

과제2-3

목표

1. 행렬의 계산을 해보자

간단설명

hadoop을 이용하여 행렬을 덧셈 처리하여 나타낸다

결과

```
Bytes Written=24
hadoop@ubuntu:~/Project$ hdfs dfs -cat matadd_test_out/part-r-00000
0      0      5
1      1      5
hadoop@ubuntu:~/Project$ hdfs dfs -cat matadd_test_out/part-r-00001
0      1      6
1      0      7
```

code

```
package ssafy;

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class MatrixAdd {
    public static class MAddMapper extends Mapper<Object, Text, Text,
IntWritable>{

        public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {

            String[] arr = value.toString().split("\t");
            Text emitkey = new Text (arr[1] + "\t" + arr[2]);
```

```

        IntWritable emitval = new IntWritable (Integer.parseInt (arr[3]));
        context.write (emitkey, emitval);

    }
}

public static class MAddReducer extends Reducer<Text, IntWritable, Text,
IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values, Context
context)
        throws IOException, InterruptedException {

        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write (key, new IntWritable (sum));

    }
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf,
args).getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: <in> <out>");
        System.exit(2);
    }
    FileSystem hdfs = FileSystem.get(conf);
    Path output = new Path(otherArgs[1]);
    if (hdfs.exists(output))
        hdfs.delete(output, true);

    Job job = new Job(conf, "matrix addition");
    job.setJarByClass(MatrixAdd.class);
    job.setMapperClass(MAddMapper.class);
    job.setReducerClass(MAddReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setNumReduceTasks(2);

    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```