

Lab 2 Report


Input: Text Data as Input.

```
Open ▾ [icon]  
Make America great again  
America is great  
We will Rise again  
Great America is number one  
Can America achieve number one?
```

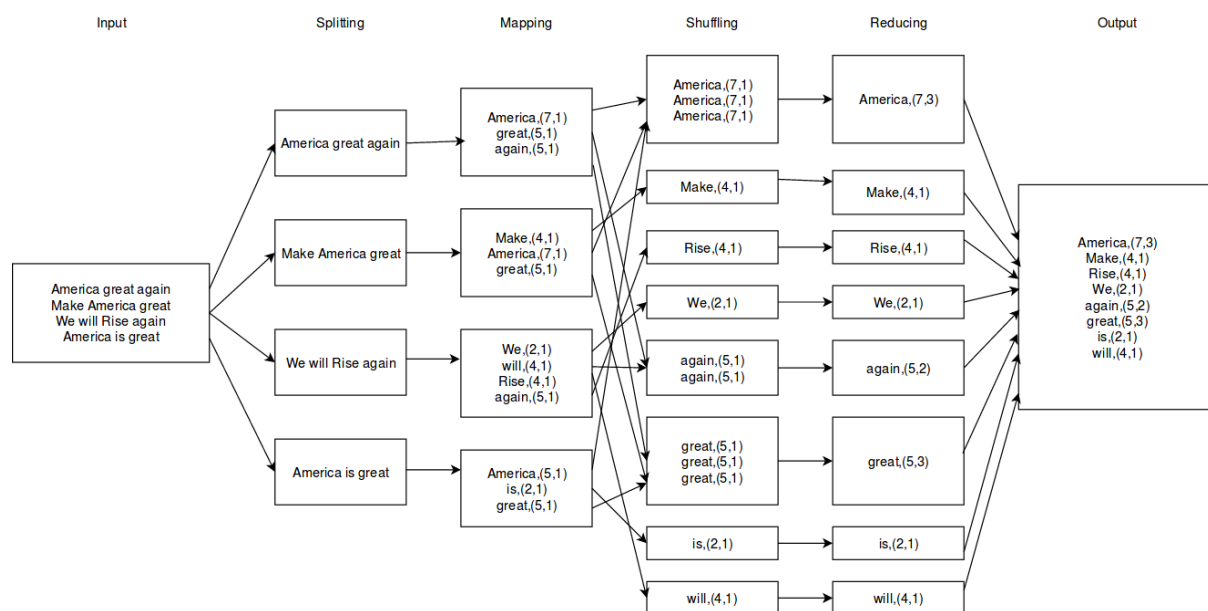
Word Count Output 1:

```
Open ▾ [icon]  
(America,(7,4))  
(Can,(3,1))  
(Great,(5,1))  
(Make,(4,1))  
(Rise,(4,1))  
(We,(2,1))  
(achieve,(7,1))
```

Output 2:

```
Open ▾   
(again,(5,2))  
(great,(5,2))  
(is,(2,2))  
(number,(6,2))  
(one,(3,1))  
(one?,(4,1))  
(will,(4,1))
```

MapReduce Process:



Spark Program Code:

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

object SparkSortWord {

  def main(args: Array[String]) {

    val sparkConf = new
SparkConf().setAppName("SparkSortWord").setMaster("local[*]")

    val sc = new SparkContext(sparkConf)

    val input = sc.textFile("inputtext")

    val wordsort=input.flatMap(line=>{line.split("
")}).map(word=>(word,word.size)).cache()           //Spark
Transformation 1 and 2

    val ignoreDuplicates=wordsort.reduceByKey((a,b)=>a)
//Spark Action 1

    val wordcount=input.flatMap(line=>{line.split("
")}).map(word=>(word,1)).cache()

    val output=wordcount.reduceByKey(_+_ )

    val join=ignoreDuplicates.join(output).sortByKey()
//Spark Transformation 3 and 4

    join.saveAsTextFile("outputtext")
//Spark Action 2

  }

}
```