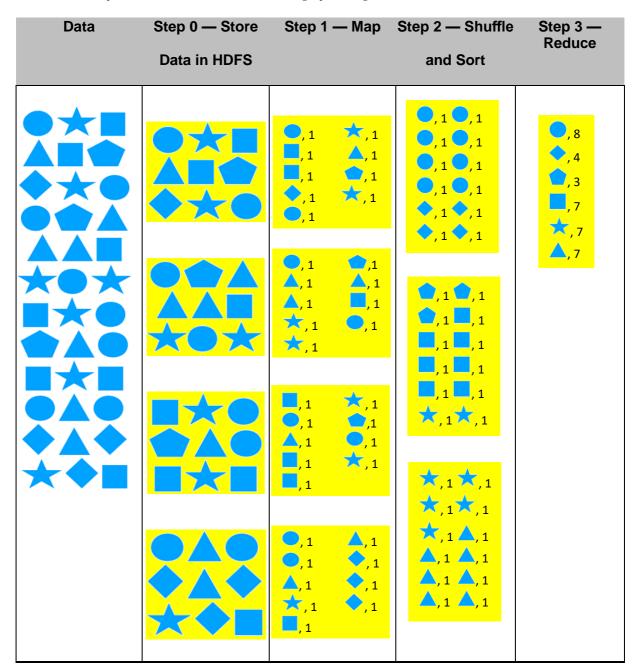
BIGDATA — Assignment 2: MapReduce

Contributors:

Suchi Siwach Muthupriya Shankaran Augustin Ionut Ionascu

Part I: HDFS, Map, Shuffle and Sort, Reduce [6 points]



Part II: Implementing MapReduce in Python [4 points]

(VM path of the containing file: ~/Assignment2/auio3142/part-00000)

The frequencies results are the following:

circle 6973
diamond 10153
pentagon 25145
square 12966
star 4963

triangle 39800

mapper.py script (VM path at ~/Assignment2/auio3142/mapper.py)

```
#!/usr/bin/env python
import sys

# input comes from STDIN (standard input)
for line in sys.stdin:

    # remove trailing spaces at end of each line
    word = line.strip()

# print mapper output to STOUT (standard output): key <tab> value
    print "%s\t%s" % (word, 1)
```

reducer.py script (VM path at ~/Assignment2/auio3142/reducer.py)

```
#!/usr/bin/env python
import sys
# initialize variables
current word = None
current count = 0
word = None
# input comes from STDIN (standard input)
for line in sys.stdin:
        # remove trailing spaces at the end of each line
        line = line.strip()
        # parse the input we got from mapper.py and store as word and count
variables
        word, count = line[:-1].strip(), line[-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: key <tab> value
                        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:
        # write result to STDOUT: key <tab> value
    print "%s\t%s" % (current_word, current_count)
```