

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.

Input: Download the input file from the link:

<http://dlab.csu.edu.cn/cloudcomputing2016/download/LiveJournal.txt>

The input file contains the adjacency list and has multiple lines in the following format:

0 1,2,3,...

Here, “0” is a unique integer ID corresponding to a unique user and “1,2,3...” is a comma-separated list of unique IDs corresponding to the friends of the user with the unique ID. Note that the friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. The data provided is consistent with that rule as there is an explicit entry for each side of each edge.

Algorithm: Let us use a simple algorithm such that, for each user U, the algorithm recommends $N = 10$ users who are not already friends with U, but have the largest number of mutual friends in common with U.

Output: The output should contain one line per user in the following format: 1 2,4,5,10,...

where “1” is a unique ID corresponding to a user and “2,4,5,10,...” is a comma-separated list of unique IDs corresponding to the algorithm’s recommendation of people that might know, ordered by decreasing number of mutual friends. Even if a user has fewer than 10 second-degree friends, output all of them in decreasing order of the number of mutual friends. If a user has no friends, you can provide an empty list of recommendations. If there are multiple users with the same number of mutual friends, ties are broken by ordering them in a numerically ascending order of their user IDs.