CSCD320 Homework6, Eastern Washington University, Spokane, Washington.

Name: **EWU ID:**

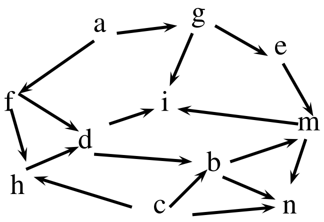
Please follow these rules carefully:

1. Verbal discussions with classmates are encouraged, but each student must independently write his/her own solutions, without referring to anybody else's solution.
2. The deadline is sharp. Late submissions will **NOT** be accepted (it is set on the Blackboard system). Send in whatever you have had by the deadline.
3. Submission must be computer typeset in the **PDF** format and sent to the Blackboard system. I encourage you all to use the LaTeX system for the typesetting, as what I am doing for this homework as well as the class slides. LaTeX is a free software used by publishers for professional typesetting and are also used by nearly all the computer science and math professionals for paper writing.
4. Your submission PDF file must be named as: **firstname_lastname_EWUID_cscd320_hw6.pdf**
   (1) We use the underline '_' not the dash '-'.
   (2) All letters are in the lower case including your name and the filename's extend.
   (3) If you have middle name(s), you don't have to put them into the submission's filename.
5. You are NOT allowed to share this homework in any way with others who are not in this class.

Problem 1 (20 points)**.** *Print the BFS and DFS that starts from the vertex f of the following graph. If a node has multiple next-hops, then search the next-hops in the order of their vertical coordinates from the lower ones to the higher ones.*

For example, node `d` *has two next-hop neighbors* `b` *and* `i`*. Say you are now visiting* `d`*, and you have not visited* `b` *and* `i` *yet. Next you will first visit* `b` *because its vertical coordinate is lower.*

Note that the textbook's DFS algorithm tries all vertices as the starting node, but here only need to show the print of the DFS that starts from the vertex f. You do not have to show the trace of the procedure, but instead you only need to print the output of the BFS and DFS. For example: `fh...`



Problem 2 (20 points)**.** *Tree can be viewed as a special type of graph and thus can be represented by an adjacency list or an adjacency matrix. You can view the binary tree as a direct graph where the direction of each edge is pointing from the parent node to the child node. Now you are given the root node of a binary tree and also the tree's size n. You are also guaranteed that each node has a distinct key drawn from $\{1, 2, \ldots, n\}$. (1) Design an efficient algorithm to create the adjacency list representation of the given binary tree, where each vertex can be represented by the key stored at the corresponding tree node. Give your algorithmic idea and its pseudocode; (2) Analyze the time complexity of your algorithm in the big-oh notation and make the bound as tight as possible.*

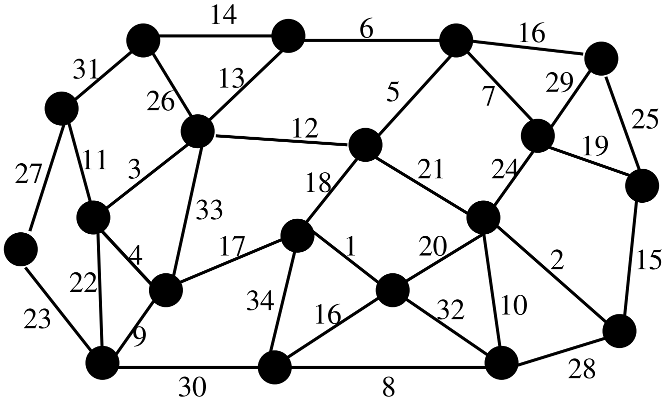
CSCD320 Homework6, Eastern Washington University, Spokane, Washington.

Name: **EWU ID:**

Problem 3 (20 points)**.** *The graph BFS algorithm*[1] *that we have discussed in class uses the adjacency list representation for the graph. Now you are asked to (1) give the same graph BFS algorithm but use the adjacency matrix representation of the graph; (2) analyze the time complexity of your algorithm and compare it with the time complexity of the one that uses the adjacency list representation.*

Problem 4 (20 points)**.** *The graph DFS algorithm*[2] *that we discussed in class uses the recursion-based structure. Now you are asked to give a non-recursion based graph DFS algorithm. Analyze the time cost of your algorithm. Hint: use and maintain your own stack.*

Problem 5 (20 points)**.** *Show a minimum spanning tree (MST) of the following connected undirected graph by using any method. You don't have to trace the algorithm that you use, but instead you can just show the MST by making the tree edges in a different color. Give the total weights of the MST that you find.*



[1] The `BFS()` function at page 595 of CLRS, 3rd Ed.

[2] The `DFS()` and `DFS_VISIT()` functions at page 604 of CLRS, 3rd Ed.