Assignment 3

MaxTemperatureMapper.java

```
package MaxMinTemp;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class MaxTemperatureMapper extends
 Mapper<LongWritable, Text, Text, IntWritable> {
private static final int MISSING = 9999;
@Override
 public void map(LongWritable key, Text value, Context context)
 throws IOException, InterruptedException {
String line = value.toString();
 String year = line.substring(15, 19);
 int airTemperature;
 if (line.charAt(87) == '+') { // parseInt doesn't like leading plus
 // signs
 airTemperature = Integer.parseInt(line.substring(88, 92));
 } else {
 airTemperature = Integer.parseInt(line.substring(87, 92));
 }
 String quality = line.substring(92, 93);
 if (airTemperature != MISSING && quality.matches("[01459]")) {
 context.write(new Text(year), new IntWritable(airTemperature));
 }
 }
}
```

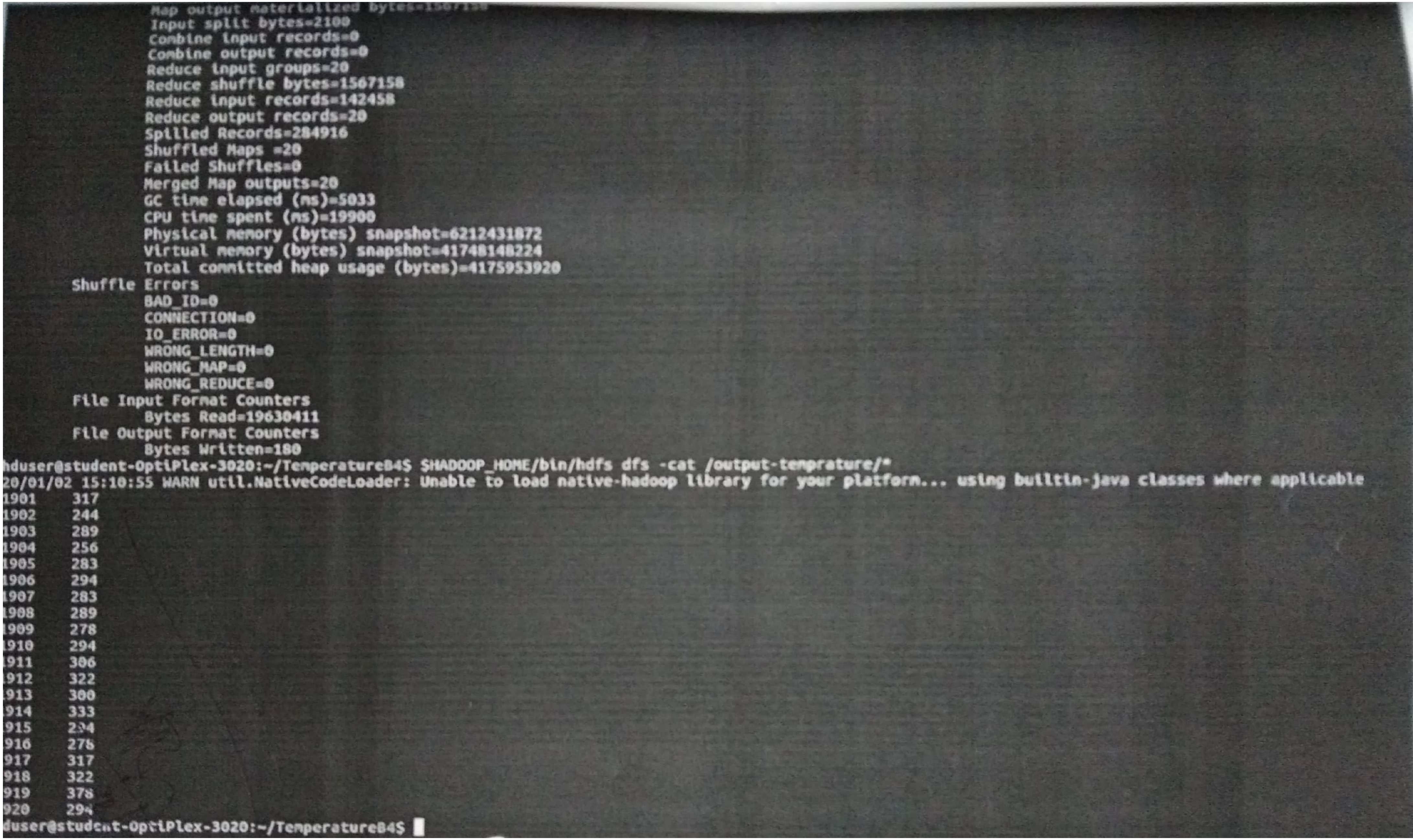
```
MaxTemperatureReducer.java
package MaxMinTemp;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MaxTemperatureReducer
extends Reducer<Text, IntWritable, Text, IntWritable> {
@Override
public void reduce(Text key, Iterable<IntWritable> values,
 Context context)
 throws IOException, InterruptedException {
 int maxValue = Integer.MIN_VALUE;
 for (IntWritable value : values) {
 maxValue = Math.max(maxValue, value.get());
 }
 context.write(key, new IntWritable(maxValue));
 }
 }
```

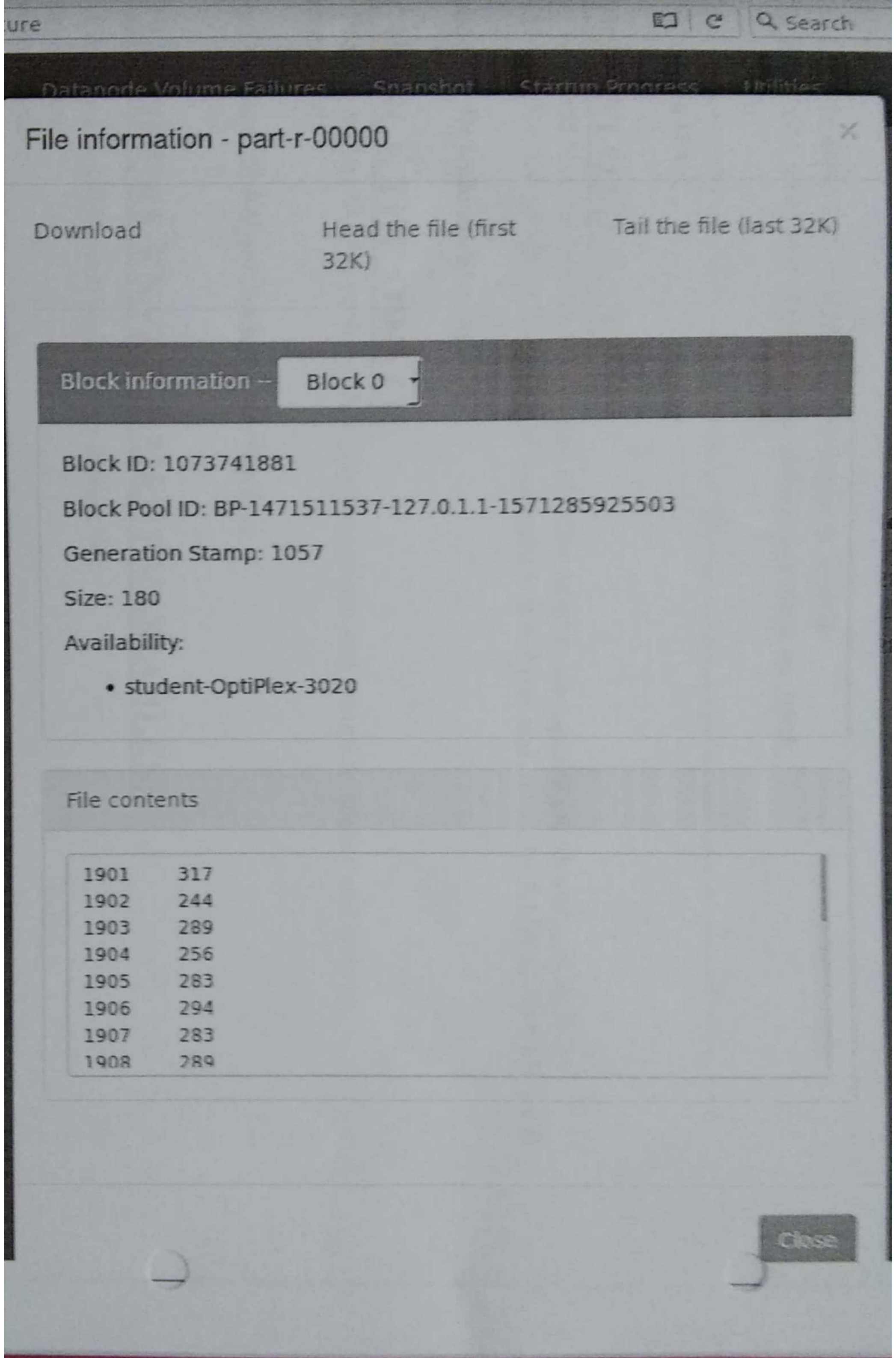
MaxTemperatureDriver.java

```
package MaxMinTemp;

import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
/*This class is responsible for running map reduce job*/
public class MaxTemperatureDriver extends Configured implements Tool{
public int run(String[] args) throws Exception
 {
 if(args.length !=2) {
 System.err.println("Usage: MaxTemperatureDriver <input path> <outputpath>");
 System.exit(-1);
 }
 Job job = new Job();
 job.setJarByClass(MaxTemperatureDriver.class);
 job.setJobName("Max Temperature");
 FileInputFormat.addInputPath(job, new Path(args[0]));
 FileOutputFormat.setOutputPath(job,new Path(args[1]));
 job.setMapperClass(MaxTemperatureMapper.class);
 job.setReducerClass(MaxTemperatureReducer.class);
 job.setOutputKeyClass(Text.class);
 job.setOutputValueClass(IntWritable.class);
 System.exit(job.waitForCompletion(true) ? 0:1);
 boolean success = job.waitForCompletion(true);
 return success ? 0 : 1;
 }
public static void main(String[] args) throws Exception {
 MaxTemperatureDriver driver = new MaxTemperatureDriver();
 int exitCode = ToolRunner.run(driver, args);
 System.exit(exitCode);
 }
}
```



```
                Map output materialized bytes=1567158
                Input split bytes=2100
                Combine input records=0
                Combine output records=0
                Reduce input groups=20
                Reduce shuffle bytes=1567158
                Reduce input records=142458
                Reduce output records=20
                Spilled Records=284916
                Shuffled Maps =20
                Failed Shuffles=0
                Merged Map outputs=20
                GC time elapsed (ms)=5033
                CPU time spent (ms)=19900
                Physical memory (bytes) snapshot=6212431872
                Virtual memory (bytes) snapshot=41748148224
                Total committed heap usage (bytes)=4175953920
        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=19630411
        File Output Format Counters
                Bytes Written=180
hduser@student-OptiPlex-3020:~/TemperatureB4$ $HADOOP_HOME/bin/hdfs dfs -cat /output-temprature/*
20/01/02 15:10:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
1901    317
1902    244
1903    289
1904    256
1905    283
1906    294
1907    283
1908    289
1909    278
1910    294
1911    306
1912    322
1913    300
1914    333
915     294
916     278
917     317
918     322
919     378
920     294
duser@student-OptiPlex-3020:~/TemperatureB4$ █
```



File information - part-r-00000

Block information -- Block 0

Block ID: 1073741881

Block Pool ID: BP-1471511537-127.0.1.1-1571285925503

Generation Stamp: 1057

Size: 180

Availability:

- student-OptiPlex-3020

File contents

```
1901	317
1902	244
1903	289
1904	256
1905	283
1906	294
1907	283
1908	289
```