APACHE MAVEN

Download newest version from http://www.apache.org/dyn/closer.cgi/maven/binaries/

```
wget http://apache.lehtivihrea.org//maven/binaries/apache-maven-3.0.3-
bin.tar.gz
```

Unpack

```
tar -xvf apache-maven-3.0.3-bin.tar.gz
```

APACHE HADOOP

Download version 0.20.203 from http://www.apache.org/dyn/closer.cgi/hadoop/common/

```
wget http://apache.linux-mirror.org//hadoop/common/hadoop-0.20.203.0/hadoop-
0.20.203.0rc1.tar.gz
```

Unpack

```
tar -xvf hadoop-0.20.203.0rc1.tar.gz
```

CONFIGURATION

CONF/CORE-SITE.XML

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

CONF/HDFS-SITE.XML

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

CONF/MAPRED-SITE.XML

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
  <property>
    <name>mapred.tasktracker.map.tasks.maximum</name>
    <value>4</value>
  </property>
  <property>
    <name>mapred.map.tasks</name>
    <value>4</value>
  </property>
</configuration>
```

SSH

Check ssh to localhost without pass

```
ssh localhost
```

On error, create ssh key

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

CLEANUP

Change <username> with your linux username

```
rm -rf /tmp/hadoop-<username>
bin/hadoop namenode -format
```

START HADOOP

```
bin/start-all.sh
```

APACHE GIRAPH

Checkout by SVN

```
svn co http://svn.apache.org/repos/asf/incubator/giraph/trunk giraph
```

First check with Giraph Tests: replace <maven3-path> with your local path to the downloaded maven3 package

```
cd giraph
<maven3-path>/bin/mvn clean test -Dprop.mapred.job.tracker=localhost:9001
```

Open http://localhost:50030 in your browser to see the jobs

GIRAPTH EXAMPLE: BREADTH-FIRST SEARCH

IMPORT GIRAPH PROJECT IN ECLIPSE

- File | Import
- Maven |Existing Maven Projects
- Root Directory | Browse
- Choose <giraph-path>
- Finish

IMPLEMENT THE VERTEX LOGIC

- Create new package 'eu.robust.giraph' in 'src/main/java'
- Create new class 'BreadthFirstSearchVertex' extending EdgeListVertex

```
public class BreadthFirstSearchVertex extends EdgeListVertex<LongWritable,
        LongWritable, NullWritable, LongWritable> {

}
```

- Implement the required 'compute'method

Pseudo-Code for 'src/main/java/eu/robust/giraph/BreadthFirstSearchVertex.java'

```
if(first superStep) {
        vertexValue = INFINITY;
}
distance = is source node ? 0 : INFINITY;
for(message : messages) distance = min(distance, message);
if(distance < vertexValue) {
        vertexValue = distance;
        sendToAncestors(distance + 1);
}
```

- Create an Unit-test: 'src/test/java/eu/robust/giraph/BreadthFirstSearchVertexTest.java'

IMPLEMENT THE I/O ADAPTERS

- Schema: '<SourceVertex>\t<DestinationVertex1>\t<DestinationVertex2> … \n'
- Create an InputFormat: 'src/main/java/eu/robust/giraph/LongLongNullLongTextInputFormat.java'
- Create an OutputFormat: 'src/main/java/eu/robust/giraph/GenericVertexWithValueTextOutputFormat.java'
- Don't forget the tests ☺

COMPILE THE PACKAGE

```
<maven3-path>/bin/mvn clean package
```

START THE BREADTH-FIRST SEARCH JOB

```
cd <hadoop-path>
bin/hadoop jar <giraph-path>/target/giraph-0.2-SNAPSHOT-jar-with-
dependencies.jar org.apache.giraph.GiraphRunner
eu.robust.giraph.BreadthFirstSearchVertex -if
eu.robust.giraph.LongLongNullLongTextInputFormat -ip file://<input-path> -
of eu.robust.giraph.GenericVertexWithValueTextOutputFormat -op
file://<output-path> -w 1 -ca bfs.sourceId=<source-node>
```

- <input-path>: absolute input path, e.g. ‘/home/user/sap_input.tsv
- <output-path>: absolute output path, e.g. ‘/home/user/sap_output.tsv
- <source-node>: e.g. 438