Task1:

Code changes:

*****************HadoopPageRank class**********************************

*******************place holder part********************************

resultJob.setMapperClass(HadoopPageRankResultMapper.class);

resultJob.setReducerClass(HadoopPageRankResultReducer.class);

resultJob.setOutputKeyClass(DoubleWritable.class);

resultJob.setOutputValueClass(Text.class);

resultJob.setInputFormatClass(TextInputFormat.class);

*****************HadoopPageRankResultMapper class***********************

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Mapper.Context;

public class HadoopPageRankResultMapper extends Mapper<LongWritable, Text, DoubleWritable, Text>{

    public void map(LongWritable key, Text value, Context context)

        throws IOException, InterruptedException {

      if (value == null || value.getLength() == 0) {

        return;

      }

```java
        int tabIdx1 = value.find("\t");
        int tabIdx2 = value.find("\t", tabIdx1 + 1);
        String link = Text.decode(value.getBytes(), 0, tabIdx1);
        String pageRank = Text.decode(value.getBytes(), tabIdx1 + 1, tabIdx2 - (tabIdx1 + 1));
        double PR = -1*Double.parseDouble(pageRank.toString());
        context.write(new DoubleWritable(PR), new Text(link));
    }
}
```

****************HadoopPageRankResultReducer class*********************
```java
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class HadoopPageRankResultReducer extends Reducer<DoubleWritable, Text, Text, DoubleWritable> {
    public void reduce(DoubleWritable key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
        double temp = 0.0;
        temp = key.get() * -1;
        for (Text value : values) {
            context.write(value, new DoubleWritable(temp));
        }
    }
}
```

Task2:

Code changes:

*****************HadoopPageRankMainJobReducer class************************

```
// I changed the calculation function of receivedContribution
// with the updated flow model calculation
import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class HadoopPageRankMainJobReducer extends Reducer<Text, Text, Text, Text> {
    @Override
    public void reduce(Text key, Iterable<Text> values, Context context)
            throws IOException, InterruptedException {

        if ( values == null ) {
            return;
        }

        String links = "";
        double receivedContribution = 0.0;

        for (Text value : values) {

            String content = value.toString();
            if (content.startsWith("|")) {
                links += content.substring("|".length());
            } else {
```

```
                        receivedContribution += Double.parseDouble(content);
                }

        }
        //the Google solution for spider traps:
        receivedContribution = 0.80*receivedContribution+0.20*0.25;//here is the change
        double newPageRank = receivedContribution;
        context.write(key, new Text(newPageRank + "\t" + links));

    }

}
```

Task3:
3 After iteration 4, b = 1/2