CMT202 Distributed and Cloud Computing. Dr. Padraig Corcoran.
Lab title: MapReduce
Learning outcomes: learn to write MapReduce programs using Python.
In this lab you will be using the python library mrjob to write MapReduce programs. For each part of this lab I provide some basic source code for this lab can be downloaded from Learning Central.

Word Count
In the lecture we studied a MapReduce program for determining the frequency of every word in a text document. The code for this program is included below and can downloaded from learning central; it is entitled MRWordCounter.py. To run this program open a terminal in the directory containing the file and enter the command "pipenv run python MRWordCounter.py fileName.txt". Here fileName.txt is the name of the text document for which you wish to perform a word count.

```
from mrjob.job import MRJob

class MRWordCounter(MRJob):
        def mapper(self, key, line):
                        for word in line.split():
                                yield word, 1

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

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

Web Data Processing
In this part we will be using anonymous web browsing data from www.microsoft.com. The data you will be processing can be downloaded from Learning Central and can also be downloaded from the URL https://kdd.ics.uci.edu/databases/msweb/msweb.html. The data in question is entitled anonymous-msweb.data (anonymous-msweb.test is a subset of this data which can be used for testing). A detailed description of the data can be viewed at the URL https://kdd.ics.uci.edu/databases/msweb/msweb.data.html

In summary, the data records the use of www.microsoft.com by 38000 anonymous, randomly-selected users. For each user, the data lists all the areas of the web site (Vroots) that the user visited in a one week timeframe. The data contains different types of records. Each line of the data file starts with a letter which tells the line's record type. The three line types of interest are:

Attribute lines:
These lines begin with an A and describe each web site (Vroots).
For example, 'A,1277,1,"NetShow for PowerPoint","/stream"'
Where:
'A' marks this as an attribute line,
'1277' is the attribute ID number for an area of the website (called a Vroot),
'1' may be ignored,
'"NetShow for PowerPoint"' is the title of the Vroot,
'"/stream"' is the URL relative to "http://www.microsoft.com"

Case and Vote Lines:
For each user (specified by an ID), there is a case line followed by zero or more vote lines. For example:
C,"10164",10164
V,1123,1
V,1009,1
V,1052,1
Where:
'C' marks this as a case line,
'10164' is the case ID number of a user,
'V' marks the vote lines for this case,
'1123', 1009', 1052' are the attribute ID's of Vroots that a user visited.
'1' may be ignored.

Part 1 – Number of visits by Vroot ID
We want to determine the total number of visits to each Vroot. The output should be Vroot ID and the corresponding number of visits if the number of visits for that Vroot is greater than 400.

I have provided a template solution entitled top_pages.py for you to complete. This program can be run using the following terminal command “pipenv run python top_pages.py anonymous-msweb.data”. However you will need to complete the template before the program will run correctly.

Part 2 – Number of visits by title
In the previous part we output a Vroot and corresponding number of visits. We now want to output a title and corresponding number of visits. However the title information is contained on the A lines while visit information is contained on the V lines. Therefore we need to join these two types of data.

We can achieve this by instead of the map function outputting a single value such as 1, we can output a tagged value using a tuple. For example the tuple ('A', "Microsoft Mail") indicates the value "Microsoft Mail" is tagged as an attribute.

I have provided a template solution entitled count_titles.py for you to complete.