NoSQL Assignment 2 – Part A

IMT2021008 – Sheikh Muteeb

IMT2021003 – Keshav Chandak

IMT2021007 – Sunny Kaushik

IMT2021076 – Devendara Rishi Nelapati

Problem 1

1. Analyzing M-Counter as a CRDT

Yes, the described M-Counter is a CRDT since it satisfies all four properties needed:

Associativity of merge operation: If we have three M-Counters $x=[x_1,.,x_n]$, $y=[y_1,.,y_n]$, and $z=[z_1,.,z_n]$, then merge(merge(x,y),z) equals $[\max(\max(x_1,y_1),z_1),.,\max(\max(x_n,y_n),z_n)]$. On the other hand, merge(x,merge(y,z)) equals $[\max(x_1,\max(y_1,z_1)),.,\max(x_n,\max(y_n,z_n))]$. Because the maximum function itself is associative, these are equal, and hence we prove the associativity of merge.

Commutativity of merge operation: For any two M-Counters $x=[x_1,.,x_n]$ and $y=[y_1,.,y_n]$, merge(x,y) returns $[\max(x_1,y_1),.,\max(x_n,y_n)]$ whereas merge(y,x) returns $[\max(y_1,x_1),.,\max(y_n,x_n)]$. Since maximum is commutative ($\max(a,b)=\max(b,a)$), these outputs are the same, establishing merge commutativity.

Idempotence of merge operation: For an M-Counter $x=[x_1,.,x_n]$, merge(x,x) returns $[\max(x_1,x_1),.,\max(x_n,x_n)]$. Since $\max(a,a)=a$ for any value, the output reduces to $[x_1,.,x_n]$, which is equal to x, establishing idempotence.

Update monotonicity: In doing y=add(x,c) at server i, with $x=[x_1,.,x_n]$ and $c>0$, we have $y=[x_1,.,x_i+c,.,x_n]$. Evaluation of merge(x,y) produces $[\max(x_1,x_1),.,\max(x_i, x_i+c),.,\max(x_n,x_n)]$, which simplifies to $[x_1,.,x_i+c,.,x_n]$ as $x_i+c>x_i$ for $c>0$. This output equals y , showing updates to be monotonic.

2. State Table Completion

For counters a (server 0) and b (server 1) with n=2:

State	Internal State	Query	History
a0	i:0, n:2, xs:[0,0]	0	{}
a1	i:0, n:2, xs:[1,0]	1	{0}

a2	i:0, n:2, xs:[1,2]	3	{0,1}
b0	i:1, n:2, xs:[0,0]	0	{}
b1	i:1, n:2, xs:[0,2]	2	{1}
b2	i:1, n:2, xs:[0,6]	6	{1,2}
b3	i:1, n:2, xs:[1,6]	7	{0,1,2}

3. CRDT Design Considerations

When designing CRDTs, there exists a tension between formal correctness and intuitive behavior. The optimal approach depends on specific use cases:

Benefits of prioritizing formal correctness:

- Guarantees system consistency and deterministic outcomes
- Provides mathematical certainty about system behavior
- Reduces unexpected states in distributed systems

Drawbacks of strict formal correctness:

- May produce results that feel counterintuitive to users
- Can sacrifice usability for mathematical purity

Benefits of emphasizing application semantics:

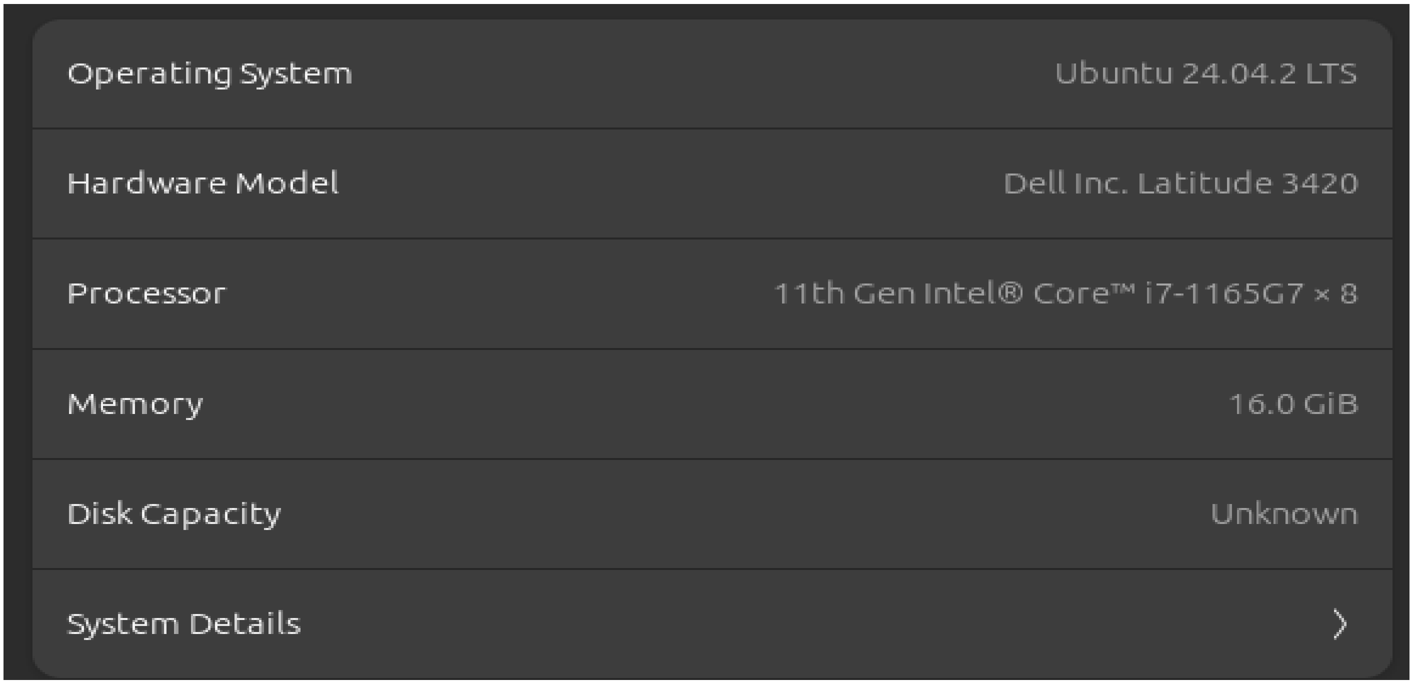
- Creates more intuitive user experiences
- Better aligns with domain-specific requirements
- Improves perceived system responsiveness

Drawbacks of prioritizing semantics:

- Might introduce edge cases where formal properties are weakened
- Could lead to subtly inconsistent behaviors in complex scenarios

The appropriate balance depends on context: mission-critical systems (banking, medical) should favor formal correctness, while collaborative applications (document editing, social media) might benefit from optimizing for intuitive user experience with acceptable consistency trade-offs.

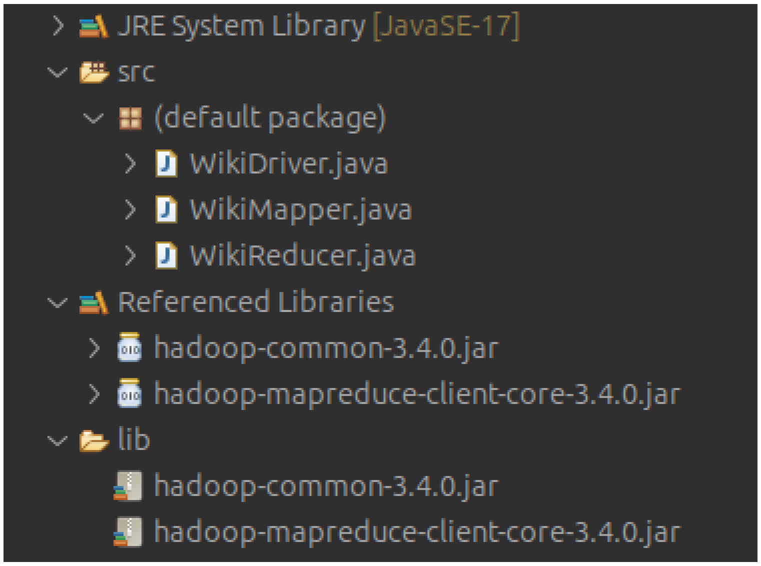
System Configuration Details:



Operating System	Ubuntu 24.04.2 LTS
Hardware Model	Dell Inc. Latitude 3420
Processor	11th Gen Intel® Core™ i7-1165G7 × 8
Memory	16.0 GiB
Disk Capacity	Unknown
System Details	›

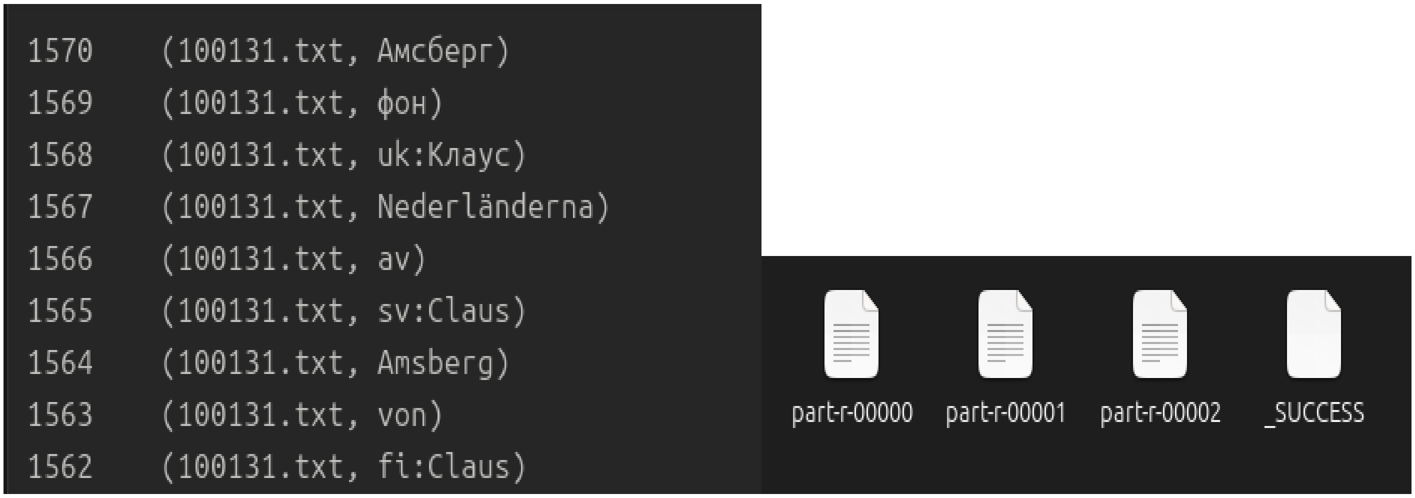
Problem 2

File structure:



```
> JRE System Library [JavaSE-17]
v src
  v (default package)
    > WikiDriver.java
    > WikiMapper.java
    > WikiReducer.java
v Referenced Libraries
  > hadoop-common-3.4.0.jar
  > hadoop-mapreduce-client-core-3.4.0.jar
v lib
    hadoop-common-3.4.0.jar
    hadoop-mapreduce-client-core-3.4.0.jar
```

1. Mapper(WikiMapper.java)
   a. Reads a Wikipedia article, **tokenizes words**, and **emits (index, (docID, word))**.
   b. docID is extracted from the filename.
2. Reducer(WkikReducer.java)
   a. Groups words **by index** and **selects the word from the highest docID**.

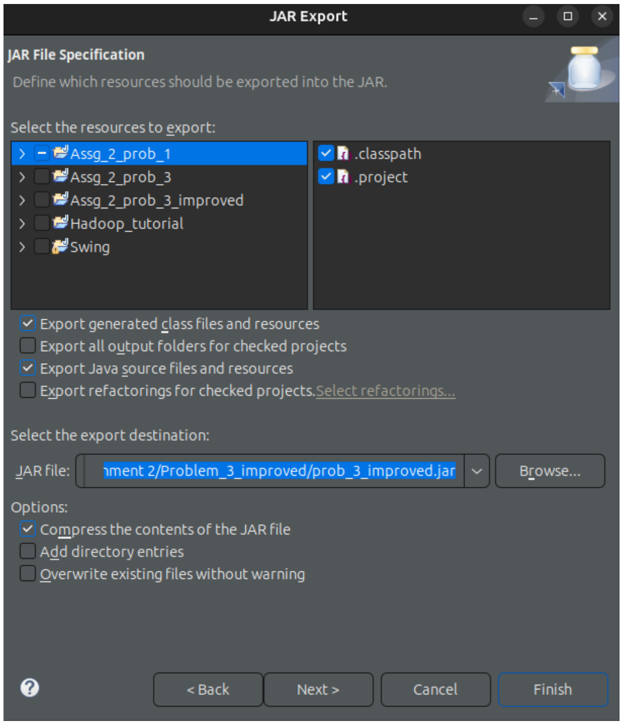


```
1570    (100131.txt, Амсберг)
1569    (100131.txt, фон)
1568    (100131.txt, uk:Клаус)
1567    (100131.txt, Nederländerna)
1566    (100131.txt, av)
1565    (100131.txt, sv:Claus)
1564    (100131.txt, Amsberg)
1563    (100131.txt, von)
1562    (100131.txt, fi:Claus)
```

Example output from the MapReduce program and the files generated

Implementation:

1. Export the whole project as a JAR file to a local directory.



Select the resources to export:

- ▣ Assg_2_prob_1
- □ Assg_2_prob_3
- □ Assg_2_prob_3_improved
- □ Hadoop_tutorial
- □ Swing

- ☑ .classpath
- ☑ .project

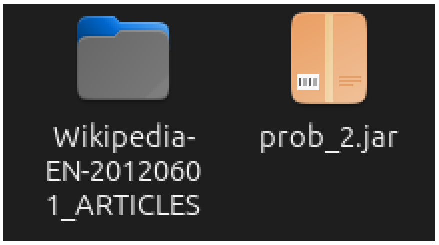
☑ Export generated class files and resources
□ Export all output folders for checked projects
☑ Export Java source files and resources
□ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: ment 2/Problem_3_improved/prob_3_improved.jar

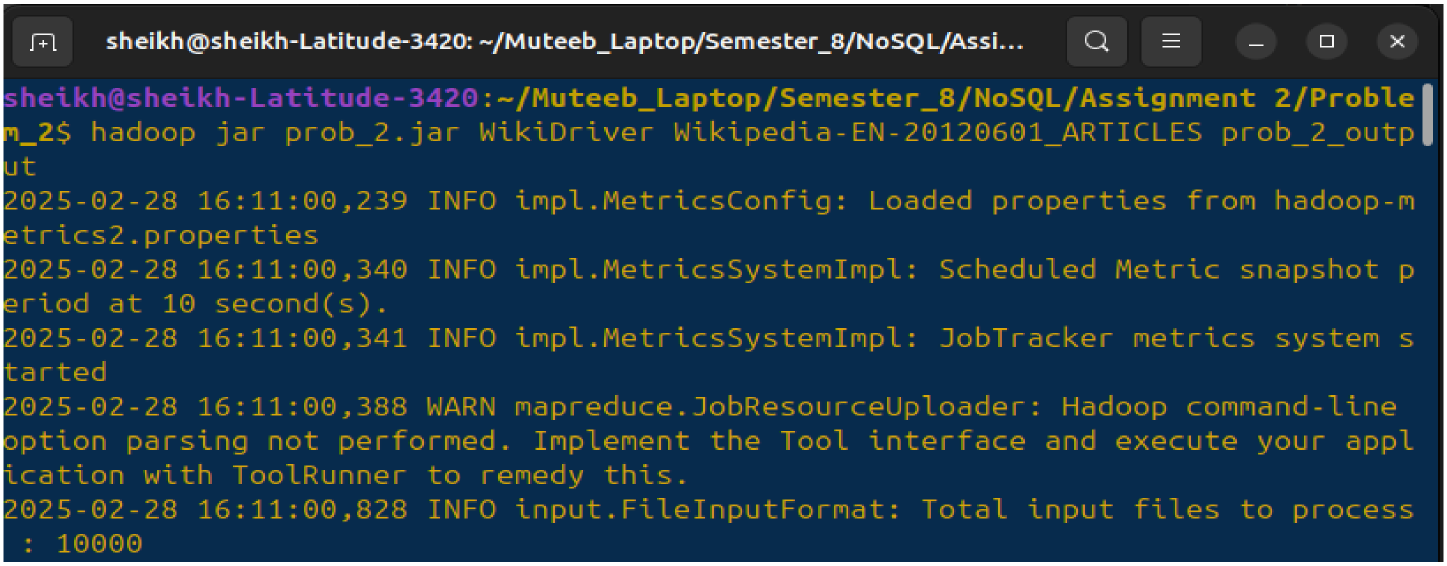
Options:

☑ Compress the contents of the JAR file
□ Add directory entries
□ Overwrite existing files without warning



prob_2.jar is the whole project exported to a local directory

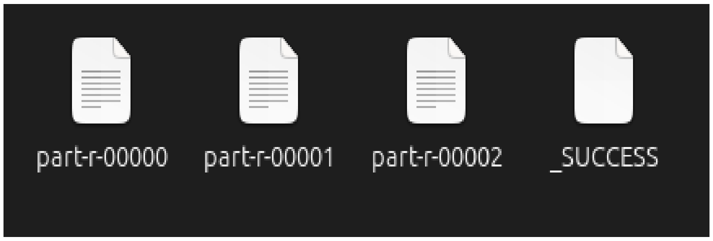
2. Then in the terminal run the following command:

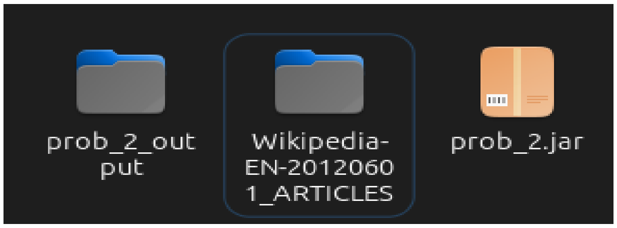


sheikh@sheikh-Latitude-3420: ~/Muteeb_Laptop/Semester_8/NoSQL/Assi...

```
sheikh@sheikh-Latitude-3420:~/Muteeb_Laptop/Semester_8/NoSQL/Assignment 2/Proble
m_2$ hadoop jar prob_2.jar WikiDriver Wikipedia-EN-20120601_ARTICLES prob_2_outp
ut
2025-02-28 16:11:00,239 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-02-28 16:11:00,340 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
eriod at 10 second(s).
2025-02-28 16:11:00,341 INFO impl.MetricsSystemImpl: JobTracker metrics system s
tarted
2025-02-28 16:11:00,388 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2025-02-28 16:11:00,828 INFO input.FileInputFormat: Total input files to process
 : 10000
```

The output gets stored in the prob_2_output folder



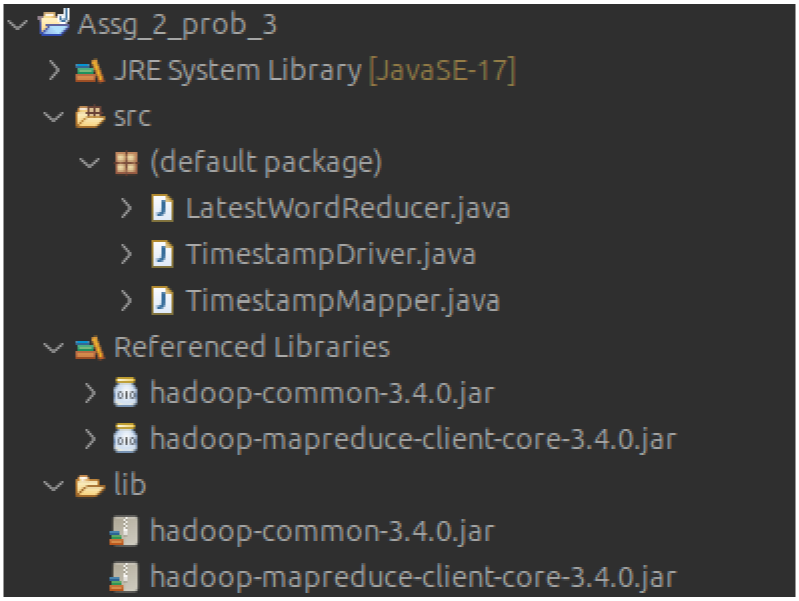


prob_2_output folder contents

Directory after running hadoop mapreduce

Problem 3

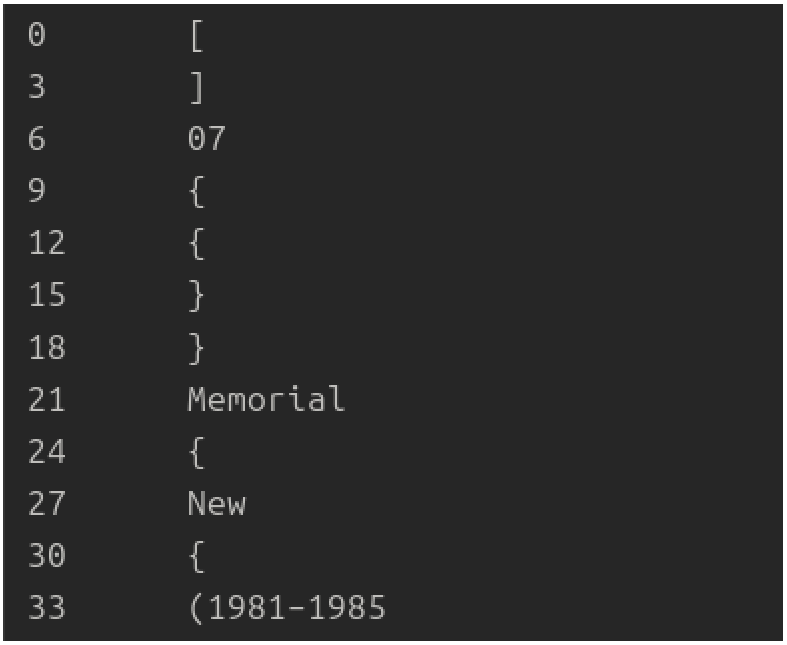
File Structure:



```
Assg_2_prob_3
  JRE System Library [JavaSE-17]
  src
    (default package)
      LatestWordReducer.java
      TimestampDriver.java
      TimestampMapper.java
  Referenced Libraries
    hadoop-common-3.4.0.jar
    hadoop-mapreduce-client-core-3.4.0.jar
  lib
    hadoop-common-3.4.0.jar
    hadoop-mapreduce-client-core-3.4.0.jar
```

1. Mapper (TimestampMapper.java) :
    a. Reads **Problem 2 output** and emits **(index, (timestamp, word))**.

2. Reducer (LatestWordReducer.java):
    a. Retains only words **from the max timestamp (latest revision).**

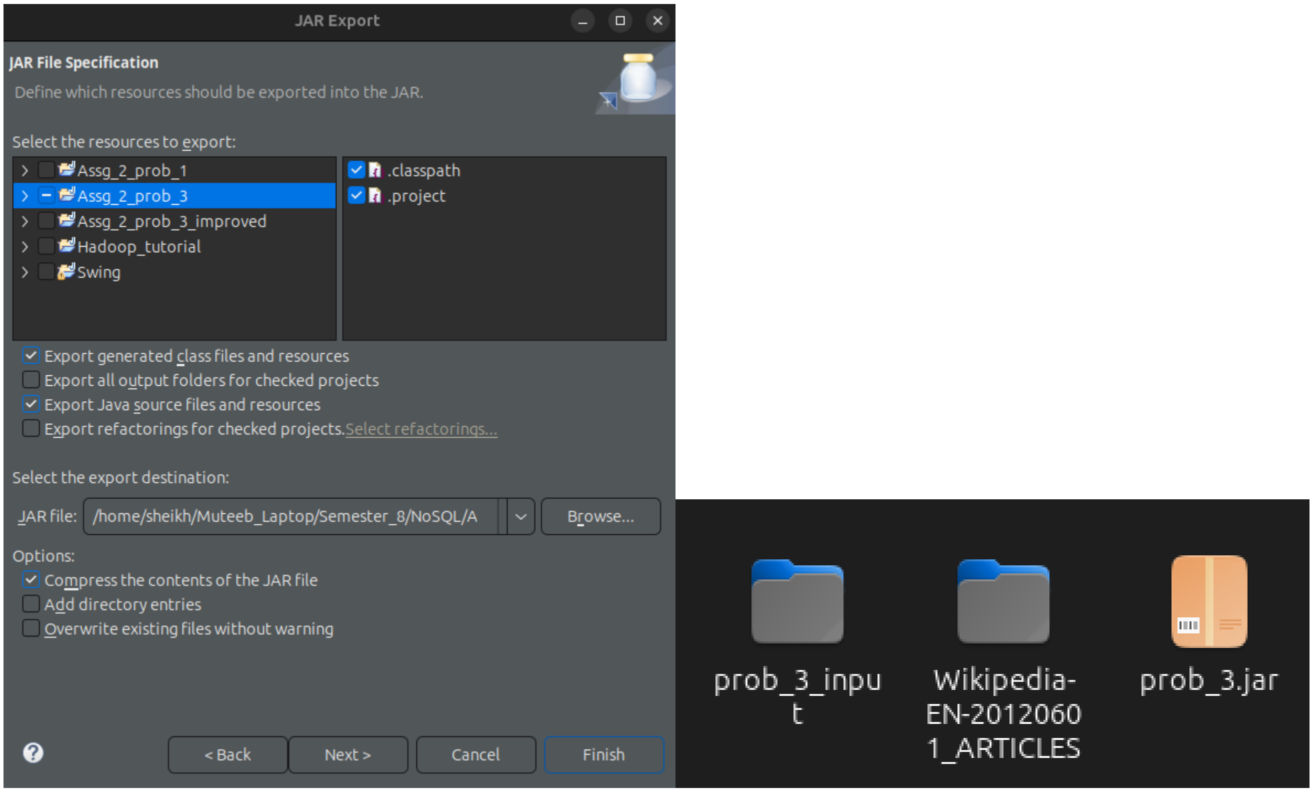


```
0         [
3         ]
6         07
9         {
12        {
15        }
18        }
21        Memorial
24        {
27        New
30        {
33        (1981-1985
```

Contents from the recieved output files

Implementation:

1. Export the whole project as a JAR file to a local directory.



JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- › □ Assg_2_prob_1
- › ▣ Assg_2_prob_3
- › □ Assg_2_prob_3_improved
- › □ Hadoop_tutorial
- › □ Swing

☑ .classpath
☑ .project

☑ Export generated class files and resources
□ Export all output folders for checked projects
☑ Export Java source files and resources
□ Export refactorings for checked projects. Select refactorings...

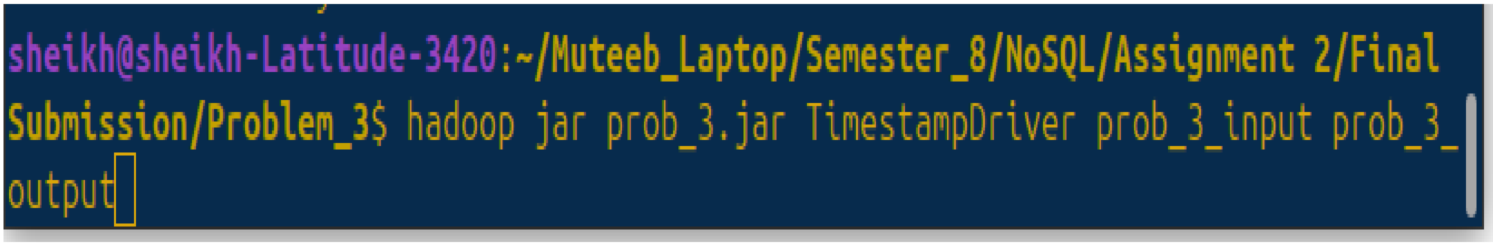
Select the export destination:

JAR file: /home/sheikh/Muteeb_Laptop/Semester_8/NoSQL/A

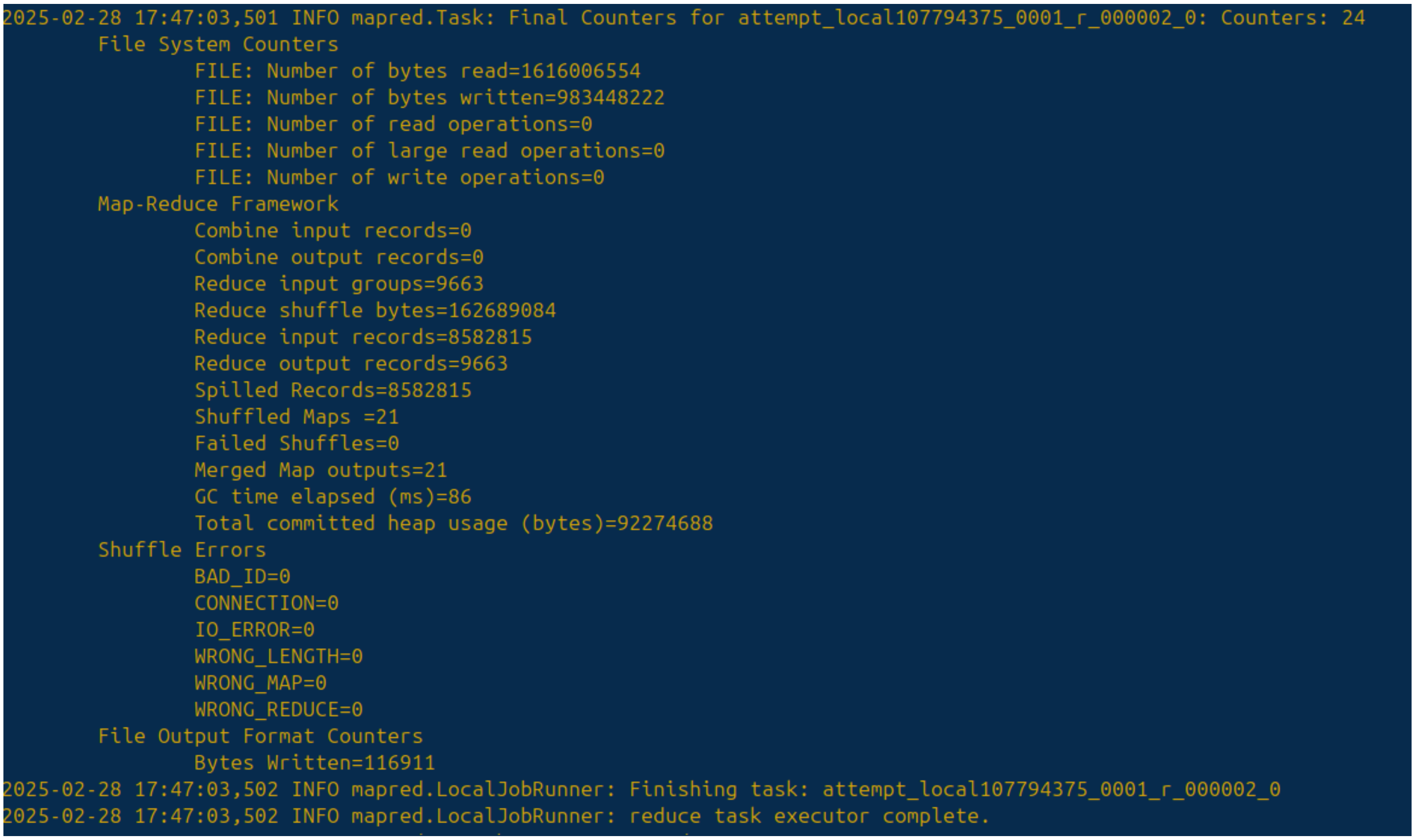
Options:
☑ Compress the contents of the JAR file
□ Add directory entries
□ Overwrite existing files without warning

prob_3.jar is the whole project exported to a local directory with all other required files

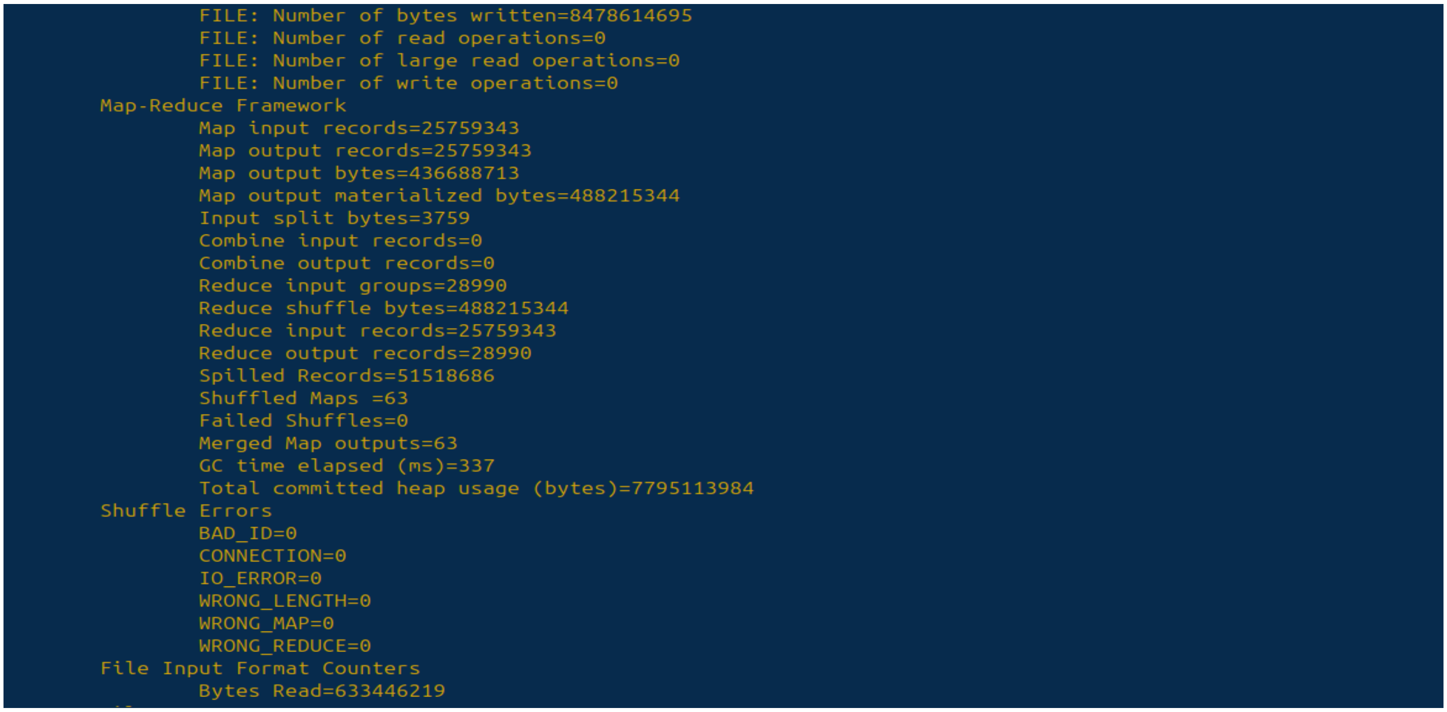
3. Then in the terminal run the following command:



```
sheikh@sheikh-Latitude-3420:~/Muteeb_Laptop/Semester_8/NoSQL/Assignment 2/Final Submission/Problem_3$ hadoop jar prob_3.jar TimestampDriver prob_3_input prob_3_output
```

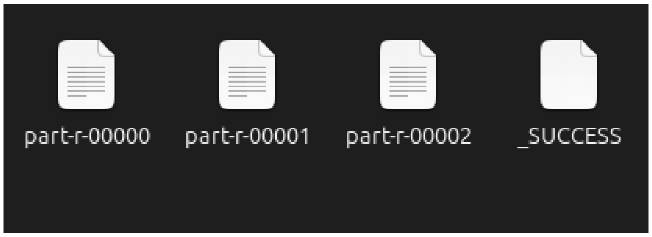


```
2025-02-28 17:47:03,501 INFO mapred.Task: Final Counters for attempt_local107794375_0001_r_000002_0: Counters: 24
        File System Counters
                FILE: Number of bytes read=1616006554
                FILE: Number of bytes written=983448222
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Combine input records=0
                Combine output records=0
                Reduce input groups=9663
                Reduce shuffle bytes=162689084
                Reduce input records=8582815
                Reduce output records=9663
                Spilled Records=8582815
                Shuffled Maps =21
                Failed Shuffles=0
                Merged Map outputs=21
                GC time elapsed (ms)=86
                Total committed heap usage (bytes)=92274688
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=116911
2025-02-28 17:47:03,502 INFO mapred.LocalJobRunner: Finishing task: attempt_local107794375_0001_r_000002_0
2025-02-28 17:47:03,502 INFO mapred.LocalJobRunner: reduce task executor complete.
```

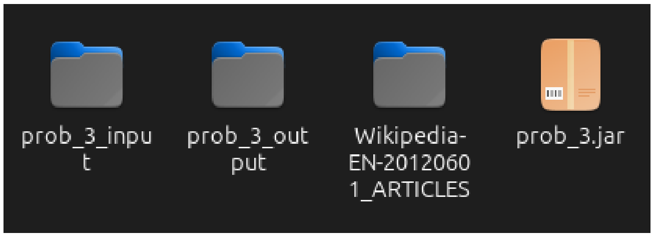


```
		FILE: Number of bytes written=8478614695
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
	Map-Reduce Framework
		Map input records=25759343
		Map output records=25759343
		Map output bytes=436688713
		Map output materialized bytes=488215344
		Input split bytes=3759
		Combine input records=0
		Combine output records=0
		Reduce input groups=28990
		Reduce shuffle bytes=488215344
		Reduce input records=25759343
		Reduce output records=28990
		Spilled Records=51518686
		Shuffled Maps =63
		Failed Shuffles=0
		Merged Map outputs=63
		GC time elapsed (ms)=337
		Total committed heap usage (bytes)=7795113984
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=633446219
```

Above is the successfull execution after running.

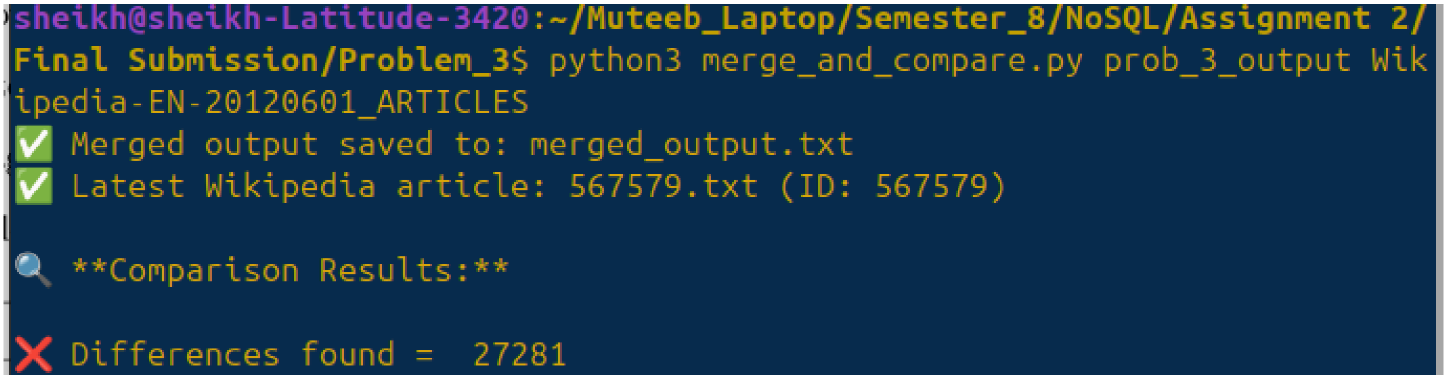


prob_3_output folder contents



Directory after running hadoop mapreduce

4. Running python program to check for differences with the latest wikipedia article:



```
sheikh@sheikh-Latitude-3420:~/Muteeb_Laptop/Semester_8/NoSQL/Assignment 2/Final Submission/Problem_3$ python3 merge_and_compare.py prob_3_output Wikipedia-EN-20120601_ARTICLES
✅ Merged output saved to: merged_output.txt
✅ Latest Wikipedia article: 567579.txt (ID: 567579)

🔍 **Comparison Results:**

❌ Differences found =  27281
```