Experiment 4

Name: Asad Shaikh
Branch: MTech CE
Registration Id: 242050023

Aim: Design MapReduce algorithms to take a very large file of integers and produce as output:
a) The largest Integer.
b) The average of all integers.
c) The same set of integers, but with each integer appearing only once.
d) The count of the number of distinct integers in the input.

Theory:

In Big Data Analytics, the first step of any data analysis process is data collection or ingestion. Data can come from various sources such as:

- Online websites (like Wikipedia, financial data sites, etc.)
- Excel files used in business environments
- CSV (Comma Separated Values) files, a standard format for tabular data

Python, with libraries like pandas, provides powerful tools to read and manipulate these data formats efficiently.

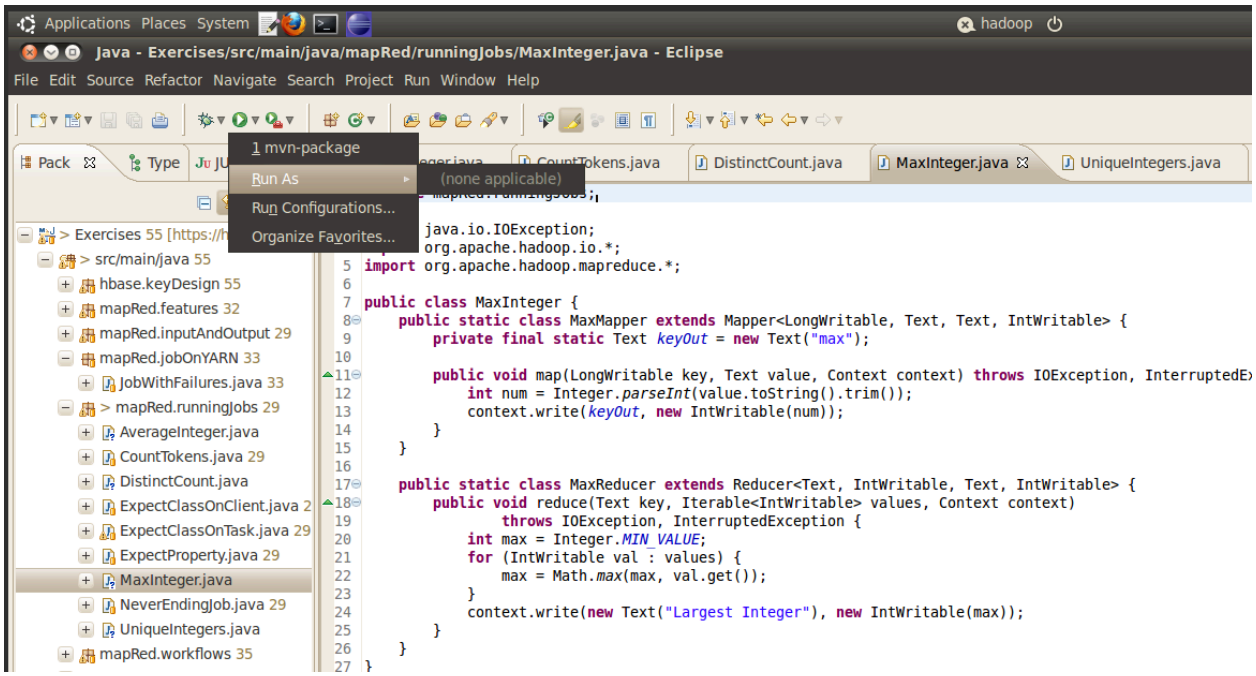
1. Reading Data from an Online Website (HTML Table):
   - Websites often contain tables of structured data (e.g., stock prices, country statistics).
   - **pandas.read_html(url)** reads all tables from a webpage and returns a list of DataFrames.
   - You need an internet connection and a well-structured HTML table for this to work.

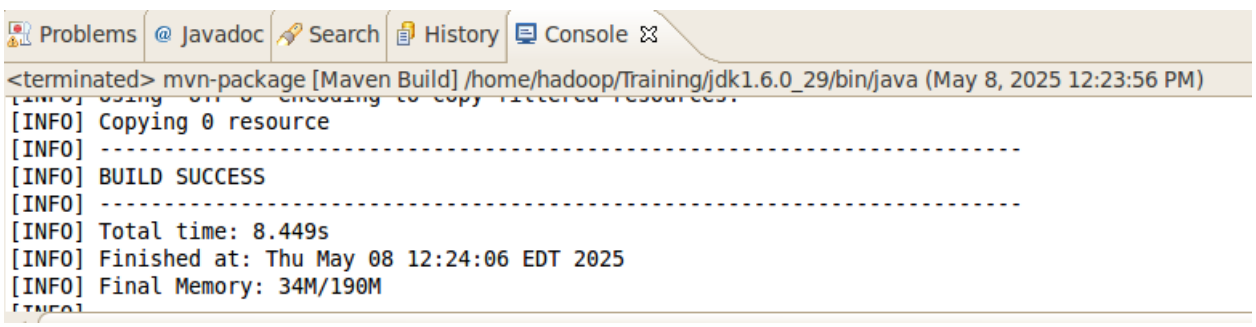
2. Reading Data from an Excel File:
   - Excel is commonly used in data reporting and storage.
   - Python uses **pandas.read_excel()** to load .xlsx files.
   - Requires the openpyxl library (for .xlsx format).

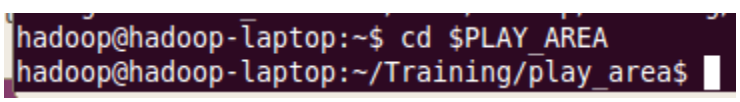
3. Reading Data from a CSV File:
   - CSV is one of the most popular text-based formats for storing tabular data.
   - Each row in the file is a data record, and each field is separated by a comma.
   - Python uses **pandas.read_csv()** to read CSV files easily and efficiently.

Code / Output:

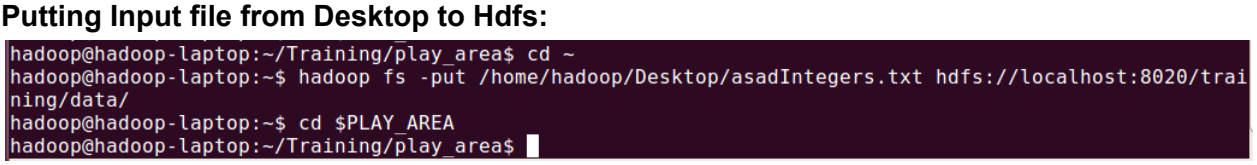
Run as mvn-package:







Putting Input file from Desktop to Hdfs:



Executing MaxInteger.java:

```
hadoop@hadoop-laptop:~$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.Max
Integer /training/data/asadInteger.txt /training/playArea/resultMax2
25/05/08 13:40:39 WARN mapreduce.JobSubmitter: Use GenericOptionsParser for pars
ing the arguments. Applications should implement Tool for the same.
25/05/08 13:40:39 INFO input.FileInputFormat: Total input paths to process : 1
25/05/08 13:40:40 INFO mapreduce.JobSubmitter: number of splits:1
25/05/08 13:40:40 INFO mapred.ResourceMgrDelegate: Submitted application applica
tion_1746725280304_0003 to ResourceManager at localhost/127.0.0.1:10040
25/05/08 13:40:40 INFO mapreduce.Job: The url to track the job: http://localhost
:8088/proxy/application_1746725280304_0003/
25/05/08 13:40:40 INFO mapreduce.Job: Running job: job_1746725280304_0003
25/05/08 13:40:44 INFO mapreduce.Job: Job job_1746725280304_0003 running in uber
 mode : false
25/05/08 13:40:44 INFO mapreduce.Job:  map 0% reduce 0%
25/05/08 13:40:47 INFO mapreduce.Job:  map 100% reduce 0%
25/05/08 13:40:49 INFO mapreduce.Job:  map 100% reduce 100%
25/05/08 13:40:49 INFO mapreduce.Job: Job job_1746725280304_0003 completed succe
ssfully
25/05/08 13:40:49 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=366
```

Reading the output:

```
hadoop@hadoop-laptop:~$ hadoop fs -cat /training/playArea/resultMax2/part-r-0000
0
130
hadoop@hadoop-laptop:~$
```

Output: **130** → This tells that 130 is the Max number in Input Integers

Executing AverageInteger.java:

```
hadoop@hadoop-laptop:~$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.Ave
rageInteger /training/data/asadInteger.txt /training/playArea/resultAverage
25/05/08 13:46:59 WARN mapreduce.JobSubmitter: Use GenericOptionsParser for pars
ing the arguments. Applications should implement Tool for the same.
25/05/08 13:47:00 INFO input.FileInputFormat: Total input paths to process : 1
25/05/08 13:47:00 INFO mapreduce.JobSubmitter: number of splits:1
25/05/08 13:47:00 INFO mapred.ResourceMgrDelegate: Submitted application applica
tion_1746725280304_0005 to ResourceManager at localhost/127.0.0.1:10040
25/05/08 13:47:00 INFO mapreduce.Job: The url to track the job: http://localhost
:8088/proxy/application_1746725280304_0005/
25/05/08 13:47:00 INFO mapreduce.Job: Running job: job_1746725280304_0005
25/05/08 13:47:04 INFO mapreduce.Job: Job job_1746725280304_0005 running in uber
 mode : false
25/05/08 13:47:04 INFO mapreduce.Job:  map 0% reduce 0%
25/05/08 13:47:07 INFO mapreduce.Job:  map 100% reduce 0%
25/05/08 13:47:08 INFO mapreduce.Job:  map 100% reduce 100%
25/05/08 13:47:08 INFO mapreduce.Job: Job job_1746725280304_0005 completed succe
ssfully
25/05/08 13:47:09 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=446
                FILE: Number of bytes written=97656
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
```

Reading the output:

```
hadoop@hadoop-laptop:~$ hadoop fs -cat /training/playArea/resultAverage/part-r-0
0000
avg     69.2
hadoop@hadoop-laptop:~$
```

Output: 69.2 → This is the Average of all the numbers in input integers

Executing DistinctCount.java:

```
hadoop@hadoop-laptop:~$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.Dis
tinctCount /training/data/asadInteger.txt /training/playArea/resultCount
25/05/08 13:51:52 WARN mapreduce.JobSubmitter: Use GenericOptionsParser for pars
ing the arguments. Applications should implement Tool for the same.
25/05/08 13:51:53 INFO input.FileInputFormat: Total input paths to process : 1
25/05/08 13:51:53 INFO mapreduce.JobSubmitter: number of splits:1
25/05/08 13:51:53 INFO mapred.ResourceMgrDelegate: Submitted application applica
tion_1746725280304_0006 to ResourceManager at localhost/127.0.0.1:10040
25/05/08 13:51:53 INFO mapreduce.Job: The url to track the job: http://localhost
:8088/proxy/application_1746725280304_0006/
25/05/08 13:51:53 INFO mapreduce.Job: Running job: job_1746725280304_0006
25/05/08 13:51:59 INFO mapreduce.Job: Job job_1746725280304_0006 running in uber
 mode : false
25/05/08 13:51:59 INFO mapreduce.Job:  map 0% reduce 0%
25/05/08 13:52:02 INFO mapreduce.Job:  map 100% reduce 0%
25/05/08 13:52:04 INFO mapreduce.Job:  map 100% reduce 100%
25/05/08 13:52:04 INFO mapreduce.Job: Job job_1746725280304_0006 completed succe
ssfully
```

Reading the output:

```
hadoop@hadoop-laptop:~$ hadoop fs -cat /training/playArea/resultCount/part-r-000
00
Distinct Count  13
hadoop@hadoop-laptop:~$
```

Output: 13 → There are 13 Distinct Integers in the Input Integers.

Executing UniqueIntegers.java:

```
hadoop@hadoop-laptop:~$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.Uni
queIntegers /training/data/asadInteger.txt /training/playArea/resultUnique
25/05/08 13:54:41 WARN mapreduce.JobSubmitter: Use GenericOptionsParser for pars
ing the arguments. Applications should implement Tool for the same.
25/05/08 13:54:41 INFO input.FileInputFormat: Total input paths to process : 1
25/05/08 13:54:42 INFO mapreduce.JobSubmitter: number of splits:1
25/05/08 13:54:42 INFO mapred.ResourceMgrDelegate: Submitted application applica
tion_1746725280304_0007 to ResourceManager at localhost/127.0.0.1:10040
25/05/08 13:54:42 INFO mapreduce.Job: The url to track the job: http://localhost
:8088/proxy/application_1746725280304_0007/
25/05/08 13:54:42 INFO mapreduce.Job: Running job: job_1746725280304_0007
25/05/08 13:54:47 INFO mapreduce.Job: Job job_1746725280304_0007 running in uber
 mode : false
```

Reading the output:

```
hadoop@hadoop-laptop:~$ hadoop fs -cat /training/playArea/resultUnique/part-r-00
000
9
11
12
23
43
45
67
77
78
98
99
120
130
hadoop@hadoop-laptop:~$
```

Output: The list of all Unique Integers from Input Integers are printed.

Conclusion:
In this experiment, we learned about Hadoop and how it is used to analyze big data. We also learned about the MapReduce programming model which is used to process large amounts of data in parallel. We also implemented four programs, ‘MaxInteger’, ‘AverageInteger’, ‘UniqueIntegers’ and ‘DistinctCount’ in MapReduce.