Spark

Download the dataset from (soc-LiveJournal1Adj.txt)

Part A

Q: Finding mutual friends

Assume the friends are stored as Person-> [List of Friends], our friends list is then:A

-> B C D
B -> A C D E
C -> A B D E
D -> A B C E
E -> B C D

Each line will be an argument to a mapper. For every friend in the list of friends, the mapper will output a key-value pair. The key will be a friend along with the person. The value will be the list of friends. The key will be sorted so that the friends are in order, causing all pairs of friends to go to the same reducer. This is hard to explain with text, so let's just do it and see if you can see the pattern. After all the mappers are done running, you'll have a list like this:

For map (A -> B C D):

(A B) -> B C D
(A C) -> B C D
(A D) -> B C D

For map (B -> A C D E): (Note that A comes before B in the key)

(A B) -> A C D E
(B C) -> A C D E
(B D) -> A C D E
(B E) -> A C D E

For map (C -> A B D E):

(A C) -> A B D E
(B C) -> A B D E
(C D) -> A B D E
(C E) -> A B D E

For map (D -> A B C E):

(A D) -> A B C E
(B D) -> A B C E
(C D) -> A B C E
(D E) -> A B C E

And finally for map (E -> B C D):

(B E) -> B C D
(C E) -> B C D
(D E) -> B C D

Before we send these key-value pairs to the reducers, we group them by their keys and get:(A B) -> (A C D E) (B C D)
(A C) -> (A B D E) (B C D)

(A D) -> (A B C E) (B C D)
(B C) -> (A B D E) (A C D E)
(B D) -> (A B C E) (A C D E)
(B E) -> (A C D E) (B C D)
(C D) -> (A B C E) (A B D E)
(C E) -> (A B D E) (B C D)
(D E) -> (A B C E) (B C D)

Each line will be passed as an argument to a reducer. The reduce function will simply intersect the lists of values and output the same key with the result of the intersection. For example, reduce ((A B) -> (A C D E) (B C D)) will output (A B): (C D) and means that friends A and B have C and D as common friends.

The result after reduction is:

(A B) -> (C D)
(A C) -> (B D)
(A D) -> (B C)
(B C) -> (A D E)
(B D) -> (A C E)
(B E) -> (C D)
(C D) -> (A B E)
(C E) -> (B D)
(D E) -> (B C)

Now when D visits B's profile, we can quickly look up (B D) and see that they have three friends in common, (A C E).

Part B

Q: Find two friends who have the highest number of mutual friends.

Part C

Q: show the mutual friends between two people whose first letter contains ‘1’ or ‘5’.

Note: We need the mutual friends list to have values of ‘1’ and ‘5’. PLEASE NOTE THAT THE 2 PEOPLEFOR WHOM YOU ARE FINDING MUTUAL FRIENDS DO NOT NEED TO BE ‘1’ OR ‘5’.

Example of correct answers:
35678, 6789 -> 134567, 1456, 16799, 56789
45678, 23456 -> 56789, 54321, 14789

Part D

Q: Find the average number of mutual friends in the dataset and show the friends who have mutual friends above the average number of mutual friends.