**Partition Algorithm**

**Environment :**

Hadoop 2.7, Single node cluster, Ubuntu 16.04

Steps to Execute:

**Part A**

Order file generation:

1. Compile using the below command

bin/hadoop com.sun.tools.javac.Main NodeOrdering.java

1. Jar the class files

jar -cvf NodeOrdering.jar NodeOrdering\*.class

1. Create input directory in distributed file system

bin/hadoop dfs -mkdir /input

1. Copy the input file to the input directory

bin/hadoop dfs -copyFromLocal input.dat /input/

1. Execute the code using the below command

bin/Hadoop jar NodeOrdering.jar NodeOrdering /input/input.dat <out1> <out2>

out1,out2 – specify the 2 output paths and the final output will be displayed in out2.

1. Merge the out2 folder content to create the order.dat file using the below command. This command will group the all the output file into single file and stores it in the local directory.

bin/hadoop dfs -getmerge /<out2>/ order.dat

**Part B**

**Vertices file generation:**

1) Run the script the vertices.sh provided with the other files with the input file as argument.

./vertices.sh input.dat

2) Once the script ran vertices.dat will be created.

**Part C**

Partition Algorithm Execution:

1. Compile using the below command

bin/hadoop com.sun.tools.javac.Main PartitionCountTriangle.java

1. Jar the class files

jar -cvf PartitionCountTriangle.jar PartitionCountTriangle\*.class

1. Create input and order directory in distributed file system

bin/hadoop dfs -mkdir /input /order/ /vertices

1. Copy the input file to the input directory

bin/hadoop dfs -copyFromLocal input.dat /input/

bin/hadoop dfs -copyFromLocal order.dat /order/

bin/hadoop dfs -copyFromLocal vertices.dat /vertices/

note: order.dat is the file generated in Part A

vertices.dat s the file generated in Part B

1. Execute the code using the below command

bin/Hadoop jar PartitionCountTriangle.jar PartitionCountTriangle /input/ <out1> /order/order.dat input/input.dat /vertices/vertices.dat <out2> <out3>

out1,out2,out3 – specify the 3 output paths and the final output will be displayed in out3.

Each line in the output file will have the vertex Id and number of triangles among its neighbors.