**DATASET TO BE CONSIDERED**

**Year Month Day Temperature**

**2024 01 01 23**

**2024 01 02 27**

**2024 01 03 25**

**2023 01 01 22**

**2023 01 02 30**

**2023 01 03 21**

**Mapper Class**

The Mapper class will extract the year and temperature from each line.

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;

public class TemperatureMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

private Text year = new Text();

private IntWritable temperature = new IntWritable();

@Override

protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

String line = value.toString();

String[] fields = line.split(",");

// Skip the header line

if (fields[0].equals("Year")) {

return;

}

year.set(fields[0]);

temperature.set(Integer.parseInt(fields[3]));

context.write(year, temperature);

}

}

**Reducer Class**

The Reducer class will find the maximum temperature for each year.

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class TemperatureReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

int maxTemperature = Integer.MIN\_VALUE;

for (IntWritable value : values) {

maxTemperature = Math.max(maxTemperature, value.get());

}

context.write(key, new IntWritable(maxTemperature));

}

}

**Driver Class**

Finally, the Driver class to configure and run the job.

import org.apache.hadoop.conf.Configuration;

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;

public class TemperatureAnalysis {

public static void main(String[] args) throws Exception {

if (args.length != 2) {

System.err.println("Usage: TemperatureAnalysis <input path> <output path>");

System.exit(-1);

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Temperature Analysis");

job.setJarByClass(TemperatureAnalysis.class);

job.setMapperClass(TemperatureMapper.class);

job.setReducerClass(TemperatureReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

**# hadoop jar TemperatureAnalysis.jar TemperatureAnalysis /input/temperature.csv /output/temperature\_analysis**

**# Output**

**The output will be a list of years and their maximum temperatures:**

**2023 30**

**2024 27**