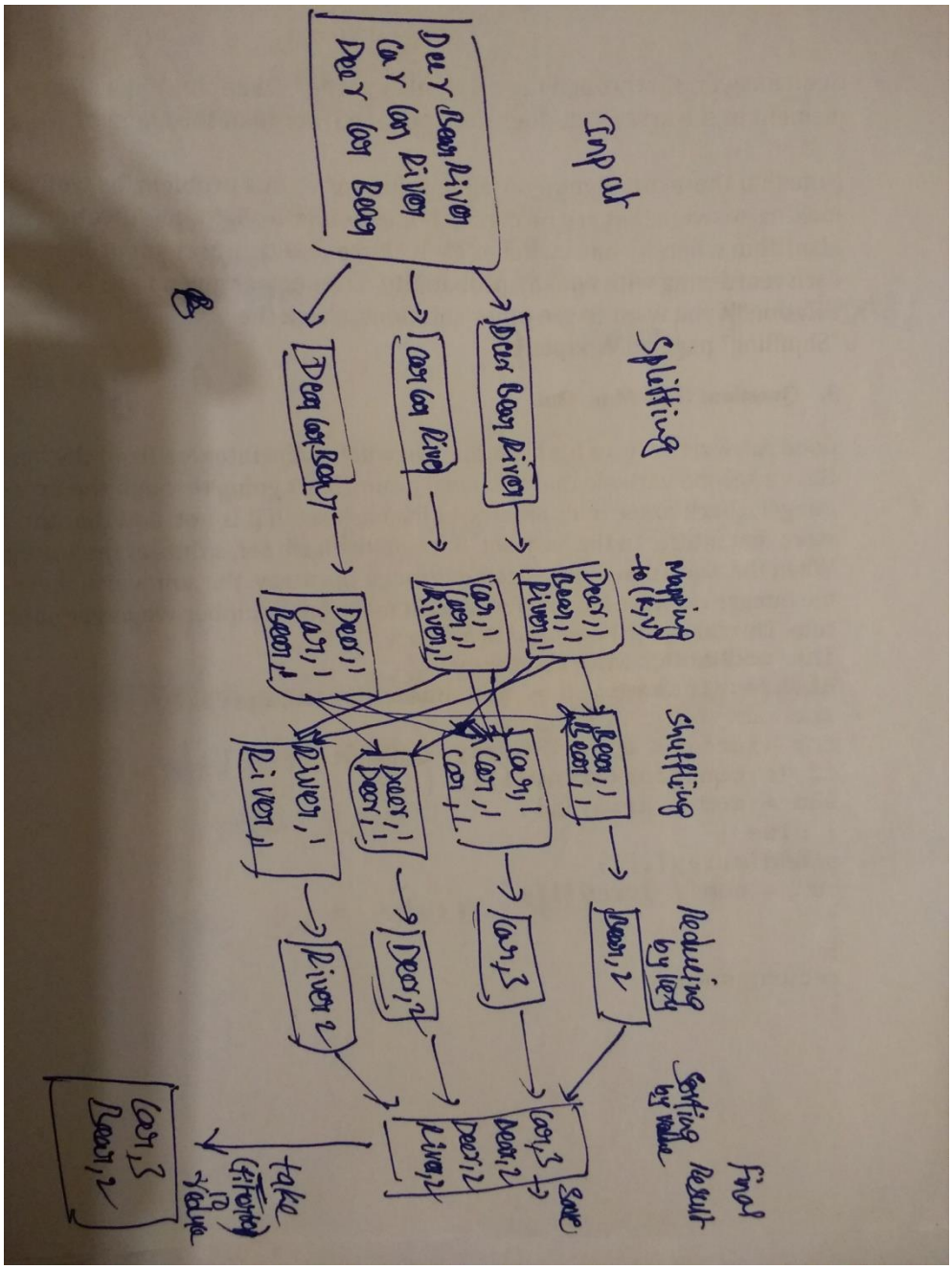


Diagram :


Input
Splitting
mapping to (k,v)
shuffling
Reducing by key
Sorting by value
Final Result
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
River, 1
Deer, 1
Deer, 1
Car, 3
Deer, 2
Bear, 2
River, 1
River, 1
Deer, 2
River, 2
Car, 3
Bear, 2
take (Category ID) value
Car, 3
Bear, 2