Design and Analysis of Algorithms
Assignment 4
Submission Date: April 25, 2019

Problem 1

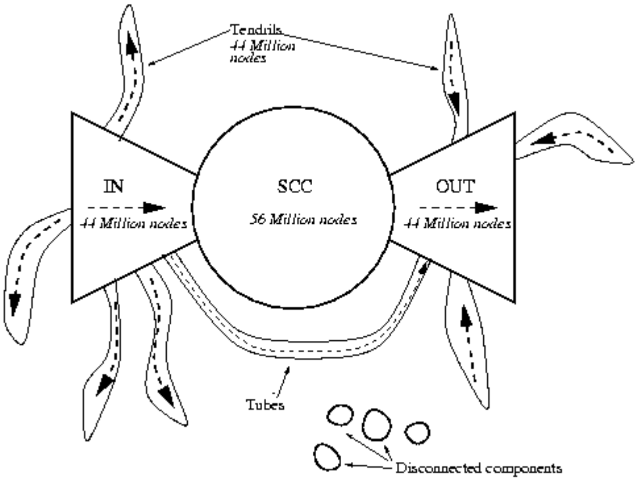
Implement the algorithm in C++ that takes a directed graph as input, compute the strongly connected components as discussed in class and output the components in a text file. Also construct the component graph for the input graph. Download the file SCC.txt (attached) and test your code on that input. The file contains the edges of a directed graph. Vertices are labeled as positive integers from 1 to 875714. Every row indicates an edge, the vertex label in first column is the tail and the vertex label in second column is the head (recall the graph is directed, and the edges are directed from the first column vertex to the second column vertex). So for example, the 9*th* row looks like : "2 47646". This just means that the vertex with label 2 has an outgoing edge to the vertex with label 47646.

In the output file you must first output the size of the strongly connected component and then its vertices in comma separated list.

Note: First test your code on smaller inputs. Also manage your memory carefully because the size of the graph is very large.

Problem 2

As you know that we can model web as directed graph where web pages are nodes and if some page ‘a’ has a hyperlink for page ‘b’ then we can make a directed edge from ‘a’ to ‘b’. Broder et al.[2000] first try to capture the structure of the web graph. The authors discovered that the web graph can be partitioned into five major groups. One strongly connected component, IN set, OUT set, TENDRILS and TUBES. There are also some small disconnected components. They named it as bow-tie structure as shown in the following figure. There is one giant strongly connected component called SCC. IN set has the following property: there is a directed path from each node of IN to (all the nodes of) SCC. OUT set comprises of all the nodes that are reachable from every vertex/node in SCC. There were also TENDRILS hanging off IN and OUT, containing nodes reachable from portions of IN or nodes going to portions of OUT. TENDRILS going from IN to OUT without touching SCC formed TUBES. There are also some DISCONNECTED components isolated from the rest of the graph.



Various other graphs/networks also exhibit this bow-tie structure. In this problem, we explore the structure of a directed social network, namely the Epinions Social Network (file attached). This is a who-trust-whom online social network of a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other. All the trust relationships interact and form the Web of Trust which is then combined with review ratings to determine which reviews are shown to the user. Here each node is a user and there is an edge from node a to b if user a trusts b. This network contains 75879 nodes and 508837 edges. Every row in the file indicates an edge, the vertex label in first column is the tail and the vertex label in second column is the head

We will try to determine whether bowtie structure exists in Epinions network or not. For that we will compute the sizes of each region in the Epinions network. How many nodes are in the SCC, IN, OUT, TENDRILS+TUBES (referring to TENDRILS and TUBES combined), and DISCONNECTED regions? Write a C++ program to calculate the size of each of the above components in Epinions network(both number of nodes and fraction of total nodes) . Use the program of Problem 1 to compute strongly connected components. You may also need to compute weakly connected components (A directed **graph** is called **weakly connected** if replacing all of its directed edges with undirected edges produces a **connected** (undirected) **graph**.) and use BFS.

Broder et al. found in their paper that given a pair of randomly chosen start and finish webpages, one can get from the start page to the finish page by traversing links only approximately 25% of the time. For the Epinions network, what is the probability that a path exists between two nodes chosen uniformly from the graph? (Hint: One way you can approach this question is to first sample many node pairs at random and then report the fraction of times pairs were reachable. You may need to compute shortest paths for this part. Start with 10 pairs and then double the node pairs. Plot a graph that has number of pairs on the x-axis and fraction of reachable pairs on the y-axis. As the number of node pairs sampled grows larger, what would you expect the fraction of reachable pairs to converge to?

What to Submit:

You should submit the code developed for both the problems and a report of your finding both for Problems 1 & 2. For Problem 1 give the number of strongly connected components. Also for each component you should report the number of vertices in each component. For Problem 2, give the sizes of SCC, IN, OUT, TENDRILS+TUBES. Also give a plot that has number of pairs on the x-axis and fraction of reachable pairs on the y-axis. As the number of node pairs sampled grows larger, fraction of reachable pairs converge. What is that fraction?

Note: Plagiarism of any sort is not acceptable and will be severely punished.