To answer the first 3 questions you will need to use some simple UNIX commands. If you are not familiar with the commands, please read the UNIX man pages: In a command window type \texttt{man command-name}; for example to get information about the \texttt{host} command, type \texttt{man host}.

1. (5 marks) Use the \texttt{host} command (/usr/sbin/host or /usr/bin/host, depending on where you are logged on) in any UNIX server where you have an account, to determine the IP address of at least one of the computers in the Department of Computer Science’s computer network (type \texttt{host <computer name>}). Report both, the symbolic name and IP address of the machine that you have selected. Assuming the classful IP addressing scheme, which class of IP addresses (A, B, C, D, or E) are assigned to the computers in the University’s network? Which is the network prefix for that IP addresses? Up to how many different computers can belong to this network?

2. (15 marks) Use the \texttt{ping} command (/usr/sbin/ping) to estimate the number of computers in the UWO network assuming the classful IP addressing scheme. To do this, generate a list \( L \) of random IP addresses with the network number that you determined in the previous question. Issue a \texttt{ping} command for each one of these IP addresses and determine the fraction of \( L \) that corresponds to actual computers. Use this fraction to estimate the size of the University’s network.

   Show the list \( L \) that you used, the fraction of addresses that correspond to actual machines, and your estimate for the size of the network.

3. (10 marks) Use the \texttt{traceroute} (/usr/sbin/traceroute) UNIX command to find the number of hops between your computer and remote destinations (e.g. to servers storing Web sites). The maximum number of hops between two computers is called the \textit{diameter of the Internet}. Report on the maximum number of hops that you can find. Include the list of servers contacted by \texttt{traceroute} and their possible geographical locations (if you can find them). To try to find the geographical location of an IP address you can use one of several available Web services, like \texttt{IP2Location} (www.ip2location.com), \texttt{WhatIs MyIPaddress} (www.whatismyipaddress.com), or \texttt{IP Address Location} (www.ipaddresslocation.org).

4. In class we designed an algorithm for solving the leader election problem on a synchronous ring. Design a synchronous algorithm for solving the leader election problem, but this time assume that

   \begin{itemize}
   \item[i.] the processors are connected in a line as shown in the figure below, and
   \item[ii.] the leader is the processor located in the middle. For example, for the following network the leader will be processor with id 27.
   \end{itemize}

   \begin{center}
   \begin{tikzpicture}
   \draw (0,0) -- (1.5,0) node at (0,0) {13} node at (1.5,0) {6} node at (3,0) {27} node at (4.5,0) {85} node at (6,0) {53};
   \end{tikzpicture}
   \end{center}

   Each processor \( p_i \) has a unique identifier. Let the number of processors in the system be \( n \) and assume that \( n \) is odd. The only information that a processor \( p_i \) knows about the network is its right neighbour (if any) and its left neighbour (if any). The processors do not know the value of \( n \).
• (17 marks) Give in pseudocode (similar to the examples that we have done in class) a synchronous distributed algorithm for solving this problem. Your algorithm must terminate after a finite time and at the end each processor must know the id of the leader.
  You must also explain in English how your algorithm works.
• (8 marks) Show that your algorithm is correct.
• (10 marks) Compute the time complexity and communication complexity of your algorithm in the worst case.

5. Let a set of \( n \) processors be connected in a ring. In class we studied an algorithm for the leader election problem which assumes that each processor knows which of its 2 neighbours appears to its right and which one appears to its left in a clockwise ordering of the ring (hence, in that algorithm we could use the instruction “send mssg to the right neighbour”). Suppose that now a processor \( p_i \) only knows that it has two neighbours \( p_j \) and \( p_k \), but it does not know which one appears to its right and which one appears to its left in a clockwise ordering of the ring.

Modify the leader election algorithm given in class so that it works in this new situation.

• (17 marks) Give in pseudocode (similar to the examples that we have done in class) a synchronous distributed algorithm for solving this problem.
  You must also explain in English how your algorithm works.
• (8 marks) Show that your algorithm is correct.
• (10 marks) Compute the time complexity and communication complexity of your algorithm in the worst case.