Week 2: Introduction to Spark and Resilient Distributed Datasets

PairRDDs

- RDDs that consist of key-value pairs
- note that "keys" are usually not unique (in contrast to python dictionaries)
- PairRDDs provide additional functionality, most notably:
  - sortBy()
  - sortByKey()
  - reduceByKey()

PairRDDs: sortBy()

- like the Python sorted() using a key= function
- example:

```
word_counts_sorted = word_count3.sortBy(lambda x: x[1],
      ascending = False)
```

PairRDDs: sortByKey()

- sorts key-value pairs by keys *alphabetically*
- example:

```
word_counts_sorted = word_counts.sortByKey()
```

PairRDDs: reduceByKey()

- takes a function that operates on the values of two elements with the same key and returns a new RDD
- example:

```
sumRDD = rdd.reduceByKey(lambda accumulator, value:
accumulator + value)
```

Word count example

```
input_file = sc.textFile("data/totc.txt")

word_count1 = input_file.flatMap(lambda line: line.split())
word_count2 = word_count1.map(lambda word: (word, 1))
word_count3 = word_count2.reduceByKey(lambda a, b: a + b)

word_counts_sorted = word_count3.sortBy(lambda x: x[1], ascending = False)

word_counts_sorted.take(10)
```