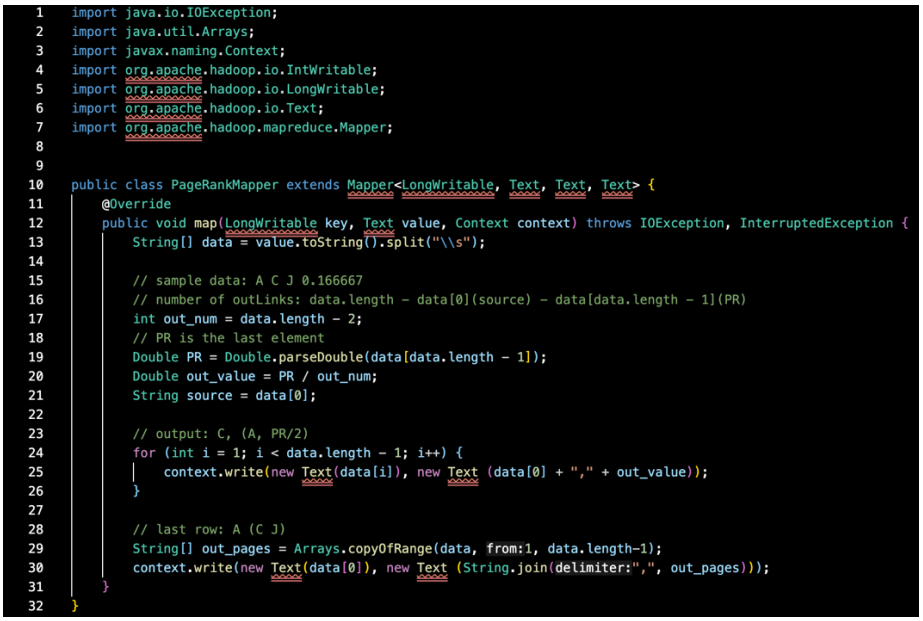
Lab 3 – PageRank Report

Net ID: tw2770

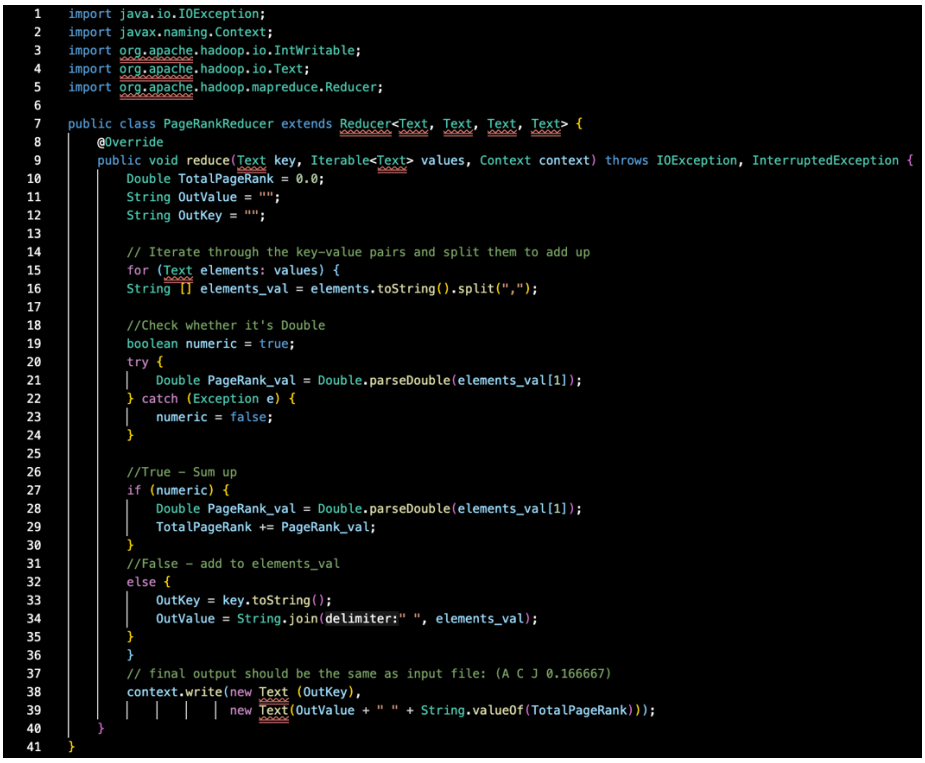
Name: Tzu-An Wang

1. PageRankMapper



```java
import java.io.IOException;
import java.util.Arrays;
import javax.naming.Context;
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 PageRankMapper extends Mapper<LongWritable, Text, Text, Text> {
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String[] data = value.toString().split("\\s");

        // sample data: A C J 0.166667
        // number of outLinks: data.length - data[0](source) - data[data.length - 1](PR)
        int out_num = data.length - 2;
        // PR is the last element
        Double PR = Double.parseDouble(data[data.length - 1]);
        Double out_value = PR / out_num;
        String source = data[0];

        // output: C, (A, PR/2)
        for (int i = 1; i < data.length - 1; i++) {
            context.write(new Text(data[i]), new Text (data[0] + "," + out_value));
        }

        // last row: A (C J)
        String[] out_pages = Arrays.copyOfRange(data, from:1, data.length-1);
        context.write(new Text(data[0]), new Text (String.join(delimiter:",", out_pages)));
    }
}
```

2. PageRankReducer



```java
import java.io.IOException;
import javax.naming.Context;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class PageRankReducer extends Reducer<Text, Text, Text, Text> {
    @Override
    public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
        Double TotalPageRank = 0.0;
        String OutValue = "";
        String OutKey = "";

        // Iterate through the key-value pairs and split them to add up
        for (Text elements: values) {
        String [] elements_val = elements.toString().split(",");

        //Check whether it's Double
        boolean numeric = true;
        try {
            Double PageRank_val = Double.parseDouble(elements_val[1]);
        } catch (Exception e) {
            numeric = false;
        }

        //True - Sum up
        if (numeric) {
            Double PageRank_val = Double.parseDouble(elements_val[1]);
            TotalPageRank += PageRank_val;
        }
        //False - add to elements_val
        else {
            OutKey = key.toString();
            OutValue = String.join(delimiter:" ", elements_val);
        }
        }
        // final output should be the same as input file: (A C J 0.166667)
        context.write(new Text (OutKey),
                      new Text(OutValue + " " + String.valueOf(TotalPageRank)));
    }
}
```

3. PageRank

```java
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.conf.Configuration;

public class PageRank {
    Run | Debug
    public static void main(String[] args) throws Exception {
      for (int i = 0; i < 3; i++) {
        Configuration config = new Configuration();
        config.set("mapred.textoutputformat.separator", " ");

        Job job = Job.getInstance();
        job.setJarByClass(PageRank.class);
        job.setJobName("Page Rank");

        FileInputFormat.addInputPath(job, new Path(args[i]));
        FileOutputFormat.setOutputPath(job, new Path(args[i+1]));
        job.setMapperClass(PageRankMapper.class);
        job.setReducerClass(PageRankReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        job.setNumReduceTasks(1);

        job.waitForCompletion(true);

        if (i == 2) {
          System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
      }
    }
}
```

4. Program runs successfully

```
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls lab3/output0
Found 2 items
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu          0 2024-03-01 21:22 lab3/output0/_SUCCESS
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu        151 2024-03-01 21:22 lab3/output0/part-r-00000
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls lab3/output1
Found 2 items
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu          0 2024-03-01 21:23 lab3/output1/_SUCCESS
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu        161 2024-03-01 21:23 lab3/output1/part-r-00000
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls lab3/output2
Found 2 items
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu          0 2024-03-01 21:23 lab3/output2/_SUCCESS
-rw-r--r--   1 tw2770_nyu_edu tw2770_nyu_edu        160 2024-03-01 21:23 lab3/output2/part-r-00000
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat lab3/output0/part-r-00000
A       C J 0.1166669
B       D E J 0.20000040000000002
C       A B 0.20000040000000002
D       A B C E J 0.05555566666666667
E       J 0.08888906666666667
J       B C 0.3388895666666667
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat lab3/output1/part-r-00000
A       C J 0.11111133333333334
B       D E J 0.2805561166666667
C       A B 0.23888936666666669
D       A B C E J 0.06666680000000001
E       J 0.07777793333333334
J       B C 0.22500045000000002
tw2770_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat lab3/output2/part-r-00000
A       C J 0.13277804333333335
B       D E J 0.24527826833333338
C       A B 0.1813892516666667
D       A B C E J 0.09351870555555557
E       J 0.10685206555555557
J       B C 0.2401856655555556
tw2770_nyu_edu@nyu-dataproc-m:~$
```