



CSE488: Big Data Analytics [SPRING 2023]

Lab-6

MapReduce program to find the occurrences of each page, most visited page, least visited page, and frequency of pairs.

Submitted by:

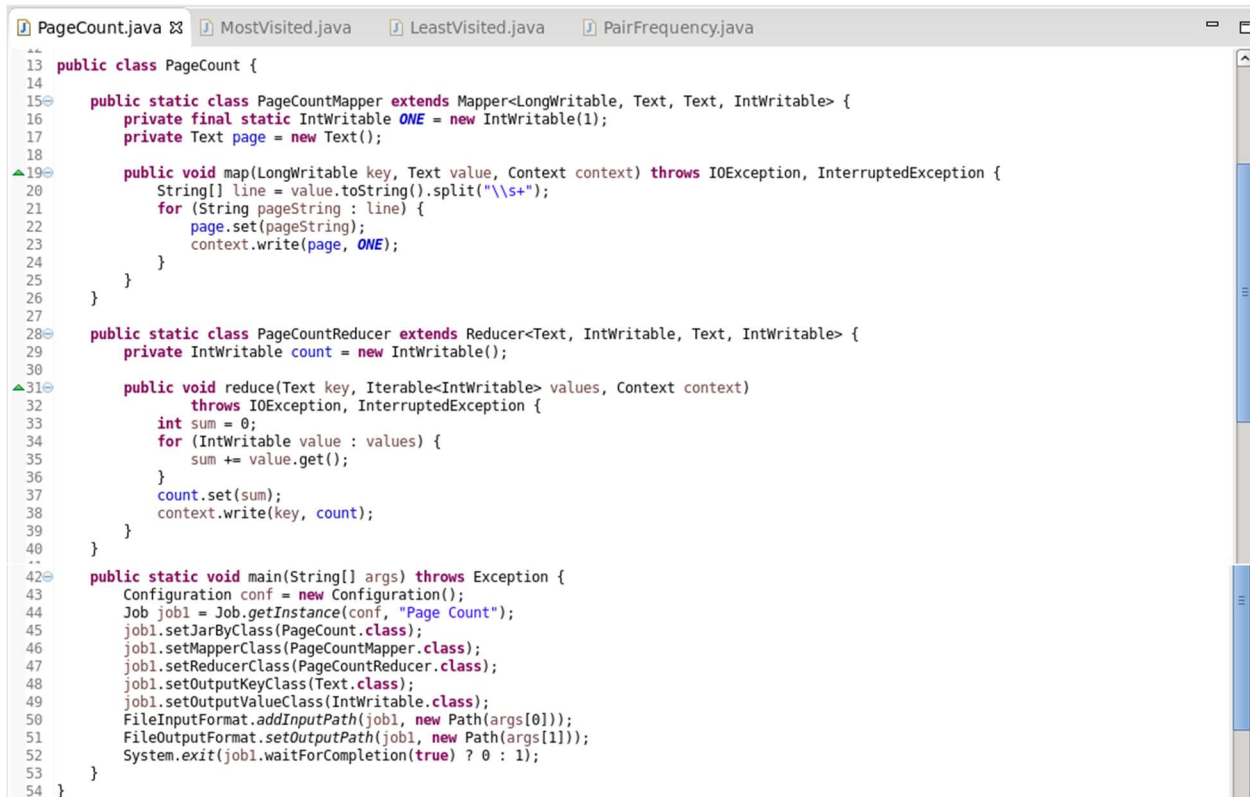
Student ID: 2019-3-60-046

Student Name: Mohsenul Kabir Mithun

Creating a Hadoop project in Eclipse:

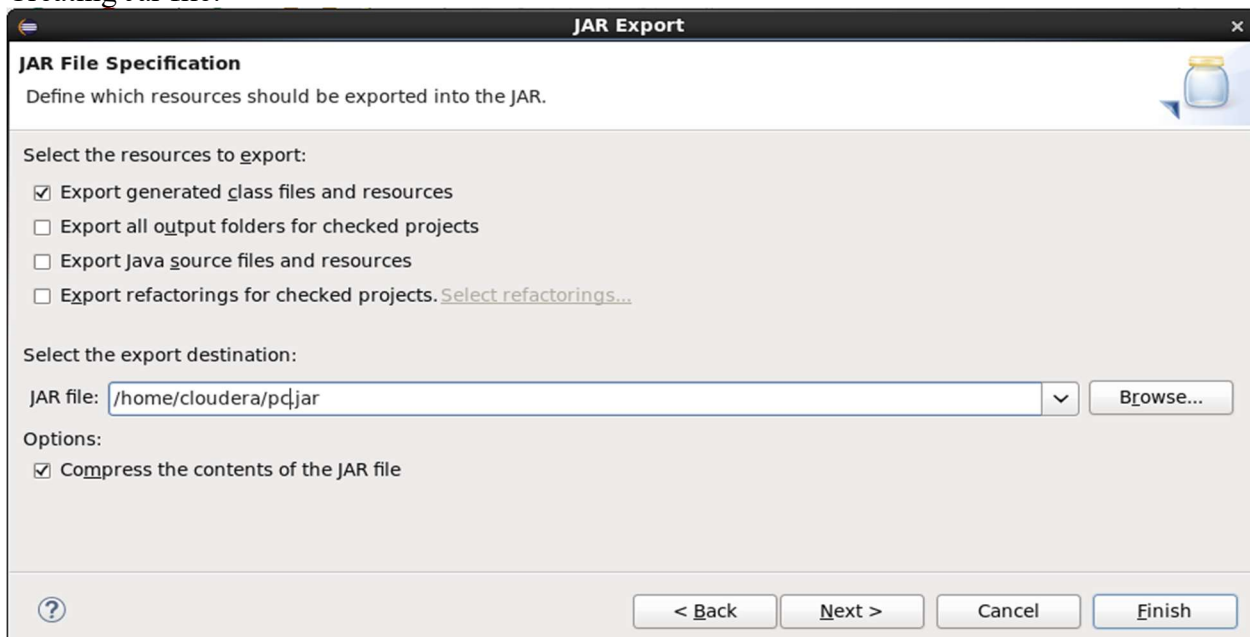
1. Find the occurrences of each page (numbered as 1, 2, 3, and so on).

Source Code SS: (Mapper and Reducer Class)



```
13 public class PageCount {
14
15     public static class PageCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
16         private final static IntWritable ONE = new IntWritable(1);
17         private Text page = new Text();
18
19         public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
20             String[] line = value.toString().split("\\s+");
21             for (String pageString : line) {
22                 page.set(pageString);
23                 context.write(page, ONE);
24             }
25         }
26     }
27
28     public static class PageCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
29         private IntWritable count = new IntWritable();
30
31         public void reduce(Text key, Iterable<IntWritable> values, Context context)
32             throws IOException, InterruptedException {
33             int sum = 0;
34             for (IntWritable value : values) {
35                 sum += value.get();
36             }
37             count.set(sum);
38             context.write(key, count);
39         }
40     }
41
42     public static void main(String[] args) throws Exception {
43         Configuration conf = new Configuration();
44         Job job1 = Job.getInstance(conf, "Page Count");
45         job1.setJarByClass(PageCount.class);
46         job1.setMapperClass(PageCountMapper.class);
47         job1.setReducerClass(PageCountReducer.class);
48         job1.setOutputKeyClass(Text.class);
49         job1.setOutputValueClass(IntWritable.class);
50         FileInputFormat.addInputPath(job1, new Path(args[0]));
51         FileOutputFormat.setOutputPath(job1, new Path(args[1]));
52         System.exit(job1.waitForCompletion(true) ? 0 : 1);
53     }
54 }
```

Creating Jar file:



Creating Input file:

The screenshot shows the Cloudera Desktop interface. On the left, there's a sidebar with 'ACTIONS' (View as binary, Download, View file location, Refresh) and 'INFO' (Last modified: March 30, 2023 12:09). The main area shows the file path '/ user / cloudera / PageCount / msnbc_1.txt'. The file content is displayed in a text area:

```
1 1
2
3 2 2 4 2 2 2 3 3
5
1
6
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
```

CMD run the program:

```
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop jar pc.jar PageCount /user/cloudera/PageCount /user/cloudera/PageCount/PageCountOut
23/03/31 02:13:07 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/31 02:13:07 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
and execute your application with ToolRunner to remedy this.
23/03/31 02:13:08 INFO input.FileInputFormat: Total input paths to process : 2
23/03/31 02:13:08 INFO mapreduce.JobSubmitter: number of splits:2
23/03/31 02:13:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678613146551_0051
23/03/31 02:13:09 INFO impl.YarnClientImpl: Submitted application application_1678613146551_0051
23/03/31 02:13:09 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1678613146551_0051
23/03/31 02:13:09 INFO mapreduce.Job: Running job: job_1678613146551_0051
23/03/31 02:13:21 INFO mapreduce.Job: Job job_1678613146551_0051 running in uber mode : false
23/03/31 02:13:21 INFO mapreduce.Job: map 0% reduce 0%
23/03/31 02:13:41 INFO mapreduce.Job: map 50% reduce 0%
23/03/31 02:13:42 INFO mapreduce.Job: map 83% reduce 0%
23/03/31 02:13:43 INFO mapreduce.Job: map 100% reduce 0%
```

```
File Edit View Search Terminal Help
Reduce shuffle bytes=6485650
Reduce input records=785273
Reduce output records=17
Spilled Records=1570546
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=714
CPU time spent (ms)=6750
Physical memory (bytes) snapshot=550178816
Virtual memory (bytes) snapshot=4509777920
Total committed heap usage (bytes)=391979008
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=2103945
File Output Format Counters
Bytes Written=143
[cloudera@quickstart ~]$ S
```

Output:

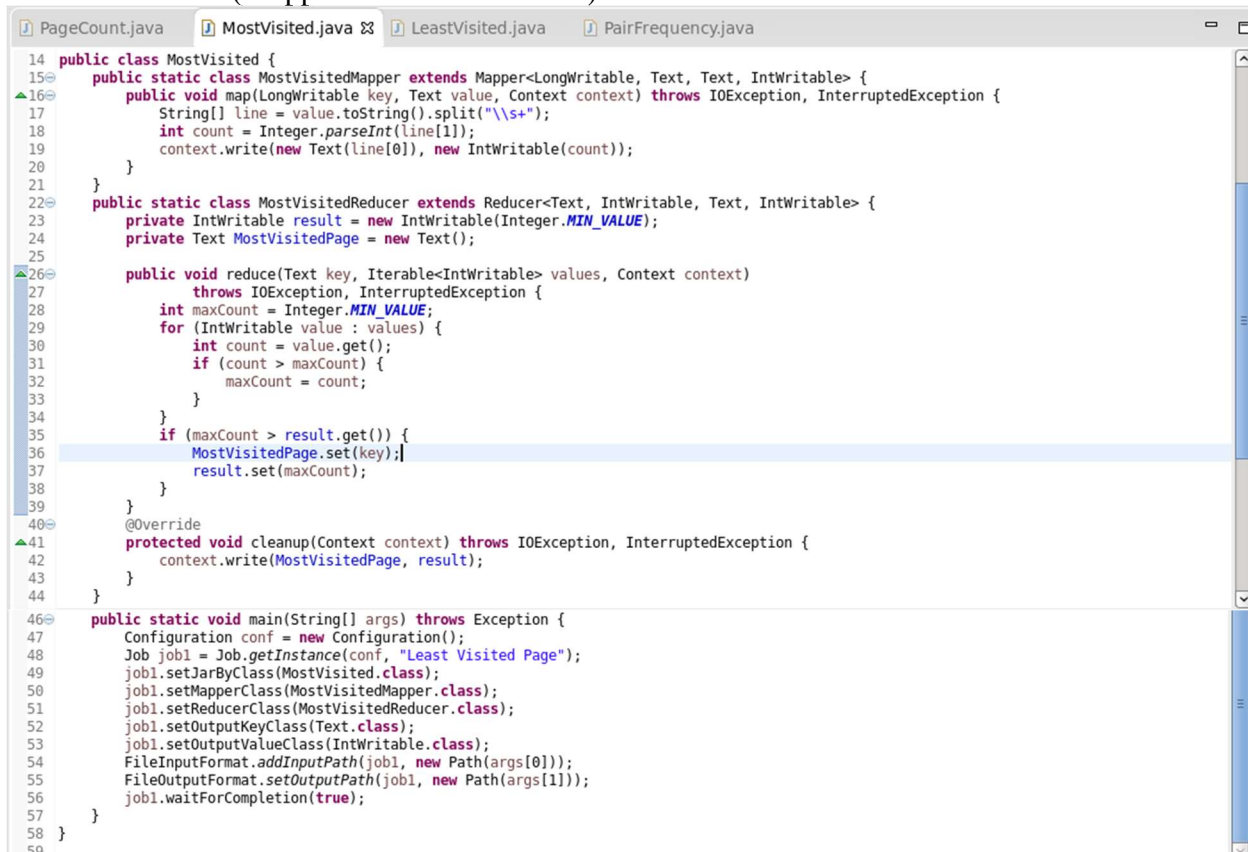


The screenshot shows a web browser interface with a navigation bar at the top. On the left is a 'Home' button with a house icon. On the right is a 'Page' indicator showing '1 of 1' and four navigation arrows (back, forward, etc.). Below the navigation bar is a breadcrumb trail: '/ user / cloudera / PageCount / PageCountOut / part-r-00000'. The main content area displays a list of page visit counts. The list is organized into two columns: the first column contains page numbers and the second column contains their corresponding visit counts. The data is as follows:

Page Number	Visit Count
1	157807
10	22559
11	16380
12	45173
13	35768
14	66621
15	10999
16	3122
17	2832
2	75962
3	33788

2. Find the most visited page. You may use the output file generated from Question 1 as an input file to solve this problem.

Source Code SS: (Mapper and Reducer Class)



The screenshot shows a code editor with four tabs: PageCount.java, MostVisited.java (selected), LeastVisited.java, and PairFrequency.java. The code in MostVisited.java is as follows:

```
14 public class MostVisited {
15     public static class MostVisitedMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
16         public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
17             String[] line = value.toString().split("\\s+");
18             int count = Integer.parseInt(line[1]);
19             context.write(new Text(line[0]), new IntWritable(count));
20         }
21     }
22     public static class MostVisitedReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
23         private IntWritable result = new IntWritable(Integer.MIN_VALUE);
24         private Text MostVisitedPage = new Text();
25
26         public void reduce(Text key, Iterable<IntWritable> values, Context context)
27             throws IOException, InterruptedException {
28             int maxCount = Integer.MIN_VALUE;
29             for (IntWritable value : values) {
30                 int count = value.get();
31                 if (count > maxCount) {
32                     maxCount = count;
33                 }
34             }
35             if (maxCount > result.get()) {
36                 MostVisitedPage.set(key);
37                 result.set(maxCount);
38             }
39         }
40         @Override
41         protected void cleanup(Context context) throws IOException, InterruptedException {
42             context.write(MostVisitedPage, result);
43         }
44     }
45
46     public static void main(String[] args) throws Exception {
47         Configuration conf = new Configuration();
48         Job job1 = Job.getInstance(conf, "Least Visited Page");
49         job1.setJarByClass(MostVisited.class);
50         job1.setMapperClass(MostVisitedMapper.class);
51         job1.setReducerClass(MostVisitedReducer.class);
52         job1.setOutputKeyClass(Text.class);
53         job1.setOutputValueClass(IntWritable.class);
54         FileInputFormat.addInputPath(job1, new Path(args[0]));
55         FileOutputFormat.setOutputPath(job1, new Path(args[1]));
56         job1.waitForCompletion(true);
57     }
58 }
59
```

Creating Jar file:

JAR File Specification
Define which resources should be exported into the JAR.

Select the resources to export:

☒ 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: Browse...

Options:

? < Back Next > Cancel Finish

Creating Input file:

ACTIONS
View as binary
Download
View file location
Refresh

INFO
Last modified
March 30, 2023 12:09

Home

Page 1 to 1 of 514

/ user / cloudera / PageCount / msnbc_1.txt

```
1 1
2
3 2 2 4 2 2 2 3 3
5
1
6
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
```

CMD run the program:

```
[cloudera@quickstart ~]$ hadoop jar mv.jar MostVisited /user/cloudera/PageCount/PageCountOut /user/cloudera/PageCount/MostVisitedOut
23/03/30 21:09:20 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/30 21:09:20 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/30 21:09:21 INFO input.FileInputFormat: Total input paths to process : 1
23/03/30 21:09:21 INFO mapreduce.JobSubmitter: number of splits:1
23/03/30 21:09:21 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678613146551_0008
23/03/30 21:09:21 INFO impl.YarnClientImpl: Submitted application application_1678613146551_0008
23/03/30 21:09:21 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1678613146551_0008/
23/03/30 21:09:21 INFO mapreduce.Job: Running job: job_1678613146551_0008
23/03/30 21:09:32 INFO mapreduce.Job: Job job_1678613146551_0008 running in uber mode : false
23/03/30 21:09:32 INFO mapreduce.Job: map 0% reduce 0%
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
Reduce shuffle bytes=150  
Reduce input records=17  
Reduce output records=1  
Spilled Records=34  
Shuffled Maps =1  
Failed Shuffles=0  
Merged Map outputs=1  
GC time elapsed (ms)=194  
CPU time spent (ms)=1410  
Physical memory (bytes) snapshot=366968832  
Virtual memory (bytes) snapshot=3008630784  
Total committed heap usage (bytes)=226365440  
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=143  
File Output Format Counters  
Bytes Written=9  
[cloudera@quickstart ~]$
```

Output:

Home Page 1 of 1

/ user / cloudera / PageCount / MostVisitedOut / part-r-00000

1	157807
---	--------

3. Find the least visited page. You may use the output file generated from Question 1 as an input file to solve this problem.

Source Code SS: (Mapper and Reducer Class)

```
PageCount.java MostVisited.java LeastVisited.java PairFrequency.java  
14 public class LeastVisited {  
15     public static class LeastVisitedMapper extends Mapper<LongWritable, Text, Text, IntWritable> {  
16         public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {  
17             String[] line = value.toString().split("\\s+");  
18             int count = Integer.parseInt(line[1]);  
19             context.write(new Text(line[0]), new IntWritable(count));  
20         }  
21     }  
22     public static class LeastVisitedReducer extends Reducer<Text, IntWritable, Text, IntWritable> {  
23         private IntWritable result = new IntWritable(Integer.MAX_VALUE);  
24         private Text leastVisitedPage = new Text();  
25     }  
26     public void reduce(Text key, Iterable<IntWritable> values, Context context)  
27         throws IOException, InterruptedException {  
28         int minCount = Integer.MAX_VALUE;  
29         for (IntWritable value : values) {  
30             int count = value.get();  
31             if (count < minCount) {  
32                 minCount = count;  
33             }  
34         }  
35         if (minCount < result.get()) {  
36             leastVisitedPage.set(key);  
37             result.set(minCount);  
38         }  
39     }  
40     @Override  
41     protected void cleanup(Context context) throws IOException, InterruptedException {  
42         context.write(leastVisitedPage, result);  
43     }  
44 }
```

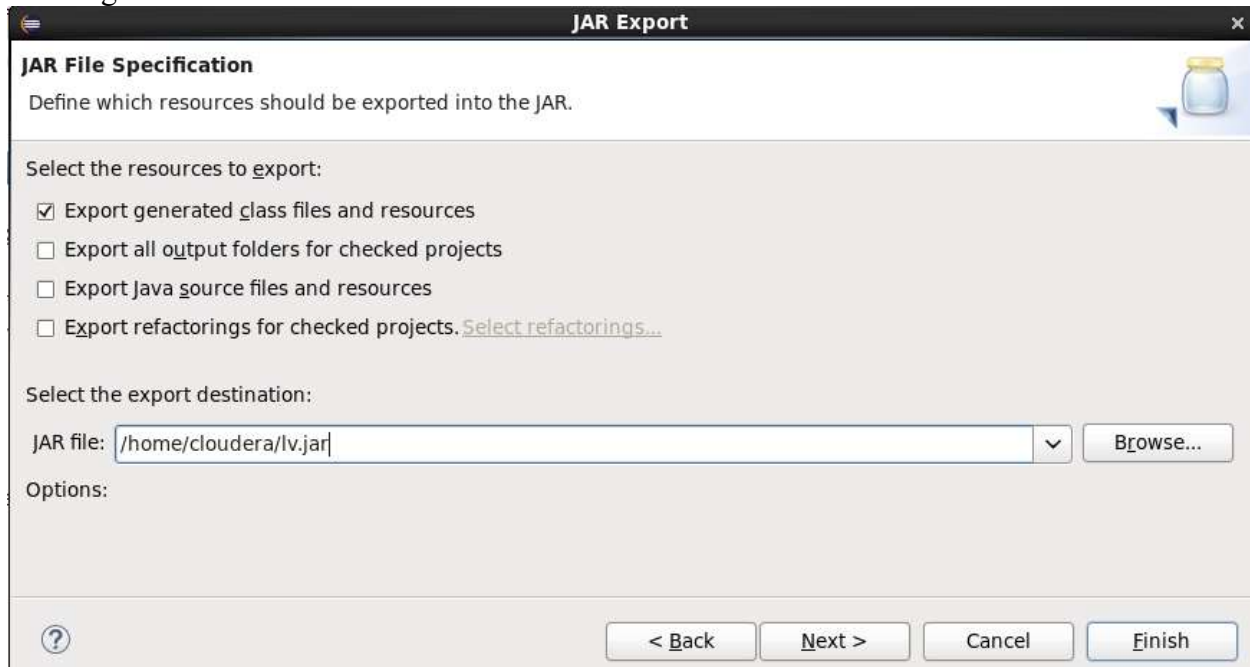


```

46 public static void main(String[] args) throws Exception {
47     Configuration conf = new Configuration();
48     Job job1 = Job.getInstance(conf, "Least Visited Page");
49     job1.setJarByClass(LeastVisited.class);
50     job1.setMapperClass(LeastVisitedMapper.class);
51     job1.setReducerClass(LeastVisitedReducer.class);
52     job1.setOutputKeyClass(Text.class);
53     job1.setOutputValueClass(IntWritable.class);
54     FileInputFormat.addInputPath(job1, new Path(args[0]));
55     FileOutputFormat.setOutputPath(job1, new Path(args[1]));
56     job1.waitForCompletion(true);
57 }
58 }

```

Creating Jar file:



JAR Export

JAR File Specification
Define which resources should be exported into the JAR.

Select the resources to export:

- ☒ 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: Browse...

Options:

? < Back Next > Cancel Finish

Creating Input file:



ACTIONS

- View as binary
- Download
- View file location
- Refresh

INFO

Last modified
March 30,
2023 12:09

Home Page 1 to 1 of 514

/ user / cloudera / PageCount / msnbc_1.txt

```

1 1
2
3 2 2 4 2 2 2 3 3
5
1
6
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1

```

CMD run the program:

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hadoop jar lv.jar LeastVisited /user/cloudera/PageCount/PageCountOut /user/cloudera/PageCount/LeastVisitedOut  
23/03/31 02:30:05 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032  
23/03/31 02:30:06 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.  
23/03/31 02:30:07 INFO input.FileInputFormat: Total input paths to process : 1  
23/03/31 02:30:07 INFO mapreduce.JobSubmitter: number of splits:1  
23/03/31 02:30:07 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678613146551_0054  
23/03/31 02:30:08 INFO impl.YarnClientImpl: Submitted application application_1678613146551_0054  
23/03/31 02:30:08 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1678613146551_0054/  
23/03/31 02:30:08 INFO mapreduce.Job: Running job: job_1678613146551_0054  
23/03/31 02:30:18 INFO mapreduce.Job: Job job_1678613146551_0054 running in uber mode : false
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
Reduce shuffle bytes=150  
Reduce input records=17  
Reduce output records=1  
Spilled Records=34  
Shuffled Maps =1  
Failed Shuffles=0  
Merged Map outputs=1  
GC time elapsed (ms)=266  
CPU time spent (ms)=1570  
Physical memory (bytes) snapshot=364797952  
Virtual memory (bytes) snapshot=3008925696  
Total committed heap usage (bytes)=226365440  
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=143  
File Output Format Counters  
Bytes Written=8  
[cloudera@quickstart ~]$
```

Output:

Home

Page 1 of 1

/ user / cloudera / PageCount / LeastVisitedOut / part-r-00000

17	2832
----	------

4. Find the frequency of pairs. In this case, remove the duplicate numbers from each transaction as a data pre-processing step. Use the concept of Java set for removing duplicates.

Source Code SS: (Mapper and Reducer Class)


```

16 public class PairFrequency {
17     public static class SumMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
18         public void map(LongWritable key, Text value, Context context)
19             throws IOException, InterruptedException {
20             String[] fields = value.toString().split(" ");
21             Set<String> uniqueFields = new HashSet<>(Arrays.asList(fields));
22             String[] uniqueFieldsArray = uniqueFields.toArray(new String[uniqueFields.size()]);
23             Arrays.sort(uniqueFieldsArray);
24             int j = 0;
25             String f = "";
26             for (int i = 0; i < uniqueFieldsArray.length; i++) {
27                 for (j = i + 1; j < uniqueFieldsArray.length; j++) {
28                     if (uniqueFieldsArray[i].equals(uniqueFieldsArray[j]) || f.equals(uniqueFieldsArray[j])) {
29                         continue;
30                     }
31                     f = uniqueFieldsArray[j];
32                     String a = uniqueFieldsArray[i].toString().trim() + " " + uniqueFieldsArray[j].toString().trim();
33                     context.write(new Text(a), new IntWritable(1));
34                 }
35             }
36         }
37     }
38     public static class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
39         public void reduce(Text key, Iterable<IntWritable> values, Context context)
40             throws IOException, InterruptedException {
41             int sum = 0;
42             for (IntWritable val : values) {
43                 sum += val.get();
44             }
45             context.write(key, new IntWritable(sum));
46         }
47     }
48     public static void main(String[] args) throws Exception {
49         Job job = Job.getInstance();
50         job.setJarByClass(PairFrequency.class);
51         job.setJobName("Pair Frequency Count");
52         FileInputFormat.addInputPath(job, new Path(args[0]));
53         FileOutputFormat.setOutputPath(job, new Path(args[1]));
54         job.setMapperClass(SumMapper.class);
55         job.setReducerClass(SumReducer.class);
56         job.setOutputKeyClass(Text.class);
57         job.setOutputValueClass(IntWritable.class);
58         System.exit(job.waitForCompletion(true) ? 0 : 1);
59     }
60 }
61 }

```

Creating Jar file:

JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

☒ 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/cloudera/pf.jar

Browse...

Options:

?

< Back

Next >

Cancel

Finish

Creating Input file:

ACTIONS

- View as binary
- Download
- View file location
- Refresh

INFO

Last modified
March 30,
2023 12:09

Home Page 1 to 1 of 514

/ user / cloudera / PageCount / msnbc_1.txt

```
1 1
2
3 2 2 4 2 2 3 3
5
1
6
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
```

CMD run the program:

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
23/03/31 02:53:29 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.  
23/03/31 02:53:29 INFO input.FileInputFormat: Total input paths to process : 1  
23/03/31 02:53:29 INFO mapreduce.JobSubmitter: number of splits:1  
23/03/31 02:53:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678613146551_0058  
23/03/31 02:53:30 INFO impl.YarnClientImpl: Submitted application application_1678613146551_0058  
23/03/31 02:53:30 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1678613146551_0058/  
23/03/31 02:53:30 INFO mapreduce.Job: Running job: job_1678613146551_0058
```

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
Reduce shuffle bytes=1995875  
Reduce input records=187742  
Reduce output records=136  
Spilled Records=375484  
Shuffled Maps =1  
Failed Shuffles=0  
Merged Map outputs=1  
GC time elapsed (ms)=233  
CPU time spent (ms)=3970  
Physical memory (bytes) snapshot=367357952  
Virtual memory (bytes) snapshot=3008917504  
Total committed heap usage (bytes)=226365440  
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=2103902  
File Output Format Counters  
Bytes Written=1260  
[cloudera@quickstart ~]$
```

Output:

 Home

Page 1 of 1



/ [user](#) / [cloudera](#) / [PageCount](#) / [PairFrequencyOut](#) / **part-r-00000**

1	10	4287
1	11	5494
1	12	7138
1	13	661
1	14	6526
1	15	1512
1	16	191
1	17	1262
1	2	12477
1	3	5497
1	4	6578

