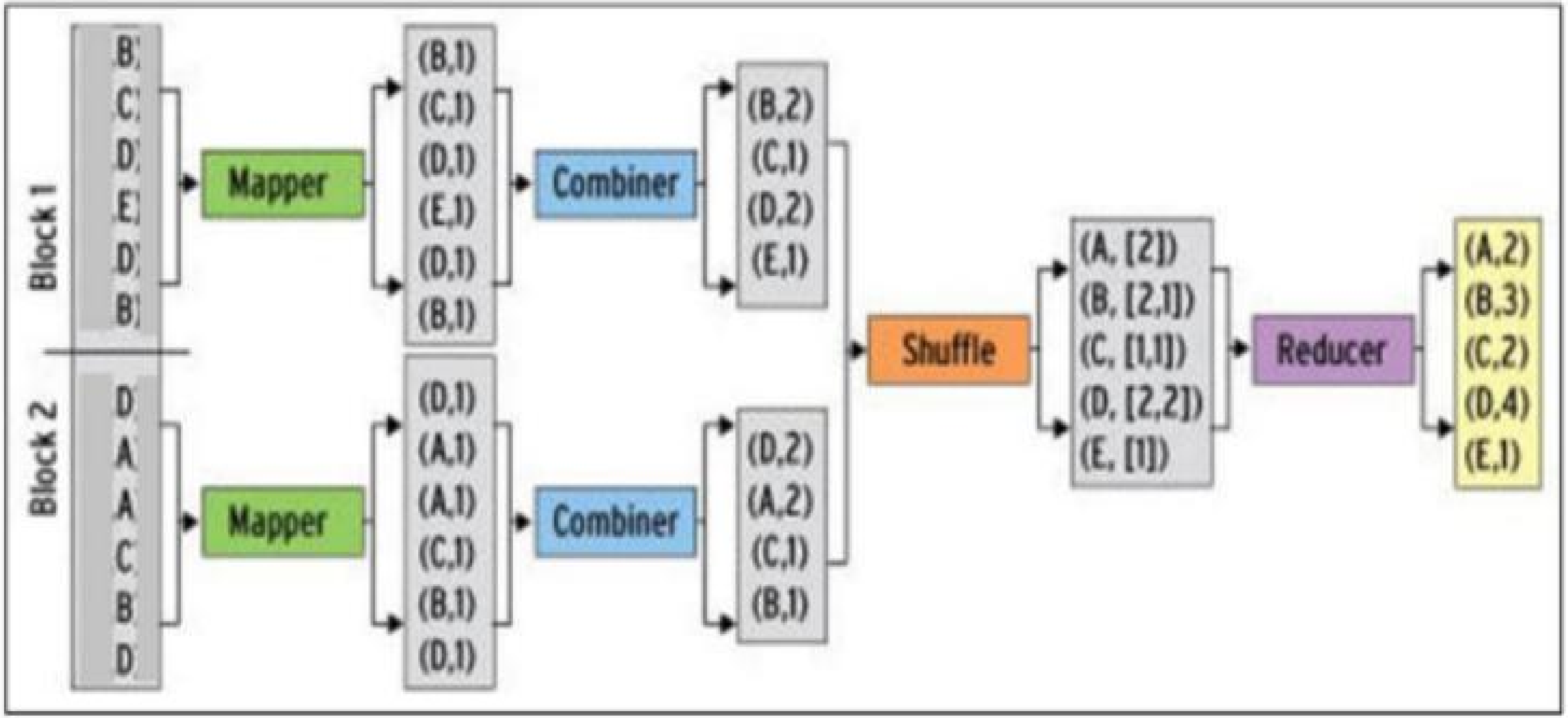
MapReduce

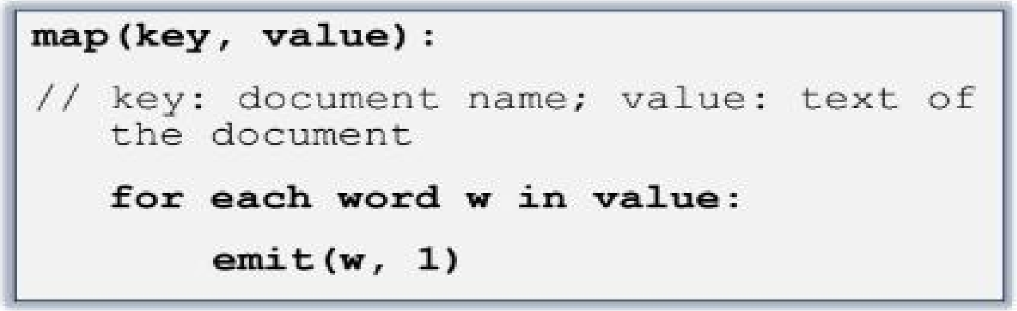
Lec 04

MapReduce

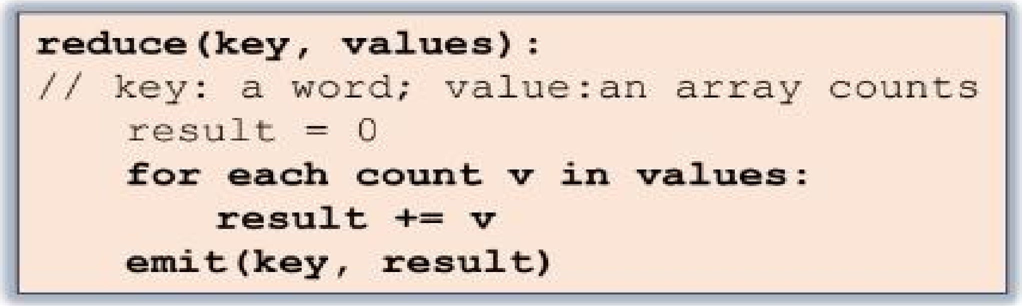

Block 1
B
C
D
E
D
B
Block 2
D
A
A
C
B
D
Mapper
Mapper
(B,1)
(C,1)
(D,1)
(E,1)
(D,1)
(B,1)
(D,1)
(A,1)
(A,1)
(C,1)
(B,1)
(D,1)
Combiner
Combiner
(B,2)
(C,1)
(D,2)
(E,1)
(D,2)
(A,2)
(C,1)
(B,1)
Shuffle
(A, [2])
(B, [2,1])
(C, [1,1])
(D, [2,2])
(E, [1])
Reducer
(A,2)
(B,3)
(C,2)
(D,4)
(E,1)

Word Count using MapReduce

```
from mrjob.job import MRJob

class WordCount(MRJob):

  def mapper(self, _, line):
      for word in line.split():
         yield(word, 1)

  def combiner(self, word, counts):
      yield(word, sum(counts))

   def reducer(self, word, counts):
     yield(word, sum(counts))

if __name__ == '__main__':
   WordCount.run()
```

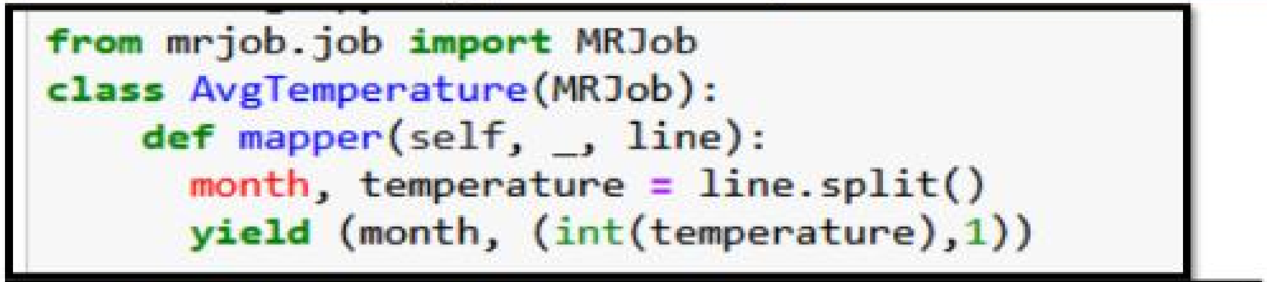


```
map(key, value):
// key: document name; value: text of
   the document
   for each word w in value:
       emit(w, 1)
```

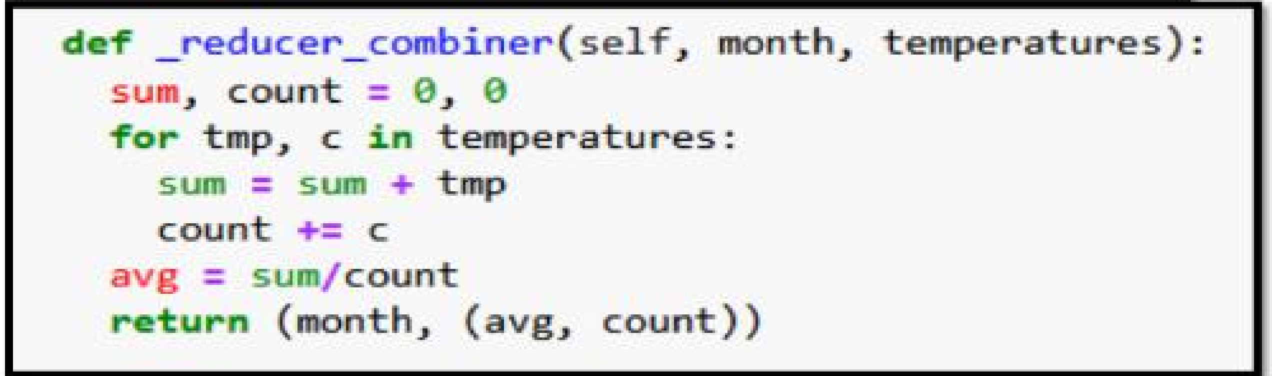


```
reduce(key, values):
// key: a word; value:an array counts
    result = 0
    for each count v in values:
        result += v
    emit(key, result)
```

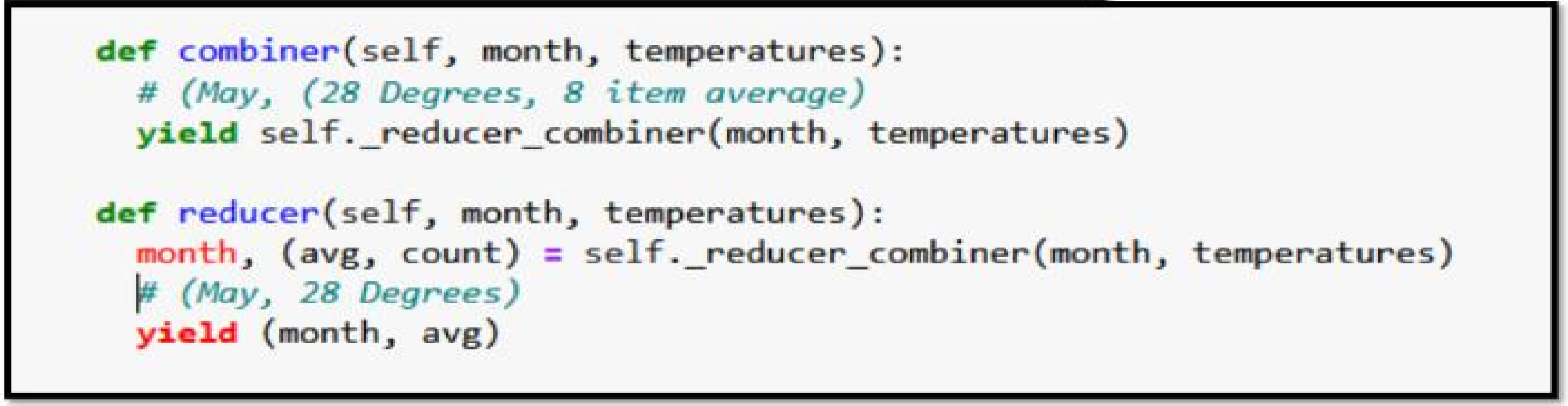
Average Temperatures



```
from mrjob.job import MRJob
class AvgTemperature(MRJob):
    def mapper(self, _, line):
      month, temperature = line.split()
      yield (month, (int(temperature),1))
```

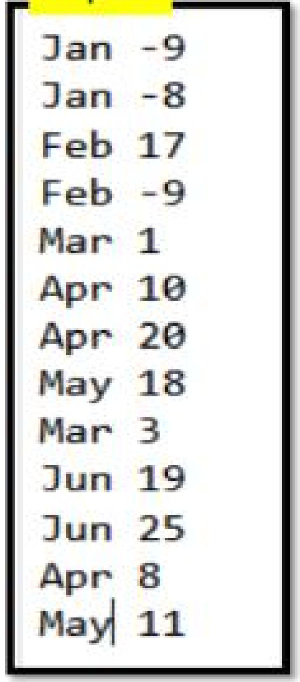


```
def _reducer_combiner(self, month, temperatures):
  sum, count = 0, 0
  for tmp, c in temperatures:
    sum = sum + tmp
    count += c
  avg = sum/count
  return (month, (avg, count))
```



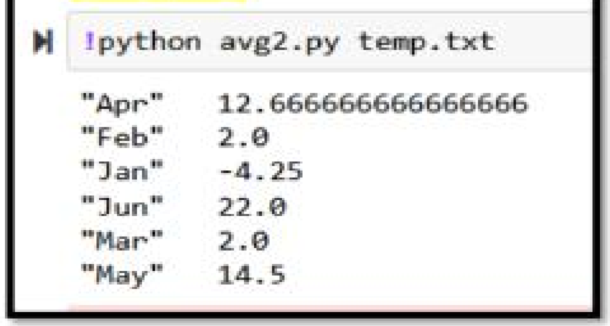
```
  def combiner(self, month, temperatures):
    # (May, (28 Degrees, 8 item average)
    yield self._reducer_combiner(month, temperatures)

  def reducer(self, month, temperatures):
    month, (avg, count) = self._reducer_combiner(month, temperatures)
    # (May, 28 Degrees)
    yield (month, avg)
```

Input



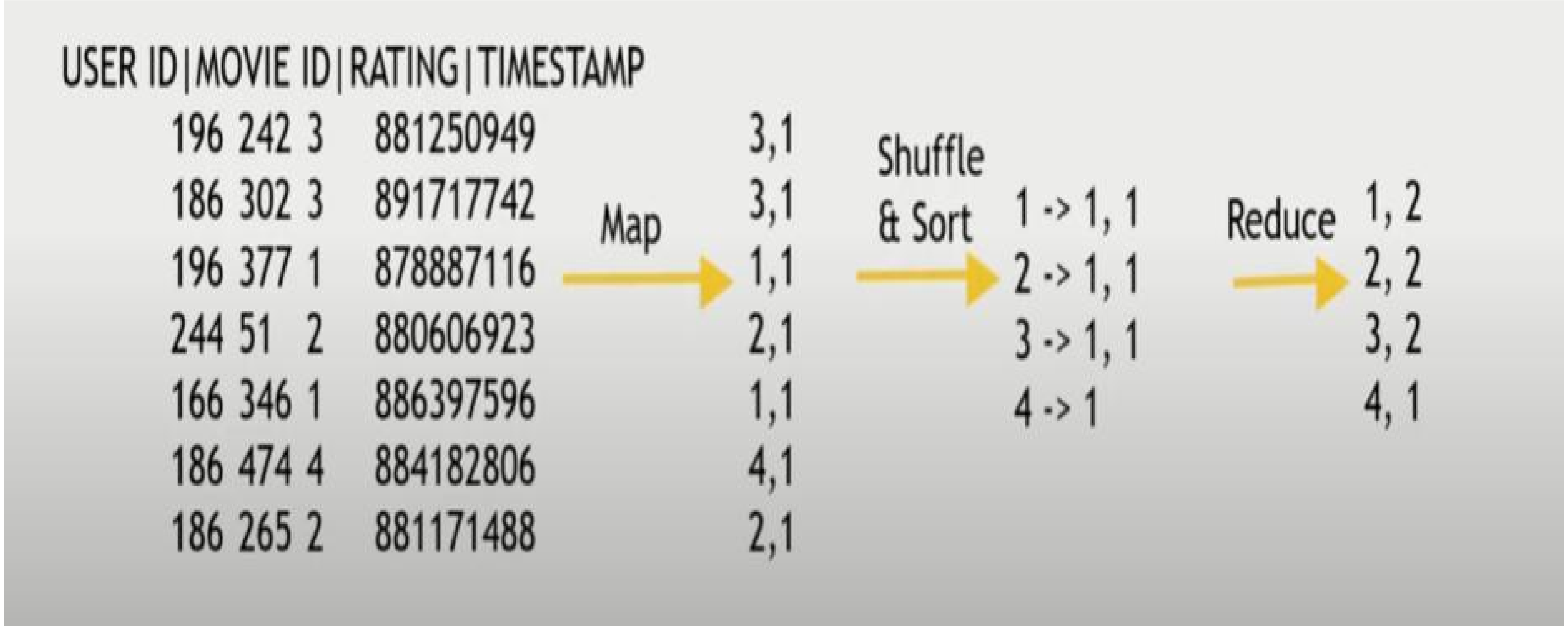
```
Jan -9
Jan -8
Feb 17
Feb -9
Mar 1
Apr 10
Apr 20
May 18
Mar 3
Jun 19
Jun 25
Apr 8
May 11
```

Output



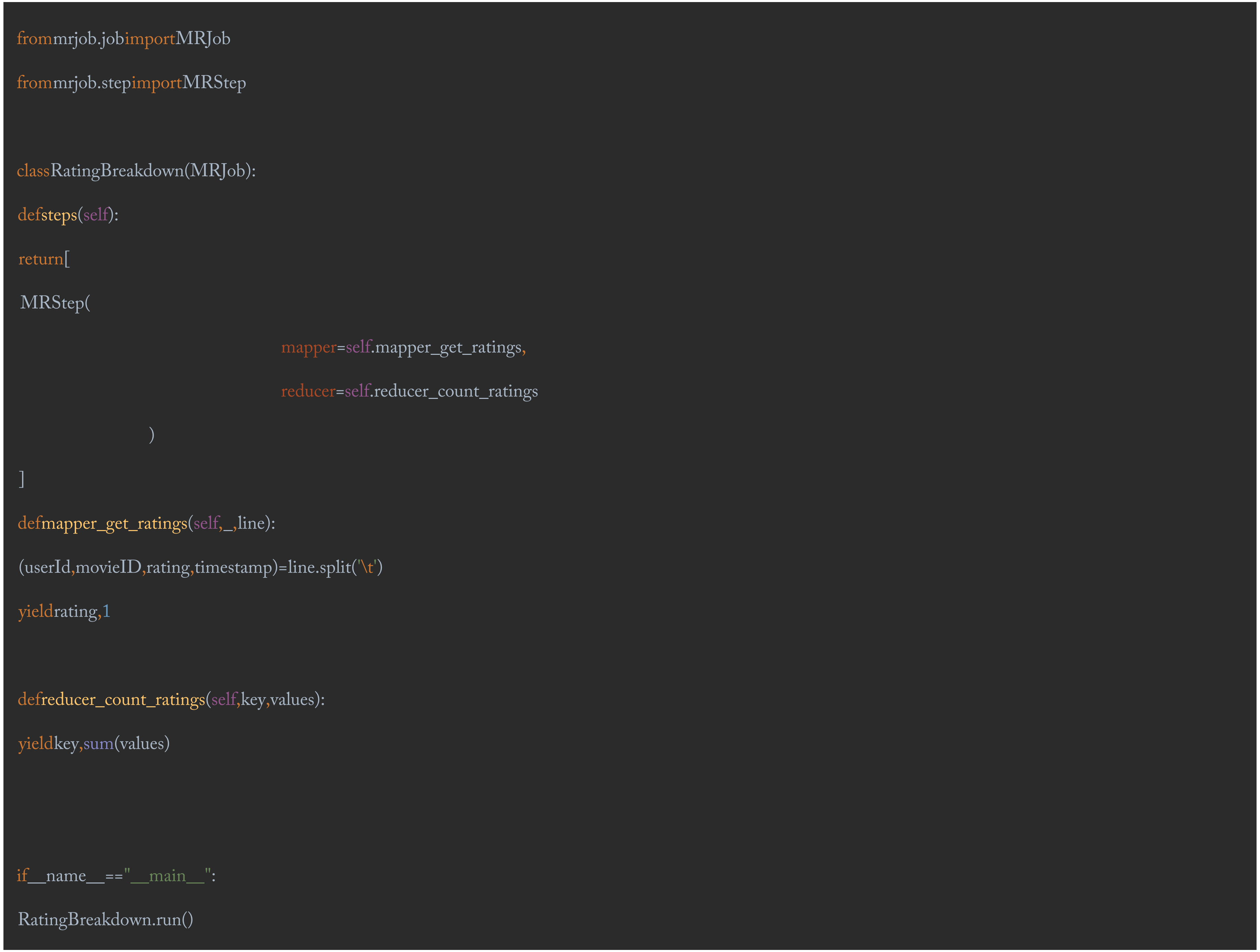
```
!python avg2.py temp.txt

"Apr"   12.666666666666666
"Feb"   2.0
"Jan"   -4.25
"Jun"   22.0
"Mar"   2.0
"May"   14.5
```

Movie Rating

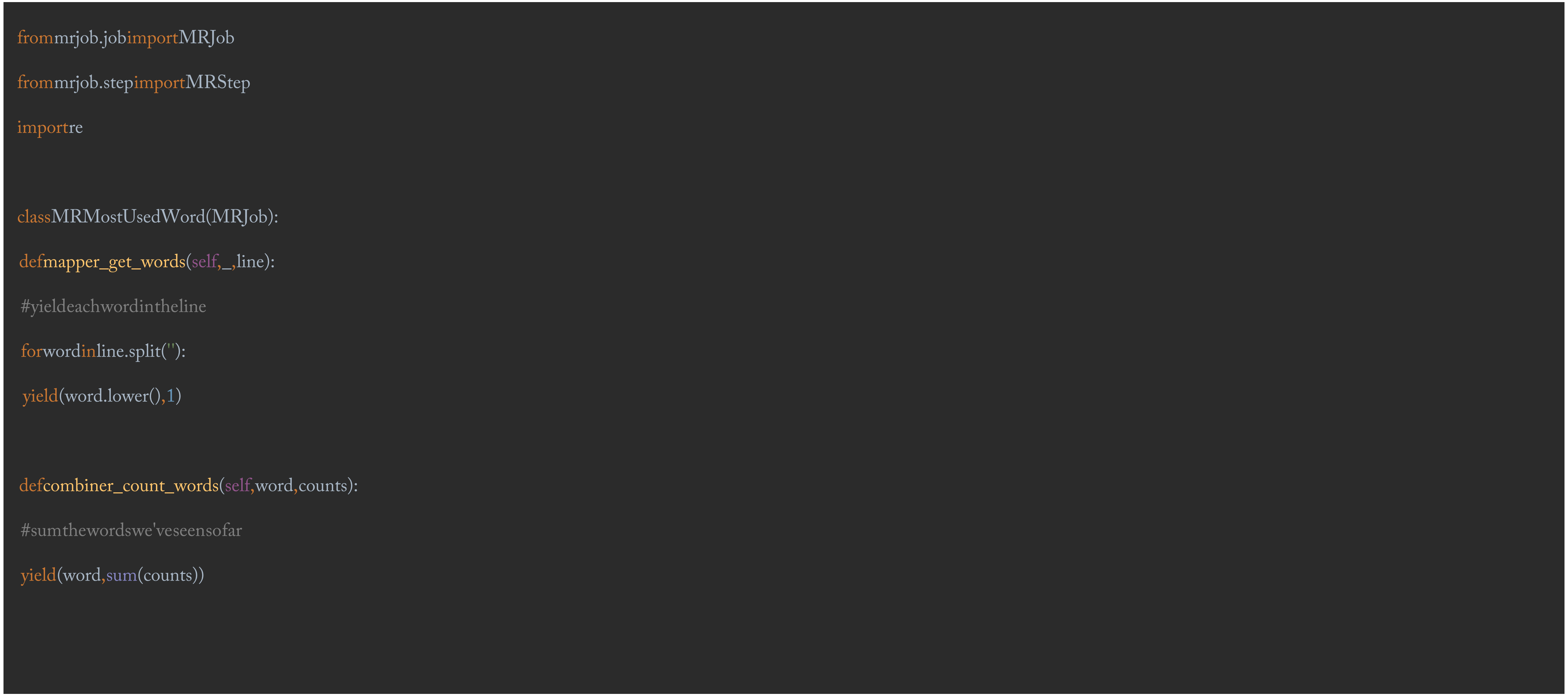


USER ID|MOVIE ID|RATING|TIMESTAMP

```
196 242 3    881250949
186 302 3    891717742
196 377 1    878887116
244 51  2    880606923
166 346 1    886397596
186 474 4    884182806
186 265 2    881171488
```



```
frommrjob.jobimportMRJob

frommrjob.stepimportMRStep


classRatingBreakdown(MRJob):

defsteps(self):

return[

 MRStep(

                                        mapper=self.mapper_get_ratings,

                                        reducer=self.reducer_count_ratings

                    )

]

defmapper_get_ratings(self,_,line):

(userId,movieID,rating,timestamp)=line.split('\t')

yieldrating,1


defreducer_count_ratings(self,key,values):

yieldkey,sum(values)



if__name__=="__main__":

RatingBreakdown.run()
```

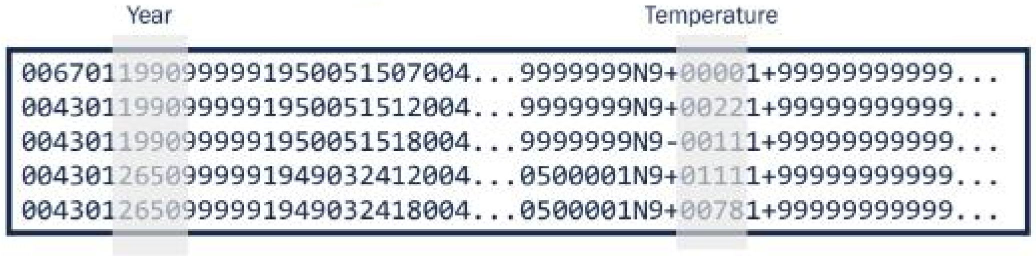


```
frommrjob.jobimportMRJob

frommrjob.stepimportMRStep

importre


classMRMostUsedWord(MRJob):

defmapper_get_words(self,_,line):

#yieldeachwordintheline

forwordinline.split(''):

yield(word.lower(),1)


defcombiner_count_words(self,word,counts):

#sumthewordswe'veseensofar

yield(word,sum(counts))
```

```
defreducer_count_words(self,word,counts):
 #sendall(num_occurrences,word)pairstothesamereducer.
 #num_occurrencesissowecaneasilyusePython'smax()function.
 yieldNone,(sum(counts),word)
 #discardthekey;itisjustNone

defreducer_find_max_word(self,_,word_count_pairs):
 #eachitemofword_count_pairsis(count,word),
 #soyieldingoneresultsinkey=counts,value=word
 yieldmax(word_count_pairs)

defsteps(self):
 return[
 MRStep(
  mapper=self.mapper_get_words,
  combiner=self.combiner_count_words,
  reducer=self.reducer_count_words),
 MRStep(
  reducer=self.reducer_find_max_word)
 ]
if__name__=='__main__':
MRMostUsedWord.run()
```

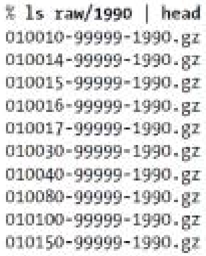
Analysis of Weather Dataset

- **Data from NCDC(National Climatic Data Center):** A large volume of log data collected by weather sensors: e.g. temperature
- **Data format**
  - *Line-oriented ASCII format with many elements*
  - *We focus on the temperature element*
  - *Data files are organized by date and weather station*


Year
Temperature

```
0067011990999991950051507004...9999999N9+00001+99999999999...
0043011990999991950051512004...9999999N9+00221+99999999999...
0043011990999991950051518004...9999999N9-00111+99999999999...
0043012650999991949032412004...0500001N9+01111+99999999999...
0043012650999991949032418004...0500001N9+00781+99999999999...
```

Contents of data files



```
% ls raw/1990 | head
010010-99999-1990.gz
010014-99999-1990.gz
010015-99999-1990.gz
010016-99999-1990.gz
010017-99999-1990.gz
010030-99999-1990.gz
010040-99999-1990.gz
010080-99999-1990.gz
010100-99999-1990.gz
010150-99999-1990.gz
```

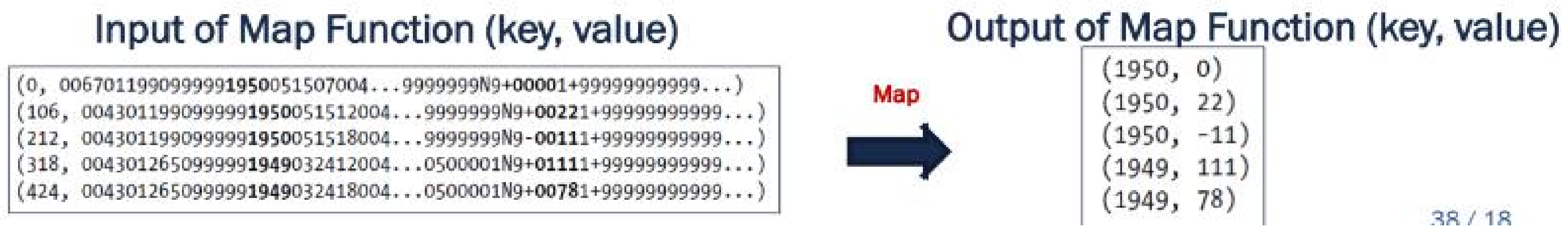
List of data files

MapReduce Design of NCDC Example

- Map phase
  - *Text input format of the dataset files*
    - Key: offset of the line (unnecessary)
    - Value: each line of the files
  - *Pull out the year and the temperature*
    - The map phase is simply data preparation phase
    - Drop bad records(filtering)

```
0067011990999991950051507004...9999999N9+00001+99999999999...
0043011990999991950051512004...9999999N9+00221+99999999999...
0043011990999991950051518004...9999999N9-00111+99999999999...
0043012650999991949032412004...0500001N9+01111+99999999999...
0043012650999991949032418004...0500001N9+00781+99999999999...
```

Input File



Input of Map Function (key, value)

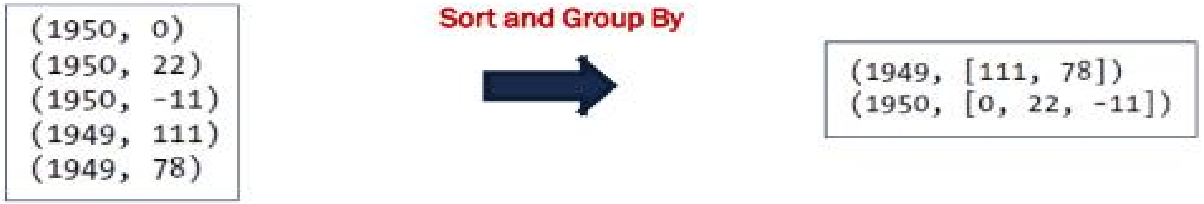
```
(0, 0067011990999991950051507004...9999999N9+00001+99999999999...)
(106, 0043011990999991950051512004...9999999N9+00221+99999999999...)
(212, 0043011990999991950051518004...9999999N9-00111+99999999999...)
(318, 0043012650999991949032412004...0500001N9+01111+99999999999...)
(424, 0043012650999991949032418004...0500001N9+00781+99999999999...)
```

Output of Map Function (key, value)

```
(1950, 0)
(1950, 22)
(1950, -11)
(1949, 111)
(1949, 78)
```

MapReduce Design of NCDC Example

The output from the map function is processed by MapReduce framework

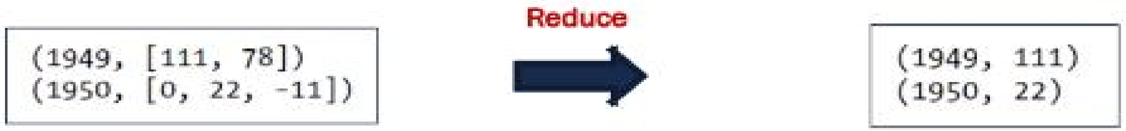


```
(1950, 0)
(1950, 22)
(1950, -11)
(1949, 111)
(1949, 78)
```

Sort and Group By

```
(1949, [111, 78])
(1950, [0, 22, -11])
```

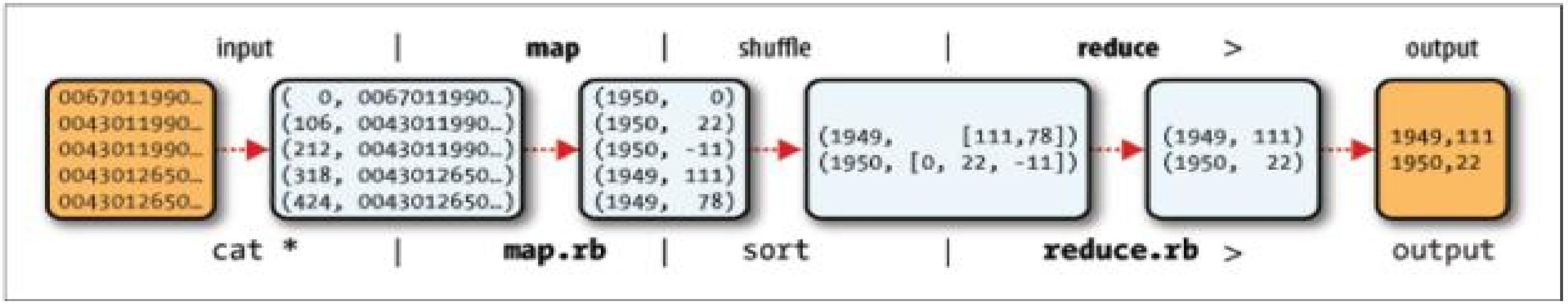
- Reduce function iterates through the list and pick up the maximum value



```
(1949, [111, 78])
(1950, [0, 22, -11])
```

Reduce

```
(1949, 111)
(1950, 22)
```



```
input | map | shuffle | reduce > output
```

```
0067011990...
0043011990...
0043011990...
0043012650...
0043012650...
```

```
(  0, 0067011990...)
(106, 0043011990...)
(212, 0043011990...)
(318, 0043012650...)
(424, 0043012650...)
```

```
(1950,   0)
(1950,  22)
(1950, -11)
(1949, 111)
(1949,  78)
```

```
(1949,      [111,78])
(1950, [0, 22, -11])
```

```
(1949, 111)
(1950,  22)
```

```
1949,111
1950,22
```

```
cat * | map.rb | sort | reduce.rb > output
```

Reference

- https://buildmedia.readthedocs.org/media/pdf/mrjob/latest/mrjob.pdf