# PEAK TRAFFIC

## CHALLENGE DESCRIPTION:

Credits: This challenge is from the facebook engineering puzzles

Facebook is looking for ways to help users find out which friends they interact with the most on the site. Towards that end, you have collected data from your friends regarding who they interacted with on the site. Each piece of data represents a desirable but one-way interaction between one user of Facebook towards another user of Facebook. By finding groups of users who regularly interact with one another, you hope to help users determine who among their friends they spend the most time with online.

Being a popular user, you have collected a lot of data; so much that you cannot possibly process it by hand. But being a programmer of no small repute, you believe you can write a program to do the analysis for you. You are interested in finding clusters of users within your data pool; in other words, groups of users who interact among one another. A cluster is defined as a set of at least three users, where every possible permutation of two users within the cluster have both received and sent some kind of interaction between the two.

With your program, you wish to analyze the collected data and find out all clusters within.

## INPUT SAMPLE:

Your program should accept as its first argument a path to a filename. The input file consists of multiple lines of aggregated log data. Each line starts with a date entry, whose constituent parts are separated by single white spaces. The exact format of the date always follows the examples given below. Following the date is a single tab, and then the email address of the user who is performing the action. Following that email is another single tab and then finally the email of the Facebook user who receives the action. The last line of the file may or may not have a newline at its end.

```
Thu Dec 11 17:53:01 PST 2008    a@facebook.com    b@facebook.com
Thu Dec 11 17:53:02 PST 2008    b@facebook.com    a@facebook.com
Thu Dec 11 17:53:03 PST 2008    a@facebook.com    c@facebook.com
Thu Dec 11 17:53:04 PST 2008    c@facebook.com    a@facebook.com
Thu Dec 11 17:53:05 PST 2008    b@facebook.com    c@facebook.com
Thu Dec 11 17:53:06 PST 2008    c@facebook.com    b@facebook.com
Thu Dec 11 17:53:07 PST 2008    d@facebook.com    e@facebook.com
Thu Dec 11 17:53:08 PST 2008    e@facebook.com    d@facebook.com
Thu Dec 11 17:53:09 PST 2008    d@facebook.com    f@facebook.com
Thu Dec 11 17:53:10 PST 2008    f@facebook.com    d@facebook.com
Thu Dec 11 17:53:11 PST 2008    e@facebook.com    f@facebook.com
Thu Dec 11 17:53:12 PST 2008    f@facebook.com    e@facebook.com
```

Every line in the input file will follow this format, you are guaranteed that your submission will run against well formed input files.

## OUTPUT SAMPLE:

You must output all clusters detected from the input log file with size of at least 3 members. A cluster is defined as N >= 3 users on Facebook that have send and received actions between all possible permutations of any two members within the cluster.

Your program should print to standard out, exactly one cluster per line. Each cluster must have its member user emails in alphabetical order, separated by a comma and a single space character each. There must not be a comma (or white space) after the final email in the cluster; instead print a single new line character at the end of every line. The clusters themselves must be printed to standard out also in alphabetical order; treat each cluster as a whole string for purposes of alphabetical comparisons. Do not sort the clusters by size or any other criteria.

```
a@facebook.com, b@facebook.com, c@facebook.com
d@facebook.com, e@facebook.com, f@facebook.com
```

Finally, any cluster that is a sub-cluster (in other words, all users within one cluster are also present in another) must be removed from the output. For this case, your program should only print the largest super-cluster that includes the other clusters. Your program must be fast, efficient, and able to handle extremely large input files.