1. Problem Statement

**The background**

A large company with an employee base of ~5000000 is very sensitive about fulfilling its employees technical knowledge needs. Technical Growth of every new employee in the organization is a high priority.

The company has an internal social network that spans across the organization, and employees usually have their friends and team-mates as their connections.  
  
With a large number of new employees, it’s important to bring them in touch with the experts to encourage meaningful interactions. To solve this challenge, the company wants to hire technical experts to find out all connections between new employees and Subject Matter Experts through the company social network, so that the new employees can be easily introduced to experts. They also want to know the route so that there is better flow of knowledge through these chains.

The list of all “New Joinees” and “SMEs(Subject Matter Experts)” is available. The social network is internal to the company, so all connections are also known and complete graph and graph format is visible.

The company head promises a grand prize to any employee who brings out a list of all paths between new-joinees and SMEs and solves this in the minimum time.

1. To Find…

* **All Paths between Employees who are ‘New Joinees’ and ‘SMEs’**
* **If there are multiple paths between same pair of New Joinee and SME, any one path is needed.**
* **For all paths, print the “New joinee”, “First node in the path” and “SME” nodes. If the path has only two nodes then print just the “New joinee” and “SME” nodes.**

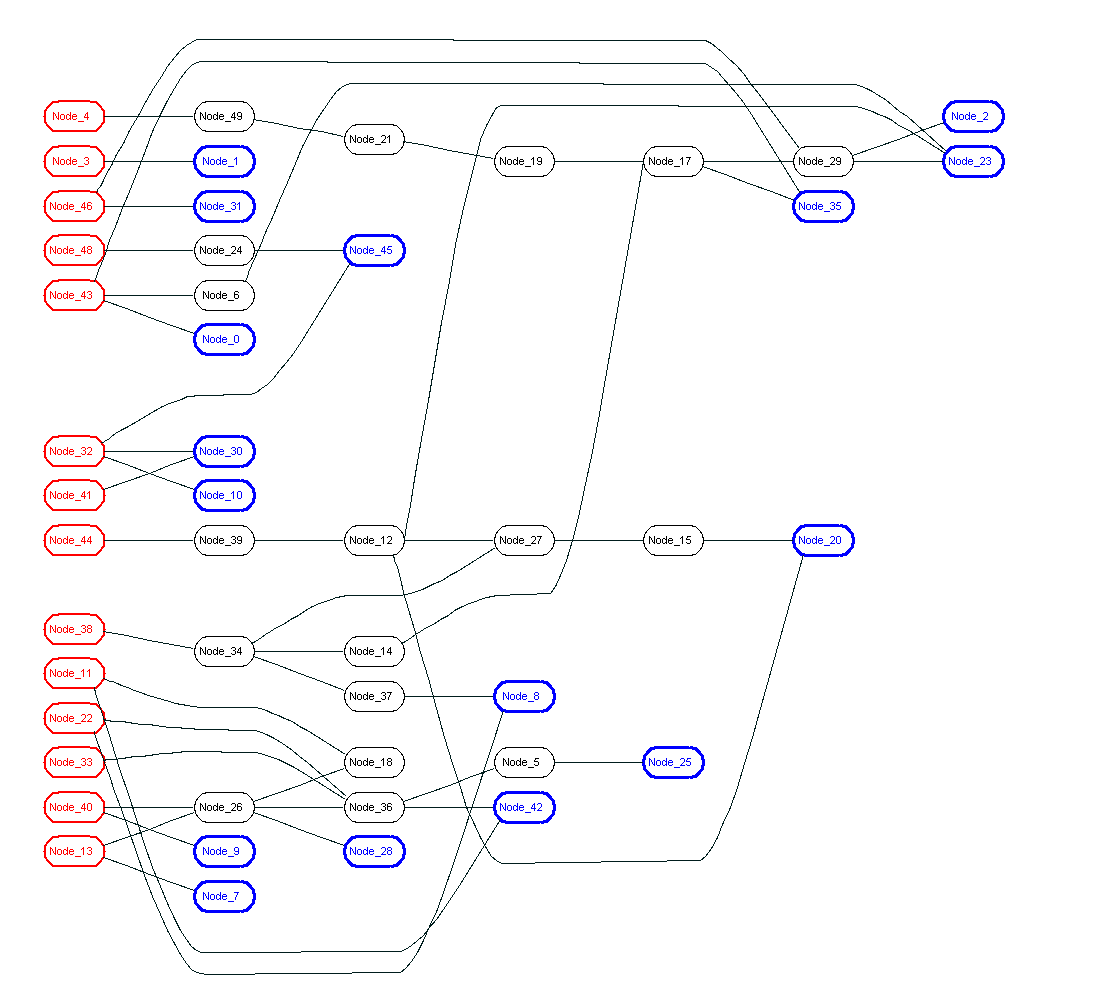
For submission, the precise final output will be dumped in a file as per the format specified in Section 3 below.

See the image below

* Nodes in RED are company “New Joinee employees”
* Node in BLUE are the “Subject Matter Experts (SMEs)”
* Nodes in BLACK are other employees (i.e. friends, team-mates, friends of friends… of Company employees)

**Note:** You should consider SME nodes (BLUE) as dead ends and no further traversal with take place from these nodes for path tracing.

This also implies in a path trace, only BLACK nodes are acceptable as intermediate nodes.



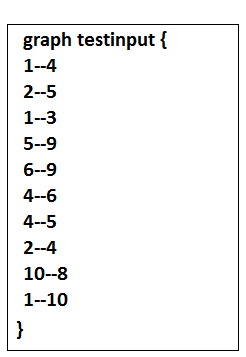
Input for this problem

There will be 3 input files

1. **Graph input text file:** A text file containing two nodes **per line**. This defines an undirected graph having connection between “New Joinees”, “friends” and “SMEs”.

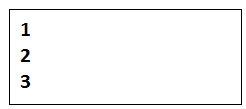
**File name** : “***input\_graph***”

Sample input file



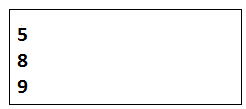
1. **New Joinee list file:** A text file containing node number of an employee **per line**.

**File name:** “***new\_employee\_list***”



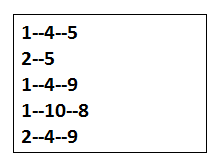
1. **SME list file** : A text file containing node number of a SME **per line**.

**File name:** “***sme\_list*** “



1. Output format file

**File name:** “***output***“

****

1. Programming language

Participants can choose either C/C++ or JAVA as a programming language for developing the solution on Linux platform.

For C/C++ - gcc/g++ would be available as default compiler

For JAVA - We will get jdk6/7 installed on the machines we offer for this contest.

5. Evaluation Criteria

Primarily, the criteria for winning would be the fastest correct solution.

We will start with a smaller problem size as input (in terms of graph netlist size) and as the contest will progress and submissions start coming, we will gradually raise the bar by introducing bigger graphs as input.

Participants are free to submit their solutions multiple times before the deadline. Please send your solutions to [code\_contest@mentor.com](mailto:code_contest@mentor.com) as a tarred and gzipped file with the following naming convention.

***<firstname>\_<lastname>\_CC2014.tar.gz***

Participants are encouraged to use multithreading and explore compiler optimizations. Submission should include a Makefile/Build-command along with source code.

6. Machines

The following machines are reserved for the duration of this contest:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl no. | hostname | OS | CPU | memory |
| 1 | innvc-cc01 | RHEL 6u3 | 8 core | 16 GB |
| 2 | innvc-cc02 | RHEL 6u3 | 8 core | 16 GB |