5c_sortbykey

February 21, 2022

0.0.1 sortByKey

Signature: sortByKey(ascending=True, numPartitions=None, keyfunc=<function RDD. at 0x7f0b28a192f0>)

Docstring: Sorts this RDD, which is assumed to consist of (key, value) pairs.

[ ]:
```python
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf

# Initialize spark
conf = SparkConf().setAppName("sortByKey")
sc = SparkContext(conf=conf)
```

[ ]:
```python
tmp = [('a', 1), ('b', 2), ('1', 3), ('d', 4), ('2', 5)]
sc.parallelize(tmp).sortByKey().collect() # combine parallelization and transformation
```

[ ]:
```python
# sortByKey - Ascending and Descending order with 2 partitions
tmpRDD = sc.parallelize(tmp)
tmpRDD.sortByKey(True, 2).collect() # Ascending Order
```

[ ]:
```python
tmpRDD.sortByKey(False, 2).collect() # Descending Order
```

0.0.2 Another example...

[ ]:
```python
tmp2 = [('Mary', 1), ('had', 2), ('a', 3), ('Little', 4), ('lamb', 5)]
tmp2.extend([('whose', 6), ('fleece', 7), ('was', 8), ('white', 9)])

tmp2RDD = sc.parallelize(tmp2)

# Convert every key to lower case and then sort by key
# The conversion is only used for sorting, original data will not be changed
tmp2RDD.sortByKey(True, 3, keyfunc=lambda k: k.lower()).collect()
```