1. Consider the undirected graph given below and answer the questions that follow.

(a) (5 points) Draw an adjacency list for the graph above.
(Note that the adjacency list may not be unique and hence there may be multiple correct answers to this part.)
(b) (5 points) State the order in which the vertices of the graph get marked visited when \texttt{explore}(G, s) is executed.

\textit{(Please note that the answer to this part is dependent on your answer to part (a) since the order in which neighbors of a particular vertex \(v\) are considered depends on the order of the neighbors in the linked list corresponding to \(v\).)}

(c) (5 points) State the order in which the vertices of the graph get marked visited when \texttt{explore}(G, h) is executed.

\textit{(As in part (b), the answer to this part depends on your answer to part (a).)}
2. (20 points) A source in a directed graph is a node that has no edges going into it. Give a linear-time algorithm that takes as input a directed graph in adjacency list format, and outputs all of its sources. Give pseudocode and discuss running time.
3. (20 points) For every node $u$ in an undirected graph, let $t$-$degree[u]$ be the sum of degrees of $u$'s neighbors. Design an algorithm to compute the entire array of $t$-$degree[.]$ values in linear time, given an undirected graph in adjacency list format. Give pseudocode and discuss running time.
4. (20 points) An undirected graph is said to be connected iff every pair of vertices in the graph are reachable from one another. Prove the following statement:

Any connected undirected graph with $n$ nodes has at least $(n - 1)$ edges.
5. (25 points) Suppose a CS program consists of $n$ mandatory courses. The prerequisite graph $G$ has a node for each course, and an edge from course $u$ to course $v$ if and only if $u$ is a prerequisite for $v$. Design an algorithm that takes as input the adjacency list of the prerequisite graph $G$ and outputs the minimum number of quarters necessary to complete the program. You may assume that there is no limit on the number of courses a student can take in one quarter. Your algorithm should have linear running time. Give pseudocode and discuss running time.