mapper.py

#!/usr/bin/env python

"""mapper.py"""

import sys

# input comes from STDIN (standard input)

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# split the line into words

words = line.split()

# increase counters

for word in words:

# write the results to STDOUT (standard output);

# what we output here will be the input for the

# Reduce step, i.e. the input for reducer.py

#

# tab-delimited; the trivial word count is 1

print ('%s\t%s' %(word, 1))

#!/usr/bin/env python

"""reducer.py"""

from operator import itemgetter

import sys

current\_word = None

current\_count = 0

word = None

# input comes from STDIN

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# parse the input we got from mapper.py

word, count = line.split('\t', 1)

# convert count (currently a string) to int

try:

count = int(count)

except ValueError:

# count was not a number, so silently

# ignore/discard this line

continue

# this IF-switch only works because Hadoop sorts map output

# by key (here: word) before it is passed to the reducer

if current\_word == word:

current\_count += count

else:

if current\_word:

# write result to STDOUT

print ('%s\t%s' %(current\_word, current\_count))

current\_count = count

current\_word = word

# do not forget to output the last word if needed!

if current\_word == word:

print ('%s\t%s' %(current\_word, current\_count))

cat index.txt |python3 mapper.py|sort|python3 reducer.py