lab2_2

```
val wordsList = List("cat", "elephant", "rat", "rat", "cat")
val wordsRDD = sc.parallelize(wordsList)

// Print out the type of wordsRDD
println(wordsRDD.getClass)
```

Took: 968 milliseconds, at 2016-9-28 13:31

```
def makePlural(word: String): String = {
//     """Adds an 's' to `word`.
//     Note:
//         This is a simple function that only adds an 's'.  No attempt is made to fol
//         pluralization rules.
//     Args:
//         word (str): A string.
//     Returns:
//         str: A string with 's' added to it.
//     """

    return word+"s"
}

println(makePlural("cat"))
```

Took: 980 milliseconds, at 2016-9-28 13:31

```
val pluralRDD = wordsRDD.map(s=>makePlural(s))
pluralRDD.collect().foreach(println)
```

Took: 1 second 351 milliseconds, at 2016-9-28 13:31

```
val pluralLambdaRDD = wordsRDD.map((s:String)=>s+'s')
pluralLambdaRDD.collect().foreach(println)
```

Took: 1 second 223 milliseconds, at 2016-9-28 13:31

```
val pluralLengths = pluralRDD.map((s:String)=> (s,s.length)).collect()
pluralLengths.foreach(println)
```

Took: 1 second 208 milliseconds, at 2016-9-28 13:31

```
val wordPairs = wordsRDD.map((s:String)=> (s,1))
wordPairs.collect().foreach(println)
```

Took: 942 milliseconds, at 2016-9-28 13:31

```
// Note that groupByKey requires no parameters
val wordsGrouped = wordPairs.groupByKey()
wordsGrouped.collect().foreach(println)
```

Took: 1 second 228 milliseconds, at 2016-9-28 13:31

```
val wordCountsGrouped = wordsGrouped.map(x => (x._1,(x._2).foldLeft(0)((a,b)=>a+b)))
wordCountsGrouped.collect().foreach(println)
```

Took: 954 milliseconds, at 2016-9-28 13:31

```
val wordCounts = wordPairs.reduceByKey((count1,count2)=>count1+count2)
wordCounts.collect().foreach(println)
```

Took: 1 second 286 milliseconds, at 2016-9-28 13:31

```
val wordCountsCollected = wordsRDD.map((s:String)=> (s,1)).reduceByKey((count1,count2)=
wordCountsCollected.foreach(println)
```

Took: 791 milliseconds, at 2016-9-28 13:31

```
val uniqueWords = wordsRDD.distinct.count
println(uniqueWords)
```

Took: 864 milliseconds, at 2016-9-28 13:31

```
val totalCount = wordCounts.map(tuple=>tuple._2.toInt).reduce((v1,v2)=>v1+v2)
val average = totalCount / uniqueWords.toFloat

println(totalCount)
println(average)
```

Took: 723 milliseconds, at 2016-9-28 13:31

```
// TODO: Replace <FILL IN> with appropriate code
import org.apache.spark.rdd.RDD

def wordCount(wordListRDD: RDD[String]): RDD[(String, Int)] = {
    wordListRDD.map((s:String)=> (s,1)).reduceByKey((count1,count2)=>count1+count2)
}

wordCount(wordsRDD).collect().foreach(println)
```

Took: 990 milliseconds, at 2016-9-28 13:32

```
// Just run this code
import scala.util.matching

def removePunctuation(text: String): String = {
    text.replaceAll("""\p{Punct}|^\s+|\s+$""", "").toLowerCase
}

println(removePunctuation("Hi, you!"))
println(removePunctuation(" No under_score!"))
```

Took: 688 milliseconds, at 2016-9-28 13:33

```
// Just run this code
val fileName="/home/akash/kth/data_intensive/labs/id2221/lab2/data/story/shakespeare.t
val shakespeareRDD = sc.textFile(fileName, 8).map(removePunctuation)
shakespeareRDD.zipWithIndex().take(15).map(x => (x._2 + 1) + ": " + x._1).foreach(prin
```

Took: 1 second 297 milliseconds, at 2016-9-28 14:28

```
//test zip with Index
print(List("spark","flink","storm").zipWithIndex)
shakespeareRDD.take(5).foreach(println)
```

Took: 974 milliseconds, at 2016-9-28 14:6

```
val shakespeareWordsRDD = shakespeareRDD.flatMap(x=>x.split("\\s"))
val shakespeareWordCount = shakespeareWordsRDD.count()

shakespeareWordsRDD.top(5).foreach(println)
println(shakespeareWordCount)
```

Took: 1 second 441 milliseconds, at 2016-9-28 14:35

```
val shakeWordsRDD = shakespeareWordsRDD.filter(word => word != "")
val shakeWordCount = shakeWordsRDD.count()

println(shakeWordCount)
```

Took: 1 second 20 milliseconds, at 2016-9-28 14:35

```
//val top15WordsAndCounts = wordCount(shakeWordsRDD).sortBy(x=>x._2,false).take(15).ma
val top15WordsAndCounts = wordCount(shakeWordsRDD).top(15)(Ordering[Int].on(x=>x._2)).
top15WordsAndCounts.map(x => x._1 + ": " + x._2).foreach(println)
```

Took: 1 second 576 milliseconds, at 2016-9-28 14:56

Took: 747 milliseconds, at 2016-9-28 15:8

Build: | **buildTime**-*Sun Sep 25 12:54:11 UTC 2016* | **formattedShaVersion**-*0.7.0-SNAPSHOT-f6cd60a95cfc19cbae80285f187da9baec04e436* | **sbtVersion**-*0.13.8* | **scalaVersion**-*2.10.6* | **sparkNotebookVersion**-*0.7.0-SNAPSHOT* | **hadoopVersion**-*2.2.0* | **jets3tVersion**-*0.7.1* | **jlineDef**-*(org.scala-lang,2.10.6)* | **sparkVersion**-*2.0.0* | **withHive**-*false* |.