In this challenge, suppose we are looking to do social network analysis for prospective customers. We want to extract from their social network a metric called "closeness centrality".  
  
Centrality metrics try to approximate a measure of influence of an individual within a social network. The distance between any two vertices is their shortest path. The \*farness\*  
of a given vertex \*v\* is the sum of all distances from each vertex to \*v\*. Finally, the \*closeness\* of a vertex \*v\* is the inverse of the \*farness\*.  
  
The first part of the challenge is to rank the vertices in a given graph by their \*closeness\*. The graph is provided in the attached file; each line of the file consists of two vertex names separated by a single space, representing an edge between those two nodes.  
  
You should deliver a git repository with your code and a short README file outlining the solution and explaining how to build and run the code. You may deliver your code in any programming language. There is no problem in using libraries, for instance for testing or network interaction, but please avoid using a library that already implements the core graph or social network algorithms.  
  
Don't shy away from asking questions whenever you encounter a problem. Also, please do get in touch at any moment if you believe the timeframe is unrealistic.  
  
References:  
 - Closeness Centrality: <http://en.wikipedia.org/wiki/Centrality#Closeness_centrality> - Shortest path: <http://en.wikipedia.org/wiki/Shortest_path_problem>