Natural Language Processing Lab (2003-)

Week 5 MapReduce

Jason S. Chang 張俊盛 jason@nlplab.cc

TA：Joanna Wu 吳婍慈 joannawu@nlplab.cc

LK Huang 黃麟凱 hlk@nlplab.cc

Course Website：https://eeclass.nthu.edu.tw/course/info/4137

2021 1014 Thur 15:30 Online

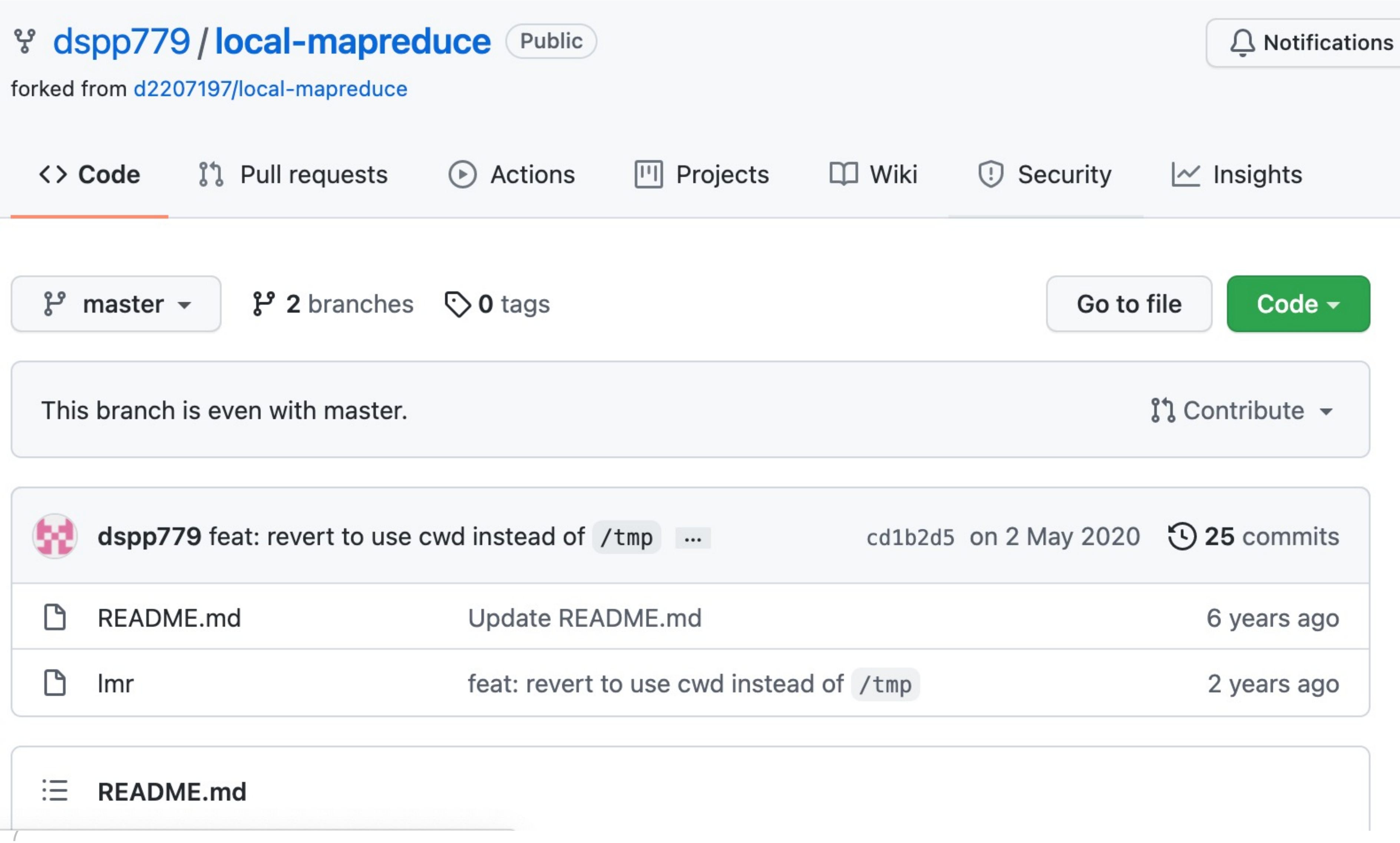
MapReduce according to Wikipedia

- MapReduce is a parallel, distributed model implemented on a computer cluster for processing and generating big data sets
  - □ Control the distributed servers in a cluster
  - □ Run many parallel tasks (mapper and reducer)
  - □ Communicate and transfer big data between tasks
  - □ Maintain redundancy and fault tolerance
- Mappers filter and sort data (e.g., key in alphabetic order)
- MR system sorts (harsh) and distributes data
- Reducers performs a summary operation (e.g., word count).
- Key contributions
  - □ Scalability and fault-tolerance

More about MapReduce

- Open source tools
  - Hadoop (flat text files)
  - Pig (SQL files and operations)
  - Apache Hive
  - Local MapReduce (invented here for this course)
- Use cases
  - word count
  - sorting
  - constructing inverted file for Web search engine
  - document clustering
  - machine learning

Local MapReduce



dspp779 / local-mapreduce Public

forked from d2207197/local-mapreduce

This branch is even with master.

dspp779 feat: revert to use cwd instead of `/tmp` … cd1b2d5 on 2 May 2020 25 commits

README.md	Update README.md	6 years ago
lmr	feat: revert to use cwd instead of `/tmp`	2 years ago

README.md

Local MapReduce and Examples

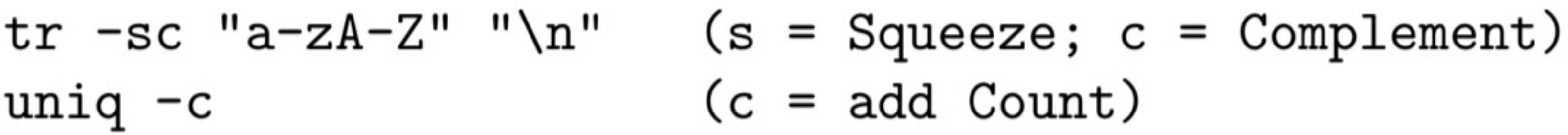
- See https://github.com/dspp779/local-mapreduce
- Usage

  ./lmr <chunk size> <#reducer> <mapper> <reducer> <directory>

  - <chunk size>: Split data into chunks with <chunk size>
  - <#reducer>: Each output line from mappers would then be hashed into <#reducer> different reducer
  - <mapper>, <reducer>: Shell command/Python program
  - <directory>: The output directory

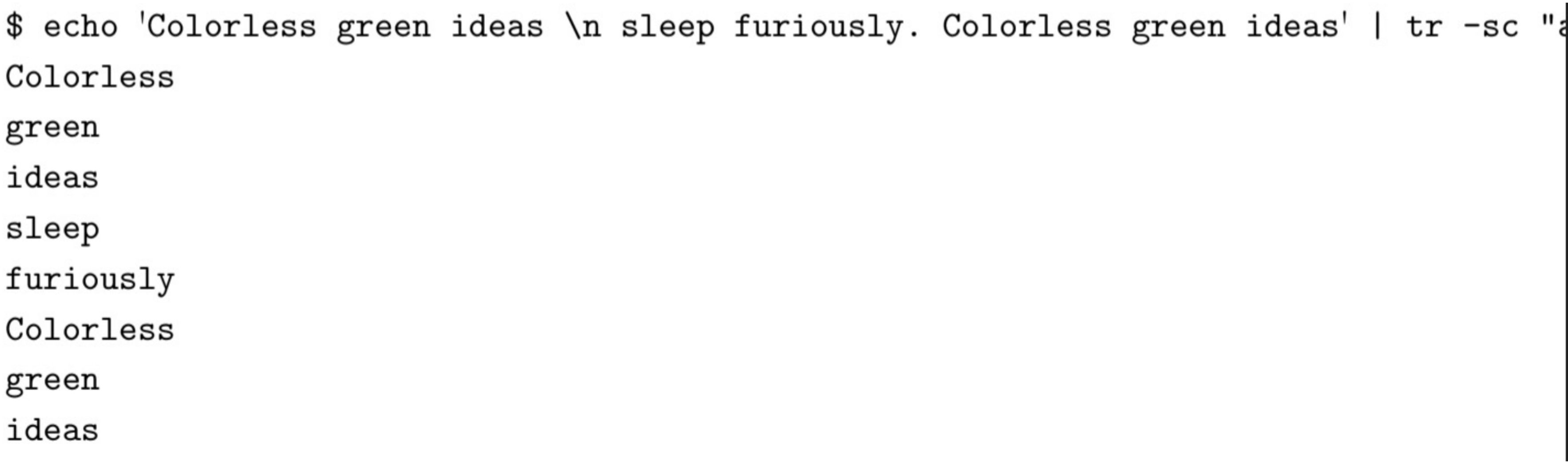
Local MapReduce–Word Count

- Mapper and Reducer



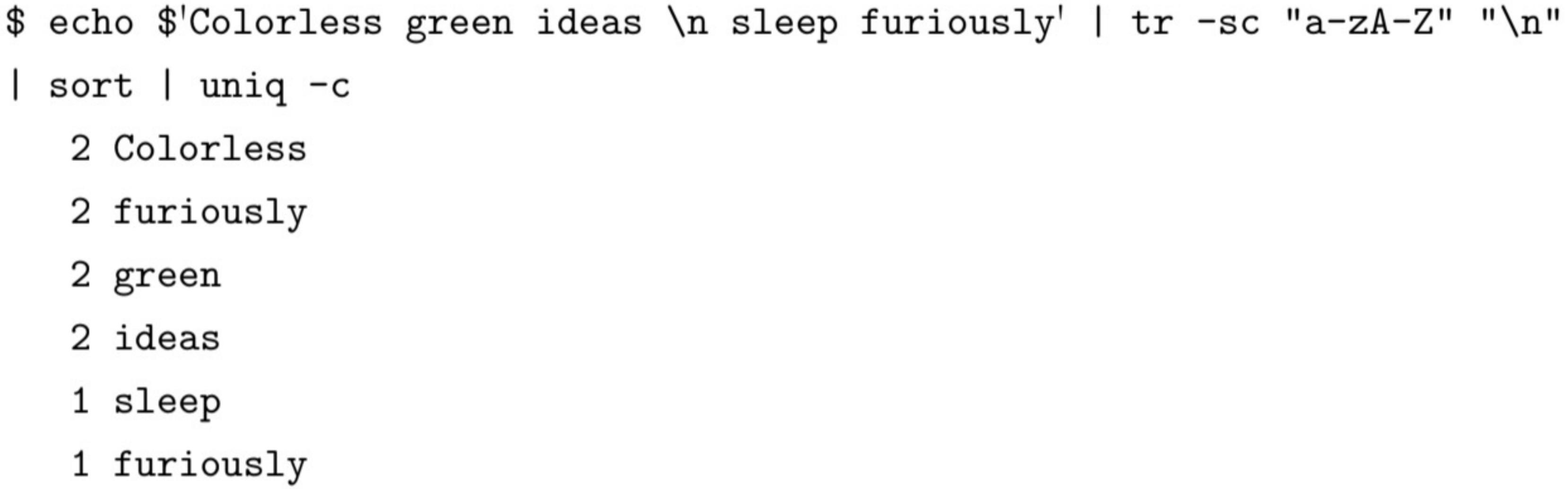
```
tr -sc "a-zA-Z" "\n"    (s = Squeeze; c = Complement)
uniq -c                 (c = add Count)
```

- Testing mapper



```
$ echo 'Colorless green ideas \n sleep furiously. Colorless green ideas' | tr -sc "a
Colorless
green
ideas
sleep
furiously
Colorless
green
ideas
```

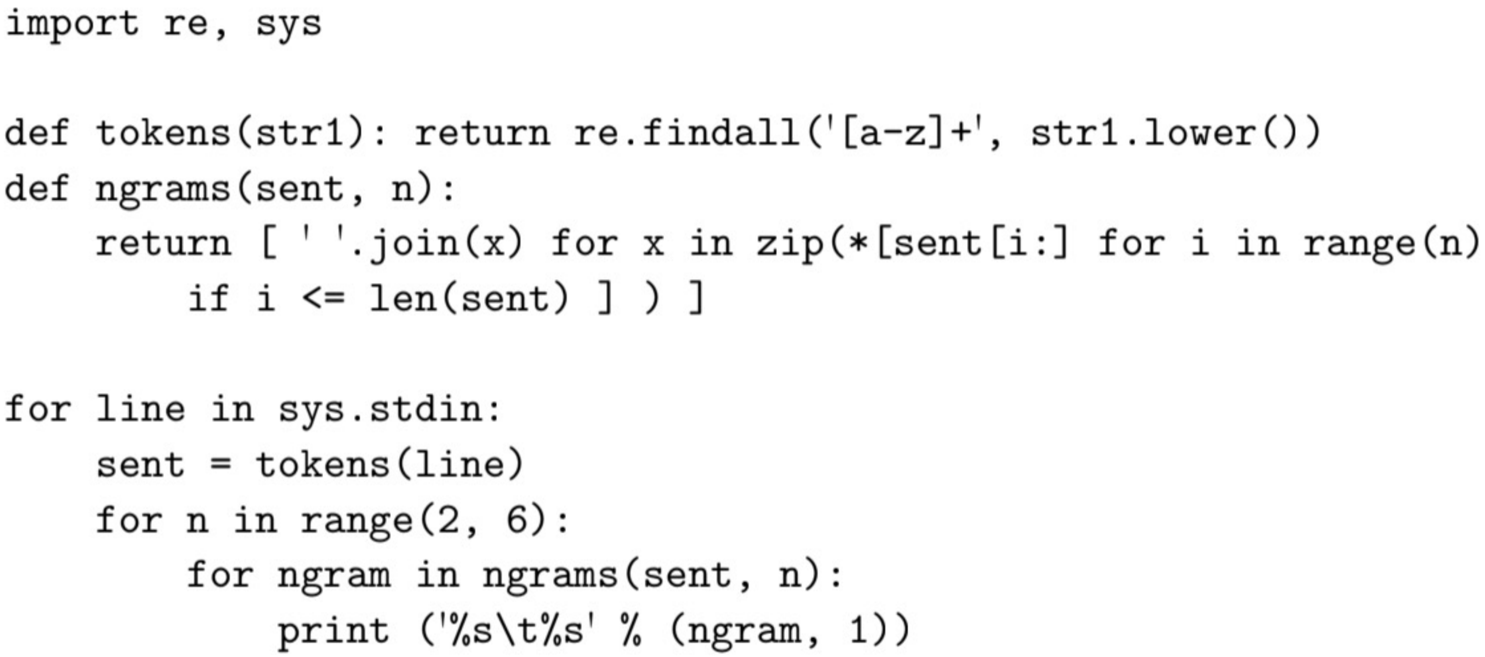
- Testing reducer



```
$ echo $'Colorless green ideas \n sleep furiously' | tr -sc "a-zA-Z" "\n"
| sort | uniq -c
   2 Colorless
   2 furiously
   2 green
   2 ideas
   1 sleep
   1 furiously
```

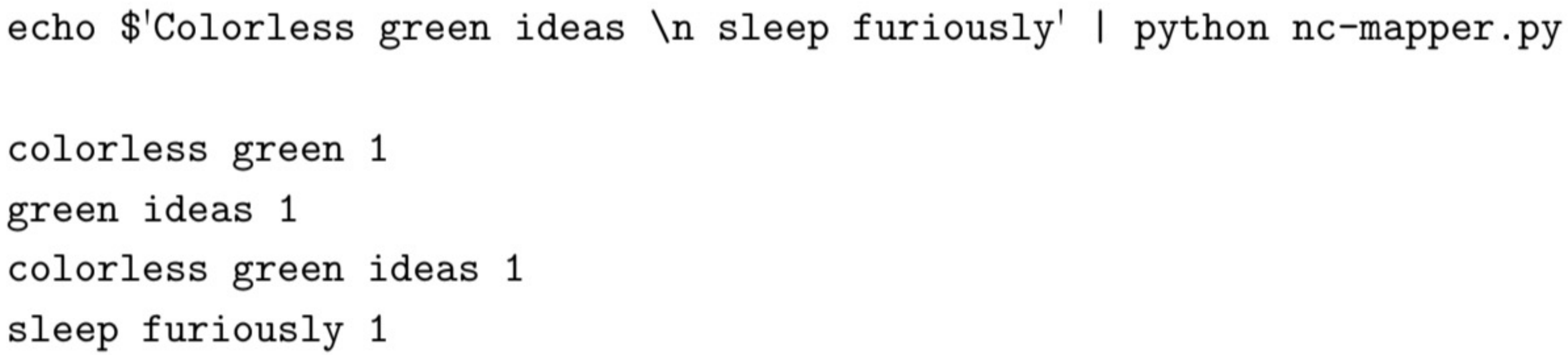
Ngram Count

- Mapper



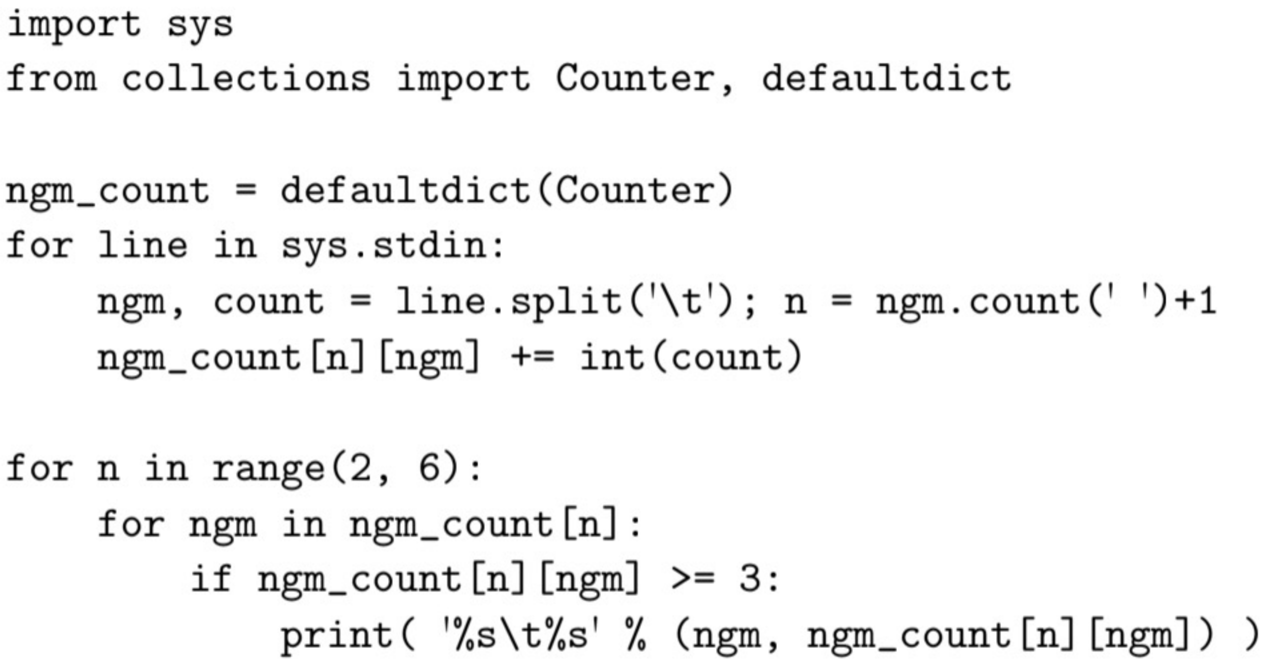
```
import re, sys

def tokens(str1): return re.findall('[a-z]+', str1.lower())
def ngrams(sent, n):
    return [ ' '.join(x) for x in zip(*[sent[i:] for i in range(n)
        if i <= len(sent) ] ) ]

for line in sys.stdin:
    sent = tokens(line)
    for n in range(2, 6):
        for ngram in ngrams(sent, n):
            print ('%s\t%s' % (ngram, 1))
```

- Testing mapper



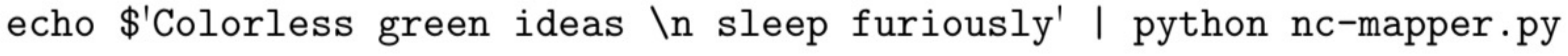
```
echo $'Colorless green ideas \n sleep furiously' | python nc-mapper.py

colorless green 1
green ideas 1
colorless green ideas 1
sleep furiously 1
```

- Reducer

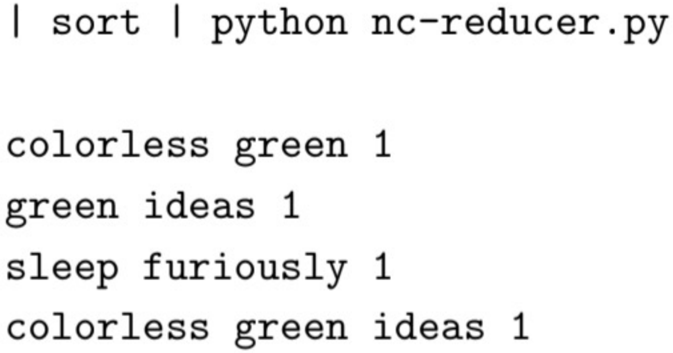


```
import sys
from collections import Counter, defaultdict

ngm_count = defaultdict(Counter)
for line in sys.stdin:
    ngm, count = line.split('\t'); n = ngm.count(' ')+1
    ngm_count[n][ngm] += int(count)

for n in range(2, 6):
    for ngm in ngm_count[n]:
        if ngm_count[n][ngm] >= 3:
            print( '%s\t%s' % (ngm, ngm_count[n][ngm]) )
```

- Testing Reducer

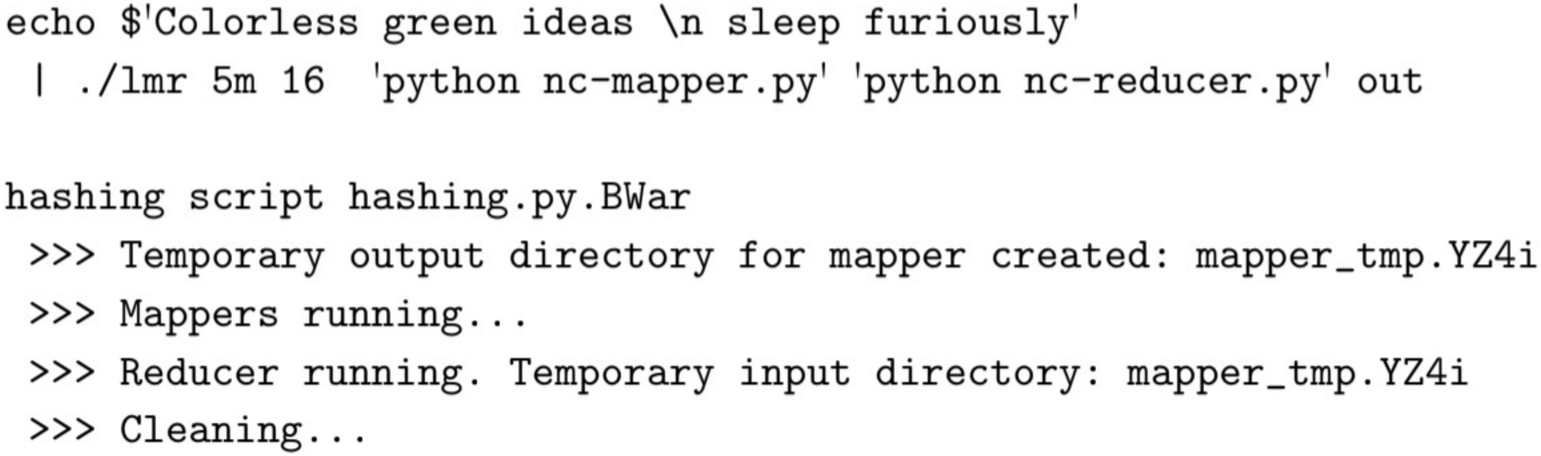


```
echo $'Colorless green ideas \n sleep furiously' | python nc-mapper.py
```



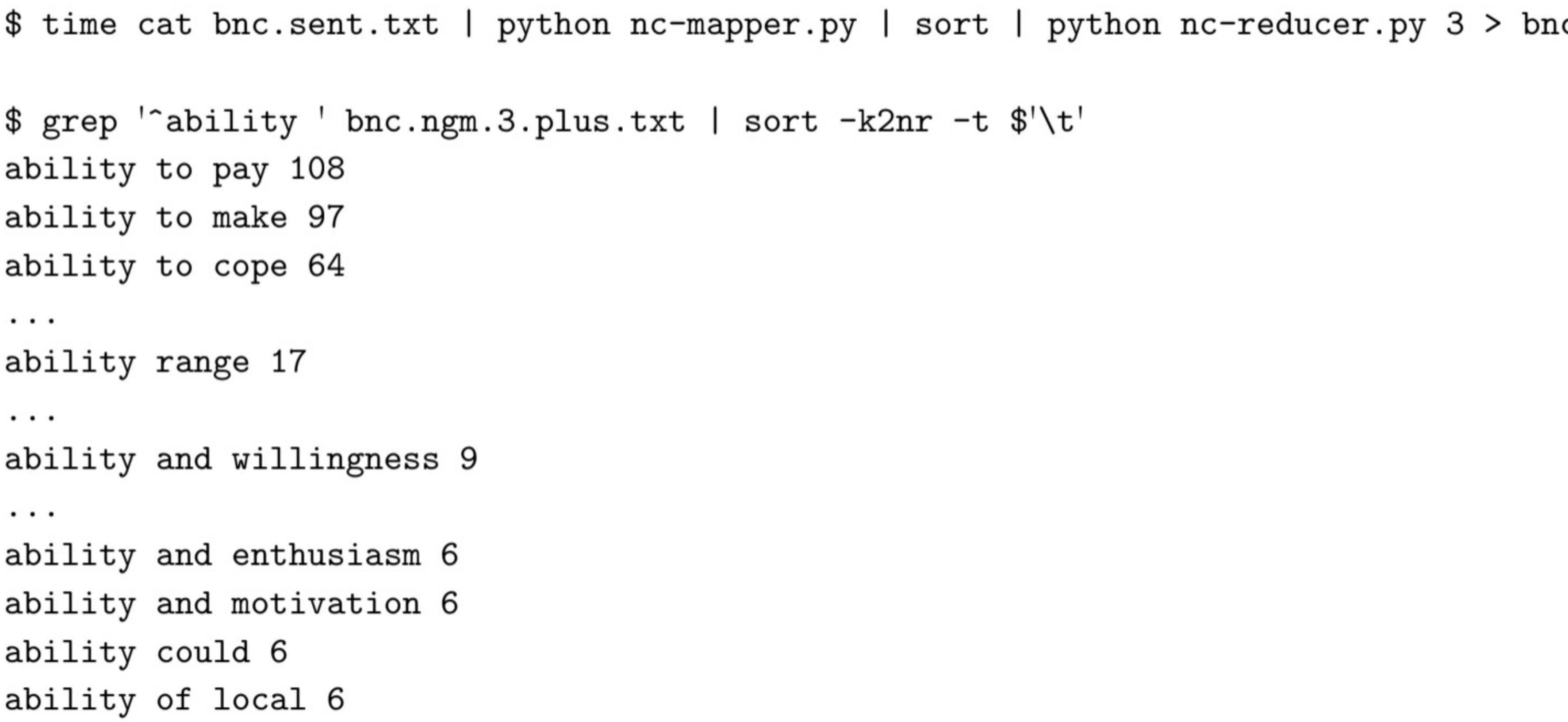
```
| sort | python nc-reducer.py

colorless green 1
green ideas 1
sleep furiously 1
colorless green ideas 1
```

- Running local MapReduce

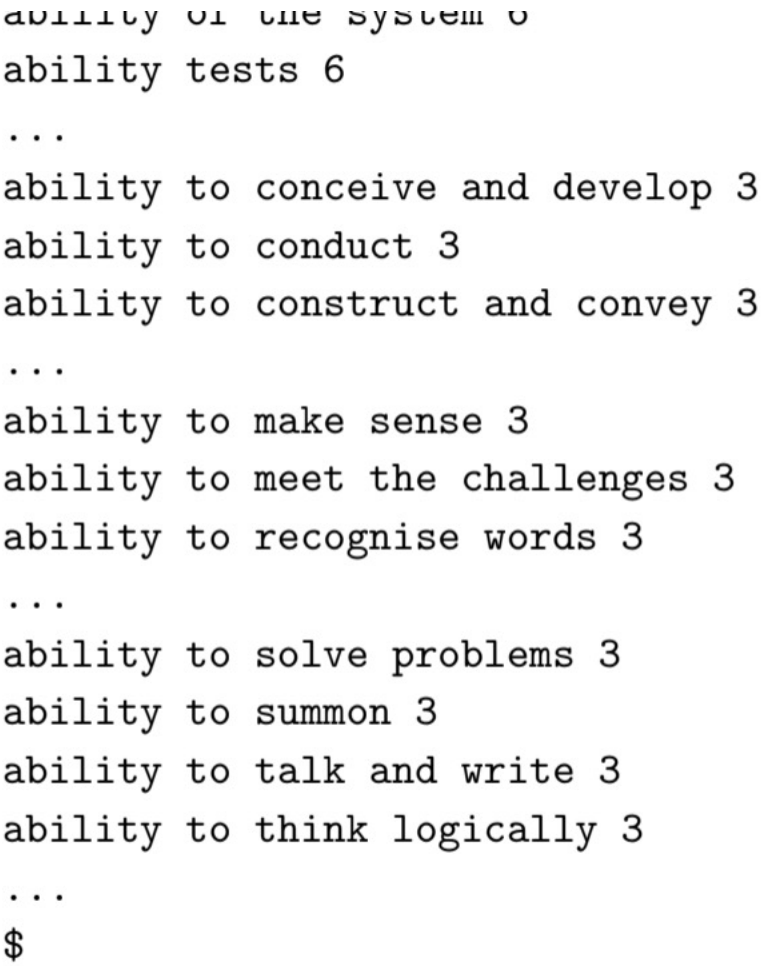


```
echo $'Colorless green ideas \n sleep furiously'
 | ./lmr 5m 16  'python nc-mapper.py' 'python nc-reducer.py' out

hashing script hashing.py.BWar
 >>> Temporary output directory for mapper created: mapper_tmp.YZ4i
 >>> Mappers running...
 >>> Reducer running. Temporary input directory: mapper_tmp.YZ4i
 >>> Cleaning...
 >>> Temporary directory deleted: mapper_tmp.YZ4i
 * Output directory: out
 * Elasped time: 0:00:02

$ cat out/*
sleep furiously 1
colorless green ideas 1
colorless green 1
green ideas 1
```

- Life-size Test on British National Corpus



```
$ time cat bnc.sent.txt | python nc-mapper.py | sort | python nc-reducer.py 3 > bnc

$ grep '^ability ' bnc.ngm.3.plus.txt | sort -k2nr -t $'\t'
ability to pay 108
ability to make 97
ability to cope 64
...
ability range 17
...
ability and willingness 9
...
ability and enthusiasm 6
ability and motivation 6
ability could 6
ability of local 6
```



```
ability of the system 6
ability tests 6
...
ability to conceive and develop 3
ability to conduct 3
ability to construct and convey 3
...
ability to make sense 3
ability to meet the challenges 3
ability to recognise words 3
...
ability to solve problems 3
ability to summon 3
ability to talk and write 3
ability to think logically 3
...
$
```

Task for this week

- TA Announcement
  - Purpose
  - Input
  - Output
  - Mapper
  - Reducer