README

a) Steps to run the program

i) Uncompress the zip file.
ii) Create seven folders for each glados, kansas, kinks, newyork, reddwarf, jopline, doors which represent multiple servers of the network.
iii) Store the Colors.java file in all the folders.
iv) Store the main bootstrap.java file along with the files to be processed in the newyork folder.
v) Store the Process_Mapreduce.java file in the remaining folder. These servers in future will be behaving either as mapper or reducer for processing the file.
vi) Compile and run all the files.
vii) View the console and follow the instructions to view the difference in time consumed while processing the file on single server and while processing it using map-reduce framework.

b) Original Contribution:

Nisha's Contribution:
Responsible for implementation of the sorting algorithm required for our map-reduce algorithm
Responsible for designing the high-level structure of the map-reduce algorithm.
Implemented the program of bootstrap server
Jinesh's Contribution:
Created test files of different sizes for map-reduce implementation
Implemented the program for mapper and reducer servers.
Designing the User interface.

The progress report was prepared by both of us.

c) Summary:

We designed our own map-reduce algorithm. Our algorithm based on the file size decides the number of mappers to be used to complete the job. After every job completion we keep a track of slow mapper and ensure that it will be used as reducer for next job and not as a mapper. We provide the statistics of the time taken to complete the job in the single machine and a set of machines using map-reduce algorithm.