**Homework - 5**

**Contents of the folder** -
1) Input (ratings.csv)
2) Output (SaiSree_Kamineni_communities.txt, SaiSree_Kamineni_betweenness.txt)
3) Solution – Scala code (SaiSree_Kamineni_community.scala), Jar files (SaiSree_Kamineni_community.jar)
4) Description File (SaiSree_Kamineni_description.pdf)

Note - I haven't set up SPARK_HOME variable. So I run the scripts and commands from inside "*spark-1.6.3-bin-hadoop2.4*" directory.
Place this UnZipped folder in *spark-1.6.3-bin-hadoop2.4* directory

**Steps to run the jar file** -
*./bin/spark-submit --driver-memory 8g --class "HW5" --master local[4] ./SaiSree_Kamineni_hw5/Solution/SaiSree_Kamineni_community.jar "./SaiSree_Kamineni_hw5/Input/ratings.csv" "./SaiSree_Kamineni_communities.txt" "./SaiSree_Kamineni_betweenness.txt"*

Output will be stored in current directory that is spark-1.6.3-bin-hadoop2.4 with name SaiSree_Kamineni_betweenness.txt/part-00000, SaiSree_Kamineni_communities.txt

**Output format**
Betweenness Sample –
(1,4,6.4)
(1,7,5.9)
(1,15,7.4)
(1,19,6.8)
(1,21,6.4)
(1,22,6.8)
(1,23,7.3)
(1,30,7.3)
(1,34,6.4)
(1,35,1.7)
(1,41,6.2)
(1,48,6.8)
(1,49,5.1)
(1,56,7.1)
(1,57,6.4)
(1,73,7.4)

Communitites Sample –

[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,
35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65
…

**Approach** –

1) Read the ratings file and identified the pairs of users who have atleast 3 movies rated commonly.
2) Created a graph structure and identified the neighbors for each user.
3) Wrote a bfs_between function that does bfs and betweenness for each vertex and returns the list of edges along with betweenness. This function is called over all the vertices using flatmap.
4) These edges are sorted in descending order of betweenness and top edges are continuously removed and modularity is calculated. This happens in a while loop and the loop breaks when modularity is smaller than the previous value.
5) Used graphx connected components method to find the communities in the graph.