Problem:

Write a MapReduce program in Hadoop that implements a simple “People You Might Know” social network friendship recommendation algorithm. The key idea is that if two people have a lot of mutual friends, then the system should recommend that they connect with each other.

Algorithmic approach to the problem.

- First, we read the file and then store them in variable as resilient distributed dataset.
- Then we map the text data in list of list of form by split the data into id and a list then we use the map function to convert the string input from the friends of the user list and convert them into integer value.
- Next, we map the friend list of all the users to the each of the friends with the key in this format ((user, friend), key)
- Finds the number of mutual friends between users who are not already friends and filter out all the users that are already friends, we get the output as (friend-pair), sum of mutual
  friends the friend-pair has in common)
- We map the input from ((friend recomendation1, friend recomendation2),mutual friend_count)) to
  (friend_recomendation1,(friend_recomendation2,mutualfriend_count))
- #We then from the given list of people we filter out the list of top 10 recommended friends in this format (user_id, [list of 10 recommended friends]).

All in all we first read the file then we map the friends with mutual friends then we use the reduce function to group and count the number of common friends each friends has and then for the given user we filter out the top ten recommendations

Output

[(924, [439, 2409, 6995, 11860, 15416, 43748, 45881]),
(8941, [8943, 8944, 8940]),
(8942, [8939, 8940, 8943, 8944]),
(9019, [9022, 317, 9023]),
(9020, [9021, 9016, 9017, 9022, 317, 9023]),
(9021, [9020, 9016, 9017, 9022, 317, 9023]),
(9022, [9019, 9020, 9021, 317, 9016, 9017, 9023]),
(9990, [13134, 13478, 13877, 34299, 34485, 34642, 37941]), (9992, [9987, 9989, 35667, 9991]),
(9993, [9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941])]