Big Data Processing: Assignment 2
Deadline: 14.02.24, 11.55 pm IST
Marks: 10

You are given a file with textual content. You goal is to implement map function from scratch that must have the shuffle and sort component as taught in the lecture 8. The map function will take the text file and will produce the most frequent k key bi-grams (one bi-gram in one line) based on their count in the input file. A key bi-gram is an ordered sequence of two words where the second word appears immediately after the first word in the input text file. In addition, the words in a key bi-gram must have at least three English letters, and must not contain any preposition, conjunction or article. Please also note that all the words must be lowercased, before you count bi-grams. While you tokenize text to get words, word boundaries must be any character other than an English letter or a number.

For example, if we take a small document "IIT Kharagpur is the oldest IIT." The key bi-grams are: (iit kharagpur) and (oldest iit).

Make sure you use python version 3.10 or newer. Your code must implement the map function with sort and shuffle with the user given buffer size.

Also note that the data file is a real life data. Thus, it is highly likely that this file contains unknown characters, corrupt bytes, non-english characters etc. Your code must be able to handle these cases and extract the bi-grams in English.

Path to data file: The data file is attached in moodle.

We will evaluate your program on a linux system from command line with the arguments as follows:

python <your-code.py> <input file> <buffer size> <value of k>

The *buffer size* is an integer which denotes the amount of RAM (in MB) your program must use for creating partitions. This format is very important for evaluation. Your program arguments must follow the sequence. A command line example in linux is given below:
$ python code.py data.txt 1024 20

Submission guidelines:

You need to submit the program as a single python file in moodle. The file name must follow the format: **assignment-2-roll.py** (where the roll denotes your roll number in capital letters that must match exactly with your IITKGP roll number). Please note that if you fail to follow the format, your program may not be evaluated at all.

Important notes:

1. No credit will be given if your program does not run and produces wrong output.
2. No credit will be given if your program does not implement map with sort and shuffle and does not adhere to the specified buffer size.
3. No submission will be accepted after deadline.
4. It is your responsibility to check that the file has been submitted successfully.
5. Plagiarism from friend or from web will invite negative (**-10)** marks.