Big Data Project

Group A

1. Implement a word count and character count program on a Hadoop Cluster (No. of nodes should be equal to number of group members.)

2. Find maximum number from two consecutive columns in file.

Objective:
Implement a map-reduce program to find the maximum integer from two columns in file.
Description:
The input file contains integer values separated by comma. Each line contains 10 integers and there may be 100 number of lines in the file (Generate it Randomly). You should find maximum number from column 1 & 2, 3 & 4, 5 & 6, 7 & 8 and 9 & 10.

Example:

```
8,4,5,6,10,9,2,3,5,6
1,2,4,3,7,6,8,9,4,5
3,5,2,3,7,5,8,9,9,0
.......
5,7,7,4,8,6,5,7,9,0
```

Then output should be:

```
1st & 2nd Column 8
3rd & 4th Column 7
....
9th & 10th Column 9
```

3. Find frequent itemsets in a large transaction file(s).

Objectives:
Write a mapreduce program to find frequent itemsets that appear in the transaction file(s).

Description:
The market-basket model of data is used to describe a common form of many-many relationships between two kinds of objects. On the one hand, we have items, and on the other we have baskets, sometimes called "transactions." Each basket consists of a set of items (an itemset), and usually we assume that the number of items in a basket is small – much smaller than the total number of items. The number of baskets is usually assumed to be very large, bigger than what can fit in main memory. The data is assumed to be represented in a file consisting of a sequence of baskets. The baskets are the objects of the file, and each basket is of type "set of items."

Definition of Frequent Itemsets:
Intuitively, a set of items that appears in many baskets is said to be "frequent". To be formal, we assume there is a number s, called the support threshold. If I is a set of items,

the support for I is the number of baskets for which I is a subset. We say I is frequent if its support is s or more.

4. Find the hottest and coolest year from given weather data.

Objectives:
Design and develop a distributed application to find the coolest/hottest year from the available weather data (Data will be provided). Use weather data from the Internet and process it using MapReduce.

Description:
Sensors senses weather data in big text format containing station ID, year, date, time, temperature, quality etc. from each sensor and store it in single line. Suppose thousands of data sensors are there, then we have thousands of records with no particular order.

5. Implement Movie Recommendation System.

Objective
Using given dataset, find Movie Recommendations using Hadoop MapReduce program.

Description
Recommendation system is a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Imagine that you own an online movie business, and you want to suggest for your client's movie recommendations. Your system runs a rating system, that is, people can rate movies with 1 to 5 stars.
Our goal is to calculate how similar pairs of movies are, so that we recommend movies similar to movies you liked. Using the correlation, we can:

- For every pair of movies A and B, find all the people who rated both A and B.
- Use these ratings to form a Movie X vector and a Movie Y vector.
- Calculate the correlation between those two vectors.
- When someone watches a movie, you can recommend the movies most correlated with it.

You want to compute how similar pairs of movies are, so that if someone watches the movie The Matrix, you can recommend movies like BladeRunner. So how should you define the similarity between two movies?

Dataset

This processing is to be done on the real world Movie Lens dataset (Data set will be provided /also available online). The data sets were collected over various periods of time, depending on the size of the set. It contains 100,000 ratings from 1000 users on 1700 movies. Users were selected at random for inclusion. Users are
represented by its id and item are also represented by id.

6. Develop a MapReduce program by creating custom InputFormat for finding out the number of people and their sex who died and survived from Titanic dataset (Dataset will be provided).

[**Hint** : You may follow following steps:
Step 1: implement a custom key which comprises 2nd and 5th column (composite key)
Step 2: create custom InputFormat
Step 3: implement custom Record Reader
Step 4: mapper using custom key and value
Step 5: reducer using custom key and value
Step 6: driver method]

7. Develop a MapReduce program to get average volume traded for each stock per month from NYSE dataset (Dataset will be provided.)

[**Hint** : You may follow following steps:
a. Develop a parser
b. Identify input and output formats.
c. Develop user-defined key and value data types.
d. Develop a mapper
e. Develop a combiner
f. Develop a reduce task
g. Driver function]

Group -B

8. Develop a MapReduce program having Linked mappers such that it performs:

Job1: wordcount program
job2: counting words that start with the same letter from the output of Job1.

9. Write a MapReduce program to select the maximum temperature for each data.

Input: Two structured textual files containing the temperatures gathered by a set of sensors (Data will be provided).
Each line of the first file has the following format
ID, date, hour, temperature
Each line of the second file has the following format
date, hour, temperature, Id

Output: the maximum temperature for each date, considering the data of both input files

10. Implement a MapReduce based K-means clustering algorithm over a Big Dataset.

11. Perform the following operations in Scala.

I. Generate Random Integers in Array of n Integers having range [0,n].

II. Write a function that swaps adjacent elements in the array created in problem 1.

III. Rewrite the previous task, but this time generate a new array with the swapped values instead of modifying the original array.

IV. Given an array of integers, produce a new array where all positive values appear first in their original order, followed by all zero or negative values, also in their original order.

V. Design a function that takes a non-empty zero-indexed array A of N integers and returns the smallest positive integer (greater than 0) that is not present in A.

Assumptions:

- N is an integer within [1..100,000] ([1,100000]).
- Each element in A falls within [−2,147,483,648..2,147,483,647] .
- The solution should aim for an expected worst-case time complexity of O(N), and it is permissible to modify the elements of the input array."

12. Implement the following program in Scala.

I. Design a "Vehicle" class that includes read-only attributes for the manufacturer, model name, and model year, along with a modifiable attribute for the license plate. Implement four constructors for the class. Each constructor should require the manufacturer and model name as mandatory inputs, while the model year and license plate can be optionally provided. If no values are given for these optional properties, set the model year to "-1" and the license plate to an empty string by default. Identify which of these constructors you would select as the primary one, and explain your reasoning.

II. Implement the same program either in Java or Python and perform a comparison.

III. Given a zero-indexed array A of N integers, a rotation operation shifts each element of the array one position to the right, with the last element wrapping around to the first position. This function should take the array A and integer K, and return the array after rotating it K times.

Assumptions:

- N and K are integers in the range [0..100] ([0,100]).
- Each element in A is within the range [−1, 000..1,000] ([−1000,1000]).

IV. Given a non-empty zero-indexed array A of N integers, where N is odd, each element in the array pairs with another element of the same value, except for one element that

does not have a pair. Write a function that, when given the array A as described, returns the unpaired element.

Assumptions:

- N is an odd integer in the range [1..1,000,000] ([1,1000000]).
- Each element in A is an integer within [1..1,000,000,000] ([1,1000000000]).
- All values in A, except one, appear an even number of times.
- The solution should have an expected worst-case time complexity of O(N).

13. Implement the following Spark program in Scala

I. Write a Spark program that reads data (13_1.txt) and creates a pair RDD where the key is the airport name, and the value is the country in which it is located. Exclude all airports in the USA from this RDD, and save the resulting pair RDD to out/result_13_1.text.

II. Develop a Spark program to read data (13_1.txt) and produce a list of airport names grouped by each country."

III. Write a Spark program to read housing data (13_2.csv) and calculate the average price for houses based on the number of bedrooms.

IV. Create a Spark program to read the data (13_1.txt), find all the airports whose latitude are bigger than 40. Then output the airport's name and the airport's latitude to out/13_4.text.

14. Implement a supervised machine learning algorithm, using Spark over a Big Dataset [Search UCI Repository or Apply Dataset available for Qn-10].

Note:

1. **Each group has a maximum of 5 members.**
2. **Each group will select two questions.**
3. **Each group must have a unique pair of questions from both the groups (One Qn. from Group A and another From Group B).**
4. **None of the questions should be left out.**
5. **Each group member must implement their respective program on their Laptop/Computer.**
6. **Each member must prepare their report individually.**