Implementation - To ease implementation, I iterated through the graph document and created a list of nodes. Each node contains an email as a unique identifier, three different scores (hub, PageRank and authority), and a list of incoming and outgoing links to that node. Once I had a list of these nodes, I proceeded to use it to implement the PageRank and HITS algorithms.

For PageRank, I first initialized each node in the list to have the inverse of the length of nodes. Then, for ten iterations, I performed the core of the algorithm. Before anything, I calculate the sum of the sink nodes‘ page ranks, defined as any node with zero outgoing values. Then I pass over the list again. For each node, I take into account the sinkValues by multiplying it against the lambda probability and adding it to the probability of a random jump (1-lambda), and finally dividing the entire value by the amount of nodes in the equation.

Next, I sum the page rank of each incoming link divided by the amount of outgoing links. This term, multiplied by the lambda that we follow one of the links, plus the random node probability and sink node value, gives us the page rank for a single node. This process is completed over each nodes a total of 10 times to give us a converging result.

For HITS, I initialized the values to be N^-.5. Initializing the values to 1, as the article on Wikipedia suggests, does not cause the values to change (at least for the final values). For ten iterations, I summed the all-incoming hubs to calculate the authority and all-outgoing authorities to calculate the hubs for each node in the graph. At the end of each step, it was necessary to normalize the values, so that they would converge.

Visualization - To decide on my graph, I investigated the roster. Here, I found several instances where prominent members of the company had registered themselves under different email addresses. I decided to merge these instances into a single entity to represent an individual.

To speed up the implementation however, I decided to focus only on the individuals who had proved important in the PageRank and HITS algorithms. I took the top ten entries from PageRank, the Hubs, and the Authorities, and also took the hubs with the most incoming links, and the authorities with the most outgoing links. If any of these links were listed as a duplicate, I swapped their names about. Since several individuals had decided to list themselves under different names, I decided to process the entire list of emails. I created a rudimentary email checker. If Andrew S Fastow wanted an email, he could choose Andy.S.Fastow, Afastow, Andrew.Fastow, AndyFastow, and a number of @s - @enron.com, @enroncompany.com, etc. I choose to assume that an email needed a last name, and a matching first letter to be similar, and that the smaller one had to be fully contained within the larger. I also removed any numbers and periods from the list to aid in detection.

This criteria meant that ‘grace.rodriguez@enron.com' and 'glenda.rodriguez@nerc.net' would be classified correctly as false, 'mark.taylor@enron.com' and 'mtaylor587@aol.com' would be correctly true, and 'richard.shapiro@enron.com', 'rickshapiro@hotmail.com' would be a false positive. These are all valid examples of emails I corrected – I was not able to find a true negative for example, but there probably was. There were also emails of the form ‘skilling@enron.com’ that I had to ignore – a company of this size will have duplicates of last names (i.e. there was a mark skilling as well as Jeffrey Skilling). If the investigation needed to be thorough, such information could be easily retrieved by a slight modification to my function. After spell checking the nodes and replacing all matches with a single email, I trimmed all possible nodes with whose correspondence was not over 50. This eliminates trivial information, and makes the important information easy to visualize.

I then decided to ignore certain persons. First, the node Pete Davis is given as a broadcast node in the roster file. While naming an automatic broadcaster is odd, there was a lack of correspondence between Davis and a select group of individuals is intriguing. Davis regularly sends out 3000+ emails to certain individuals, but not these. Either they are too low to receive said emails, or too high to be bothered by such trivialities. But the roster lists Steven Kean as Vice President, and all of these individuals send out far too many emails to be of little importance. For that reason, I would classify this cluster as important.

Secondly, the structured flow of the information was intriguing. Certain individuals received information in a one way direction, but did not return information. However, the same cluster that did not pass receive information from Pete Davis had dependencies on each other. I decided that I would focus on this cluster.

A few facts about the graph – each node contains the name of an individual, with an arrow pointing to a list of email addresses the individual used. Included are seven individuals who had high levels of emails, and three others – Kenneth Lay, Andrew Fastow, and Jeffrey Skilling. These individuals represent the top of the management (CEO, CFO and Present respectively). However, their email amounts are staggering low in comparison. I would deem this highly suspicious, and either assume they had no knowledge of the scandal because they were out of touch with the company, or more likely were involved in a coverup of a greater magnitude. Perhaps the list of emails do no contain all the information, or they have hidden their activities under different email addresses.

Finally, for visualization, I manually added a few touches to the graph – renamed all the emails to a name, and placed the information in the graph as square nodes. It is not as visually appealing, but it conveys a higher level of information. The original file can be attained through running the code.