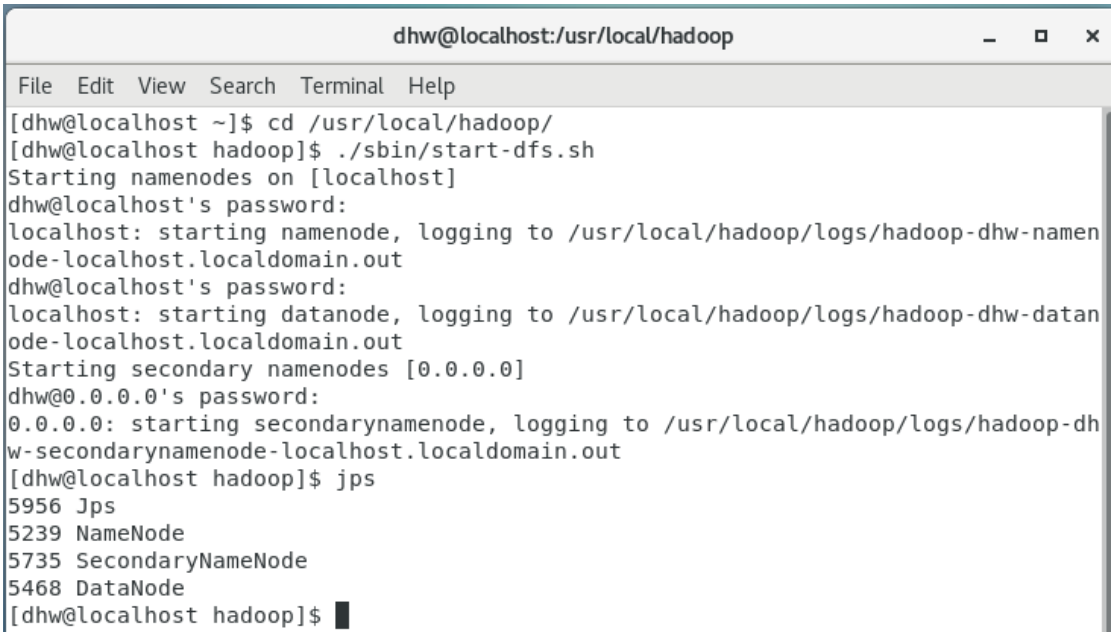
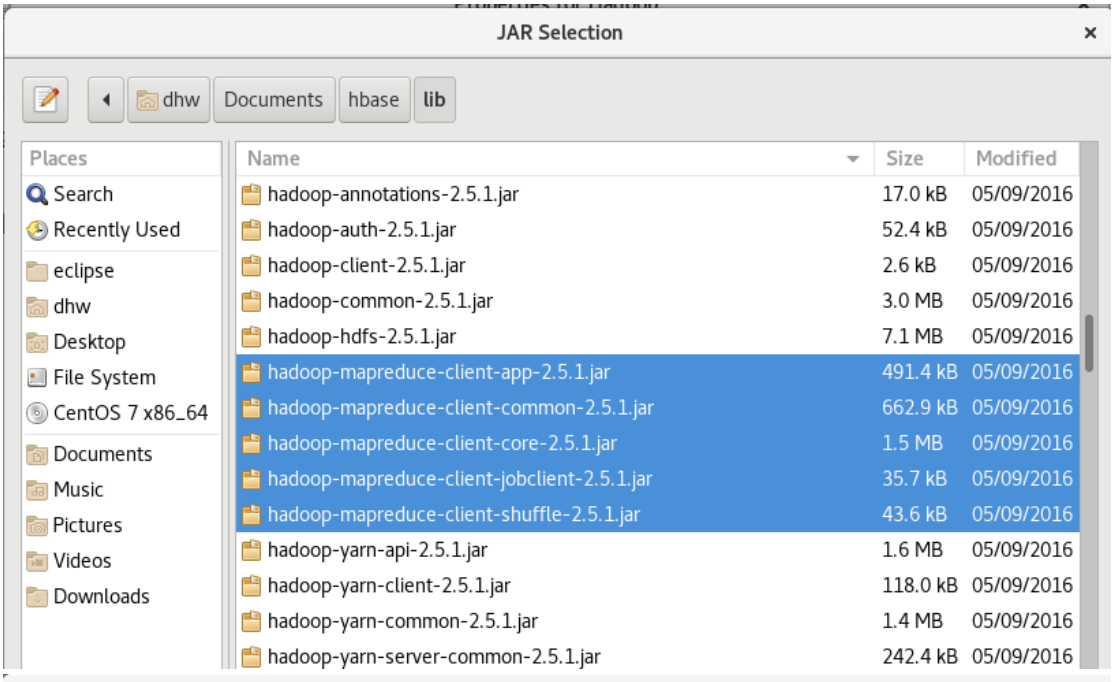
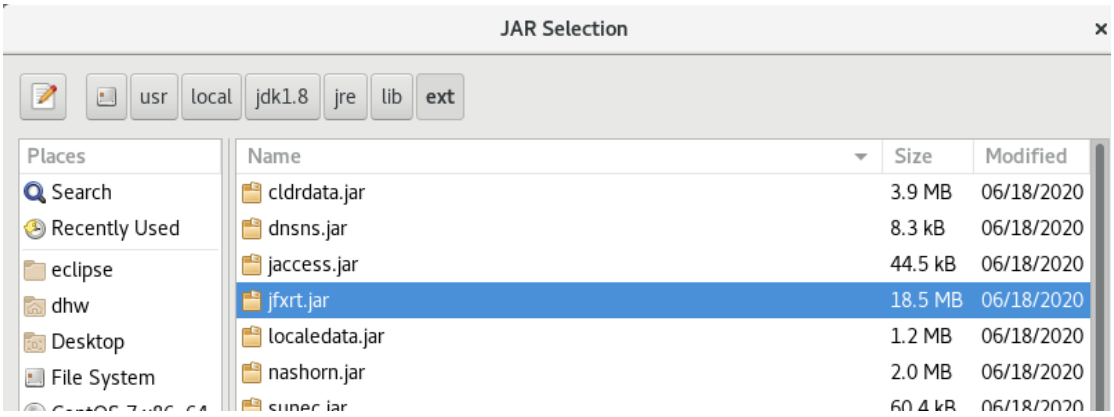
启动 Hadoop：



导入 jar 包：



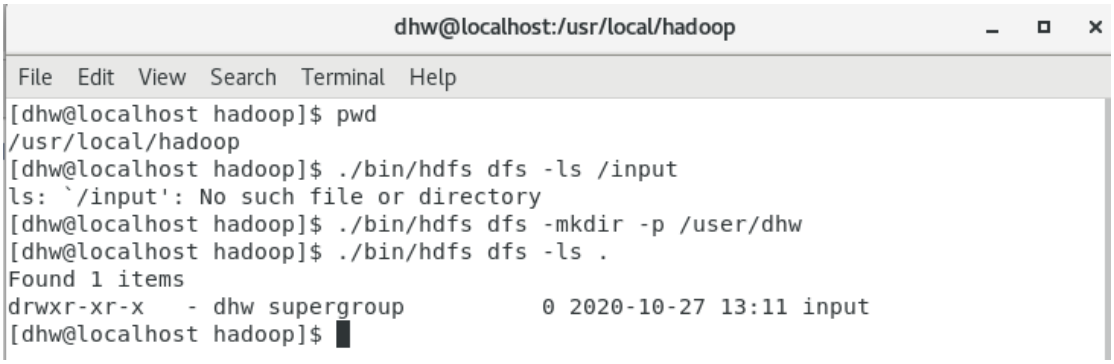


对于其他 jar 包，鼠标放上去按 ctrl 键点击左键，即可提示安装 jar 名称

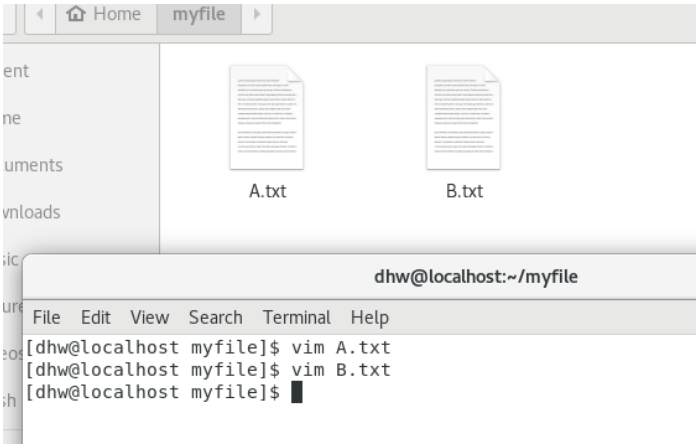
实验

（一） 编程实现文件合并和去重操作

首先创建 input 目录



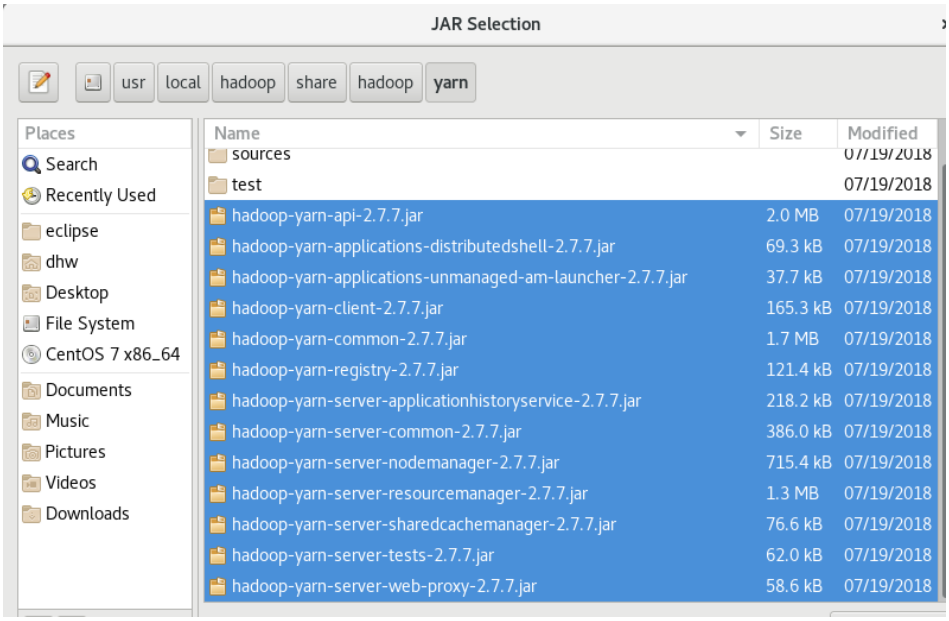
然后写入指定内容并上传文件



```
dhw@localhost:/usr/local/hadoop
File  Edit  View  Search  Terminal  Help
[dhw@localhost hadoop]$ ./bin/hdfs dfs -put /home/dhw/myfile/A.txt input
[dhw@localhost hadoop]$ ./bin/hdfs dfs -put /home/dhw/myfile/B.txt input
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls input
Found 3 items
-rw-r--r--   1 dhw supergroup         66 2020-11-21 17:44 input/A.txt
-rw-r--r--   1 dhw supergroup         55 2020-11-21 17:45 input/B.txt
-rw-r--r--   3 dhw supergroup         49 2020-10-31 14:27 input/t1.txt
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat input/A.txt
20180101 x
20180102 y
20180103 x
20180104 y
20180105 z
20180106 x
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat input/B.txt
20180101 y
20180102 y
20180103 x
20180104 z
20180105 y
[dhw@localhost hadoop]$
```

（删除上述 t1.txt 文件后在操作，避免干扰）报错添加 jar 包:

```
Console
<terminated> t6_1 [Java Application] /usr/local/jdk1.8/bin/java (Nov 21, 2020 6:55:23 PM)
found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j system properly.
che.org/log4j/1.2/faq.html#noconfig for more info.
ang.NoClassDefFoundError: org/apache/hadoop/yarn/util/Apps
defineClass1(Native Method)
defineClass(ClassLoader.java:756)
assLoader.defineClass(SecureClassLoader.java:142)
r.defineClass(URLClassLoader.java:468)
r.access$100(URLClassLoader.java:74)
r$1.run(URLClassLoader.java:369)
r$1.run(URLClassLoader.java:363)
troller.doPrivileged(Native Method)
r.findClass(URLClassLoader.java:362)
loadClass(ClassLoader.java:418)
lassLoader.loadClass(Launcher.java:355)
loadClass(ClassLoader.java:351)
ed.LocalDistributedCacheManager.setup(LocalDistributedCacheManager.java:93)
```



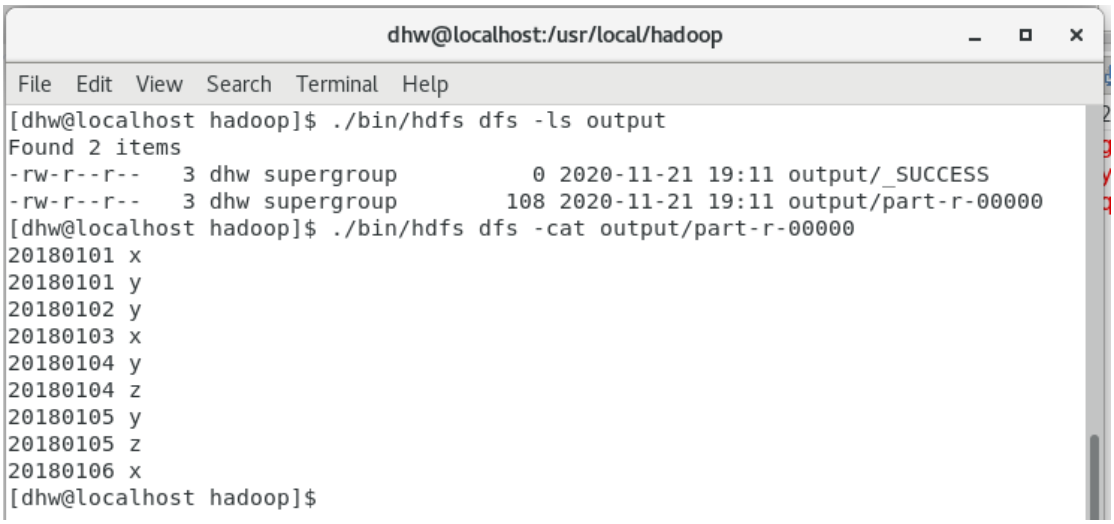
t6_1.java

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.Merger;
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;


public class t6_1 {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("fs.defaultFS", "hdfs://localhost:9000");
        conf.set("fs.hdfs.impl", "org.apache.hadoop.hdfs" +
                ".DistributedFileSystem");

        String[] path = new String[] {"input", "output"};

        Job job = Job.getInstance(conf, "Merge and duplicate removal");
        job.setJarByClass(Merger.class);  // name
        job.setMapperClass(MyMap.class);  // Map class
        job.setReducerClass(MyReduce.class);  // add Reduce class
        job.setOutputKeyClass(Text.class);  // set output type
        job.setOutputValueClass(Text.class);  // set input type

        FileInputFormat.addInputPath(job, new Path(path[0]));  // input file
        FileOutputFormat.setOutputPath(job, new Path(path[1]));  // output file

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

```
public static class MyMap extends Mapper<Object, Text, Text, Text> {
    private static Text text = new Text();
    public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
        this.text = value;
        context.write(text, new Text(""));
    }
}

public static class MyReduce extends Reducer<Text, Text, Text, Text>{
    public void reduce(Text key, Iterable<Text> values, Context context)
            throws IOException, InterruptedException {
        context.write(key, new Text(""));
    }
}
```

Console

```
<terminated> t6_1 [Java Application] /usr/local/jdk1.8/bin/java (Nov 21, 2020 7:11:12 PM)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metric
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for mo
```



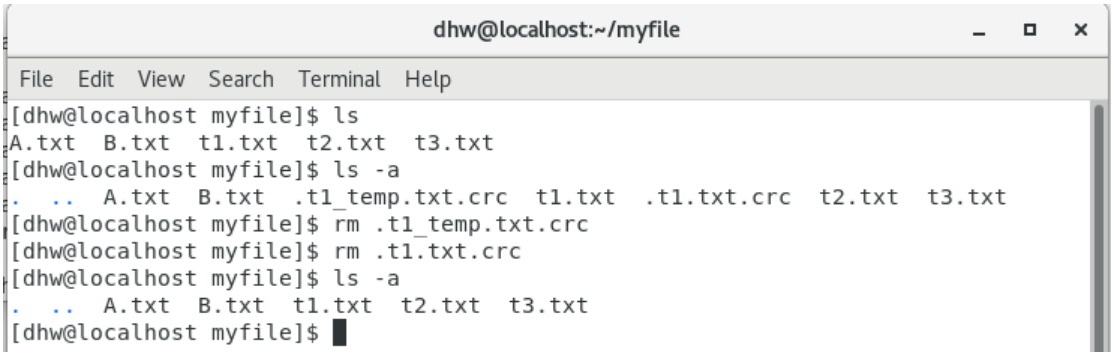
（二） 实现对输入文件的排序

创建目录 input2 用来存 3 个文件，同上创建文件

```
[dhw@localhost hadoop]$ ./bin/hdfs dfs -mkdir -p /usr/dhw
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls .
Found 2 items
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:10 input
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:11 output
[dhw@localhost hadoop]$ ./bin/hdfs dfs -mkdir -p /usr/dhw input2
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls .
Found 3 items
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:10 input
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:23 input2
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:11 output
[dhw@localhost hadoop]$
```

若出现如下错误则:

```
01)
        at org.apache.hadoop.fs.shell.Command.run(Command.java:165)
        at org.apache.hadoop.fs.FsShell.run(FsShell.java:287)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:84)
        at org.apache.hadoop.fs.FsShell.main(FsShell.java:340)
put: Checksum error: file:/home/dhw/myfile/t1.txt at 0 exp: -1197748375 got: -44
4126375
[dhw@localhost hadoop]$
```

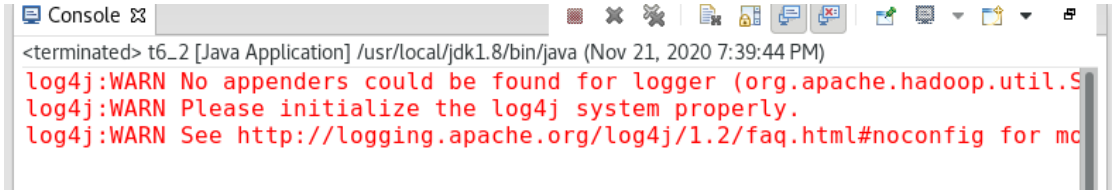


然后重新上传

```
[dhw@localhost hadoop]$ ./bin/hdfs dfs -put /home/dhw/myfile/t1.txt input2
[dhw@localhost hadoop]$ ./bin/hdfs dfs -put /home/dhw/myfile/t2.txt input2
[dhw@localhost hadoop]$ ./bin/hdfs dfs -put /home/dhw/myfile/t3.txt input2
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls input2/t1.txt
-rw-r--r--   1 dhw supergroup         12 2020-11-21 19:30 input2/t1.txt
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat input2/t1.txt
33
37
12
40
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat input2/t2.txt
4
16
39
5
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat input2/t3.txt
1
45
25
[dhw@localhost hadoop]$
```

t6_2.java

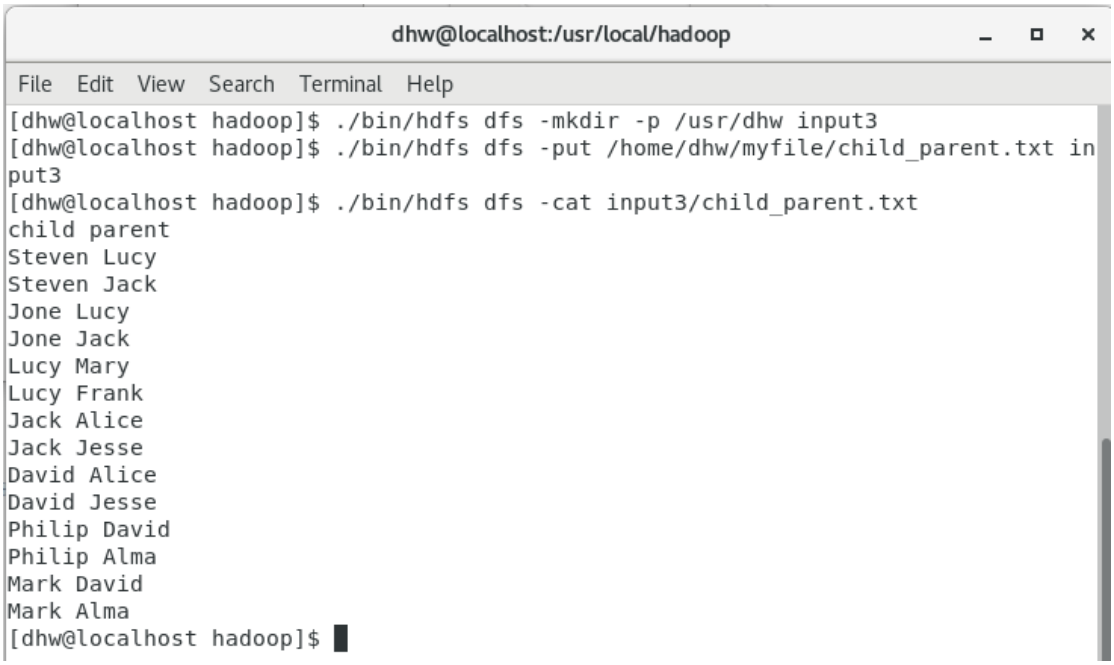
```java
public class t6_2 {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("fs.defaultFS", "hdfs://localhost:9000");
        conf.set("fs.hdfs.impl", "org.apache.hadoop.hdfs" +
                ".DistributedFileSystem");

        String[] path = new String[] {"input2", "output2"};
        String[] other = new GenericOptionsParser(conf, path).getRemainingArgs();

        Job job = Job.getInstance(conf, "mergesort");
        job.setJarByClass(MergeSort.class);  // name
        job.setMapperClass(MyMap.class);  // Map class
        job.setReducerClass(MyReduce.class);  // add Reduce class
        job.setOutputKeyClass(IntWritable.class);  // set output type
        job.setOutputValueClass(IntWritable.class);  // set input type

        FileInputFormat.addInputPath(job, new Path(other[0]));  // input file
        FileOutputFormat.setOutputPath(job, new Path(other[1]));  // output file

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

    public static class MyMap extends Mapper<Object, Text, IntWritable, IntWritable> {
        private static IntWritable data = new IntWritable();

        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            String line = value.toString();
            this.data.set(Integer.parseInt(line));
            context.write(data, new IntWritable(1));
        }
    }

    public static class MyReduce extends Reducer<IntWritable, IntWritable,
                                                  IntWritable, IntWritable>{
        private static IntWritable linenum = new IntWritable(1);

        public void reduce(IntWritable key, Iterable<IntWritable> values, Context context)
                throws IOException, InterruptedException {
            for (IntWritable num : values) {
                context.write(linenum, key);
                linenum = new IntWritable(linenum.get() + 1);
            }
        }
    }
```




```
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls .
Found 4 items
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:26 input
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:31 input2
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:11 output
drwxr-xr-x   - dhw supergroup          0 2020-11-21 19:40 output2
[dhw@localhost hadoop]$ ./bin/hdfs dfs -ls output2
Found 2 items
-rw-r--r--   3 dhw supergroup          0 2020-11-21 19:40 output2/_SUCCESS
-rw-r--r--   3 dhw supergroup         54 2020-11-21 19:40 output2/part-r-00000
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat output2/part-r-00000
1       1
2       4
3       5
4       12
5       16
6       25
7       33
8       37
9       39
10      40
11      45
[dhw@localhost hadoop]$
```

（三） 对给定的表格进行信息挖掘

查看准备文件



t6_3.java

```java
public class t6_3 {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("fs.defaultFS", "hdfs://localhost:9000");
        conf.set("fs.hdfs.impl", "org.apache.hadoop.hdfs" +
                ".DistributedFileSystem");

        String[] path = new String[] {"input3", "output3"};

        Job job = Job.getInstance(conf, "Single table join");
        job.setJarByClass(t6_3.class);  // name
        job.setMapperClass(MyMap.class);  // Map class
        job.setReducerClass(MyReduce.class);  // add Reduce class
        job.setOutputKeyClass(Text.class);  // set output type
        job.setOutputValueClass(Text.class);  // set input type

        FileInputFormat.addInputPath(job, new Path(path[0]));  // input file
        FileOutputFormat.setOutputPath(job, new Path(path[1]));  // output file

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static int time = 0;

    public static class MyMap extends Mapper<Object, Text, Text, Text> {
        @Override
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {

            String line = value.toString();
            String[] childAndParent = line.split(" ");
            List<String> list = new ArrayList<String>(2);

            for (String childOrParent : childAndParent) {
                if (! "".equals(childOrParent)) {
                    list.add(childOrParent);
                }

            }

            if (!"child".equals(list.get(0))) {
                String childName = list.get(0);
                String parentName = list.get(1);

                String relationType = "1";
                context.write(new Text(parentName), new Text(relationType + "+"
                        + childName + "+" + parentName));

                relationType = "2";
                context.write(new Text(childName), new Text(relationType + "+"
                        + childName + "+" + parentName));
            }
        }
```

```
    public static class MyReduce extends Reducer<Text, Text, Text, Text> {

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

            if (time == 0) {
                context.write(new Text("grand_child"), new Text("grand_parent"));
                time ++;

            }

            List<String> grandChild = new ArrayList<String>();
            List<String> grandParent = new ArrayList<String>();

            for (Text text : values) {
                String s = text.toString();
                String[] relation = s.split("\\+");
                String relationType = relation[0];
                String childName = relation[1];
                String parentName = relation[2];

                if ("1".equals(relationType)) {
                    grandChild.add(childName);
                } else {
                    grandParent.add(parentName);
                }

            }

            int grandParentNum = grandParent.size();
```

```
            int grandParentNum = grandParent.size();
            int grandChildNum = grandChild.size();

            if (grandParentNum != 0 && grandChildNum != 0) {
                for (int m = 0; m < grandChildNum; m++) {
                    for (int n = 0; n < grandParentNum; n++) {
                        context.write(new Text(grandChild.get(m)),
                                new Text(grandParent.get(n)));
                    }
                }
            }
        }
    }
}
```

实验结果:

```
[dhw@localhost hadoop]$ ./bin/hdfs dfs -cat output3/part-r-00000
grand_child     grand_parent
Mark    Jesse
Mark    Alice
Philip  Jesse
Philip  Alice
Jone    Jesse
Jone    Alice
Steven  Jesse
Steven  Alice
Steven  Frank
Steven  Mary
Jone    Frank
Jone    Mary
[dhw@localhost hadoop]$
```