# LARGE SCALE DATA PROCESSING (CSE 3025)
## WIN SEMESTER 2017-18

NAME: B.SHUBANKAR

REGNO: 15BCE1123

PROFESSOR: DR. Maheswari N

---

### LAB-10

1.Create a text file with country names and state names. Sort the country names in the ascending order and for every country sort the state names in the descending order using map reduce programming.

**PROGRAM:**

```
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
//import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;


public class sec_sort {
```

```
public static class CompositeKeyWritable implements
Writable,WritableComparable<CompositeKeyWritable> {

private String Country;

private String state;

public CompositeKeyWritable() {

}

public CompositeKeyWritable(String Country, String state) {

this.Country = Country;

this.state = state;

}

public String toString() {

return (new StringBuilder().append(Country).append("\t").append(state)).toString();

}

public void readFields(DataInput dataInput) throws

IOException {

Country = WritableUtils.readString(dataInput);

state = WritableUtils.readString(dataInput);

}

public void write(DataOutput dataOutput) throws

IOException {

WritableUtils.writeString(dataOutput,Country);

WritableUtils.writeString(dataOutput,state);

}

public int compareTo(CompositeKeyWritable objKeyPair) {

int result =
```

```
Country.compareTo(objKeyPair.Country);

if (0 == result) {

result = state.compareTo(objKeyPair.state);

}

return result;

}

}

public static class mapper1 extends Mapper<LongWritable, Text,
CompositeKeyWritable, NullWritable> {

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

if (value.toString().length() > 0) {

String arrstateAttributes[] = value.toString().split("\t");

context.write(new
CompositeKeyWritable(arrstateAttributes[0].toString(),(arrstateAttributes[1].toString
())),

NullWritable.get());

}

}

}

public static class SecondarySortBasicPartitioner extends
Partitioner<CompositeKeyWritable, NullWritable> {

public int getPartition(CompositeKeyWritable key, NullWritable value,int
numReduceTasks) {

return (key.Country.hashCode() % numReduceTasks);

}
```

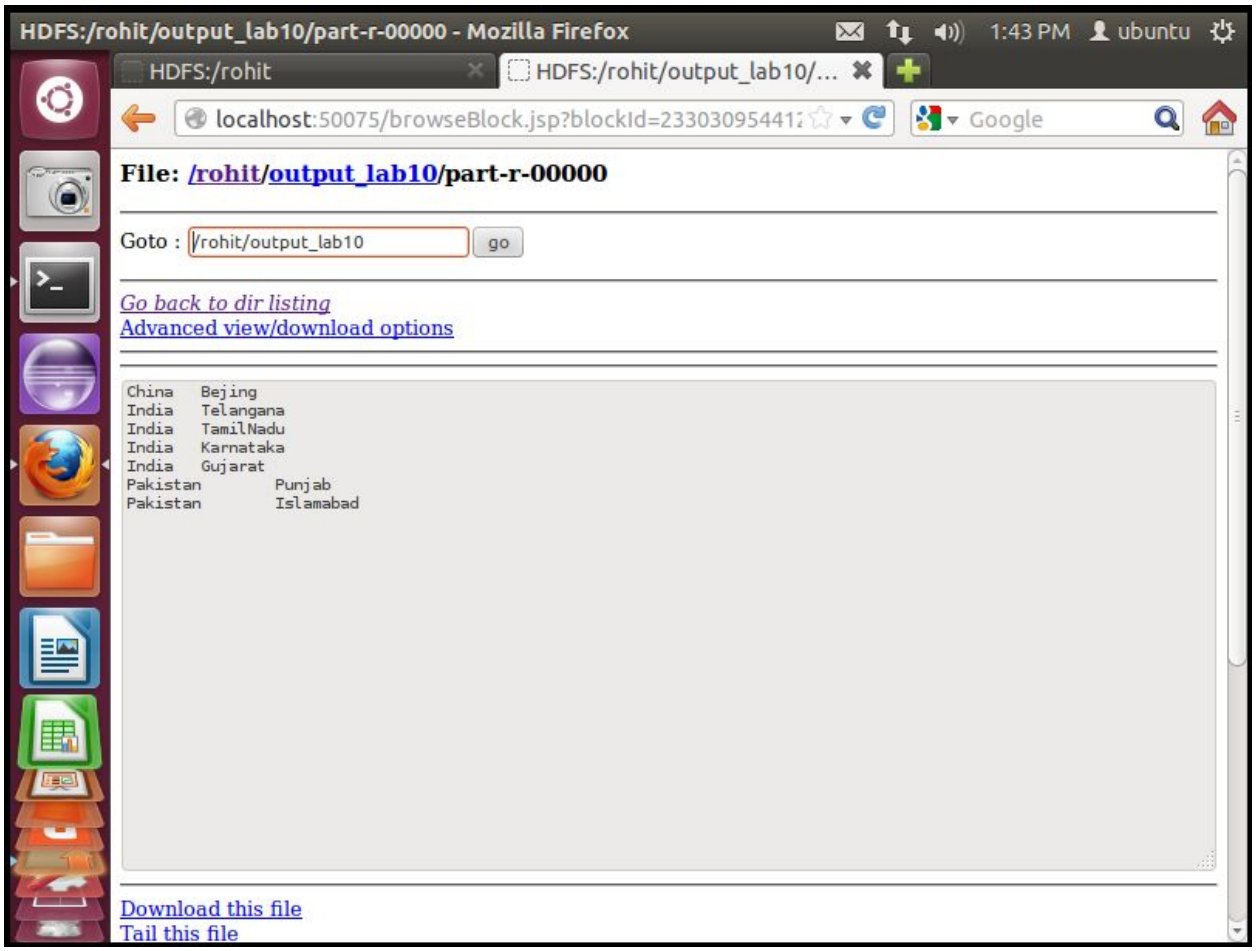
```
}
public static class SecondarySortBasicCompKeySortComparator extends
WritableComparator {
protected SecondarySortBasicCompKeySortComparator()
{
super(CompositeKeyWritable.class, true);
}
@SuppressWarnings("rawtypes")
public int compare(WritableComparable w1,WritableComparable w2) {
CompositeKeyWritable key1 = (CompositeKeyWritable) w1;
CompositeKeyWritable key2 = (CompositeKeyWritable) w2;
int cmpResult = key1.Country.compareTo(key2.Country);
if (cmpResult == 0)
{
return -key1.state.compareTo(key2.state);
}
return cmpResult;
}
}
public static class SecondarySortBasicGroupingComparator
extends WritableComparator {
protected SecondarySortBasicGroupingComparator() {
super(CompositeKeyWritable.class, true);
}
@SuppressWarnings("rawtypes")
```

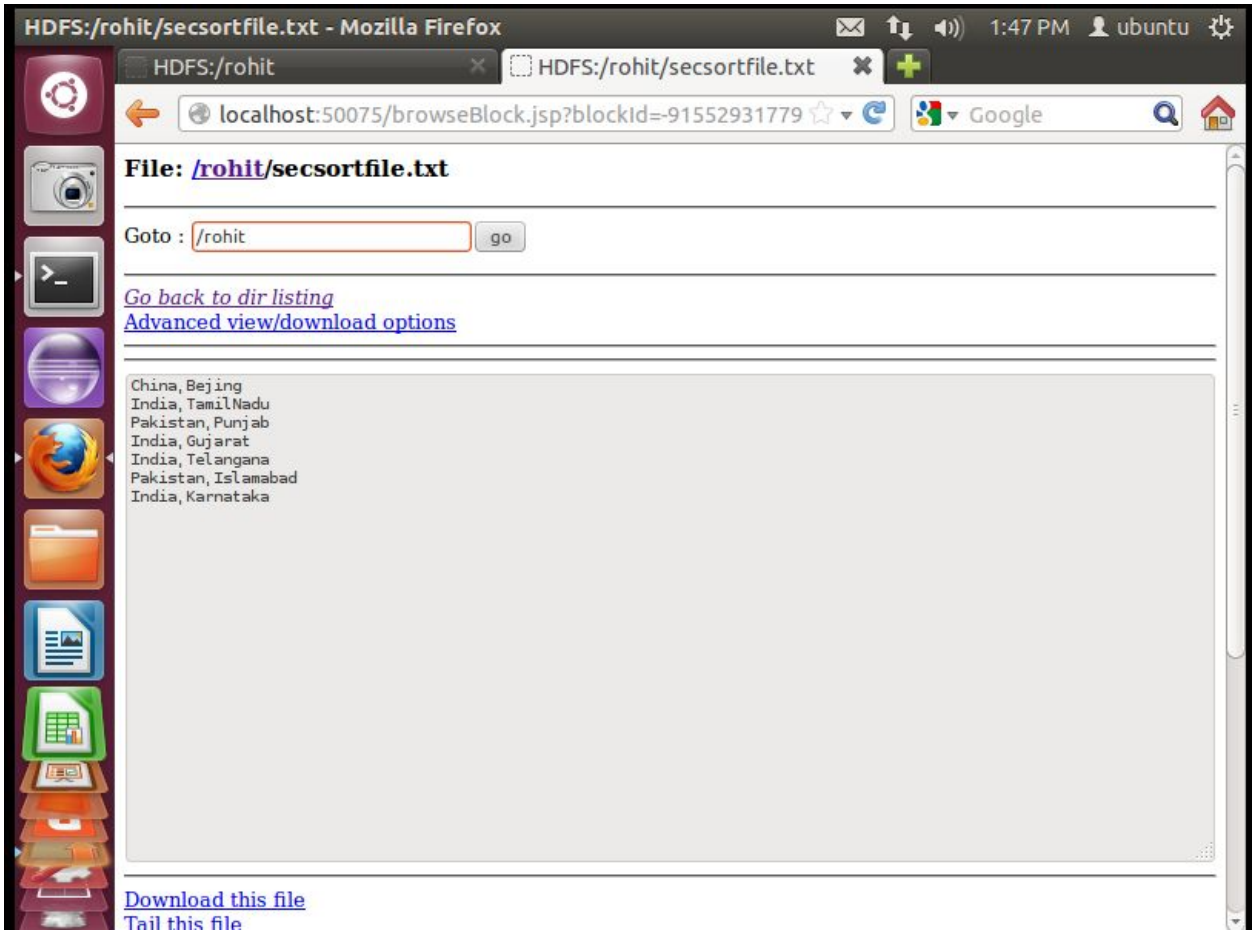
```
public int compare(WritableComparable w1,WritableComparable w2) {
CompositeKeyWritable key1 = (CompositeKeyWritable) w1;
CompositeKeyWritable key2 = (CompositeKeyWritable) w2;
return key1.Country.compareTo(key2.Country);
}
}
public static class SecondarySortBasicReducer extends
Reducer<CompositeKeyWritable, NullWritable,
CompositeKeyWritable, NullWritable> {
public void reduce(CompositeKeyWritable key, Iterable<NullWritable> values,
Context context) throws
IOException, InterruptedException {
for(NullWritable value : values)
{
        context.write(key,NullWritable.get());
}
}
}
public static void main(String[] args) throws Exception { //Driver
        Configuration conf=new Configuration();
   Job job=new Job(conf,"Secondary sort");
   job.setJarByClass(sec_sort.class);
   job.setOutputKeyClass(Text.class);
   job.setOutputValueClass(Text.class);
   job.setMapperClass(mapper1.class);
```

```
    job.setMapOutputKeyClass(CompositeKeyWritable.class);
    job.setMapOutputValueClass(NullWritable.class);
    job.setPartitionerClass(SecondarySortBasicPartitioner.class);
    job.setSortComparatorClass(SecondarySortBasicCompKeySortComparator.class);

job.setGroupingComparatorClass(SecondarySortBasicGroupingComparator.class);
    job.setReducerClass(SecondarySortBasicReducer.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.waitForCompletion(true);
}
}
```

**OUTPUT:**


HDFS:/rohit/output_lab10/part-r-00000 - Mozilla Firefox
1:43 PM
ubuntu
HDFS:/rohit
HDFS:/rohit/output_lab10/...
localhost:50075/browseBlock.jsp?blockId=23303095441
Google
File: /rohit/output_lab10/part-r-00000
Goto : /rohit/output_lab10
go
Go back to dir listing
Advanced view/download options
China Bejing
India Telangana
India TamilNadu
India Karnataka
India Gujarat
Pakistan Punjab
Pakistan Islamabad
Download this file
Tail this file


HDFS:/rohit/secsortfile.txt - Mozilla Firefox
1:47 PM
ubuntu
HDFS:/rohit
HDFS:/rohit/secsortfile.txt
localhost:50075/browseBlock.jsp?blockId=-91552931779
Google
File: /rohit/secsortfile.txt
Goto : /rohit
go
Go back to dir listing
Advanced view/download options
China, Bejing
India, TamilNadu
Pakistan, Punjab
India, Gujarat
India, Telangana
Pakistan, Islamabad
India, Karnataka
Download this file
Tail this file