CSCI 3230, Fall 2018 Assignment 4

Posted: October 27 **Due before 11:00 pm, Friday, November 16, 2018**

Upload to "Assignment 4" folder in Folio Dropbox with file name "**Lastname_ID.pdf**".

(Mandatory assignment cover-sheet; without it, your work will not be marked.)

Submitting student: Blair Weaver Your mark: □ out of 100

Collaborating classmates: Matthew Moye

Other collaborators: ______________________

References other than the textbook and handouts:

geeksforgeeks.com

____

____

____

Regardless of the collaboration method allowed, you must always properly acknowledge the sources you used and people you worked with. Your professor reserves the right to give you an exam (oral, written, or both) to determine the degree that you participated in the making of the deliverable, and how well you understand what was submitted. For example, you may be asked to explain any solution that was submitted and why you choose to write it that way. This may impact the mark that you receive for the deliverable.

So, whenever you submit a deliverable, especially if you collaborate, you should be prepared for an individual inspection/walkthrough in which you explain what every line of assignment does and why you choose to write it that way.

1. **(30 points)**: Suppose G is an directed graph with vertices labeled 1 through 8. Adjacent vertices for each vertex are listed as follows Vertex Adjacency.

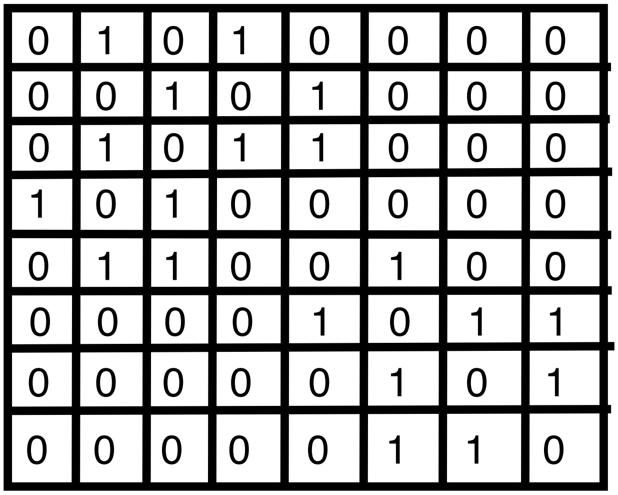
1 :	2, 4
2 :	3, 5
3 :	2, 4, 5
4 :	1, 3
5 :	2, 3, 6
6 :	5, 7, 8
7 :	6, 8
8 :	6, 7

(a) Construct an Adjacency matrix from the given adjacency list.

(b) Write a code to order the vertices as they are visited in a **depth-first traversal** starting at vertex 1. What is the traversal order? Include screenshots of the program running and submit your java code in a separate file.

(c) Write a code to order the vertices as they are visited in a **breadth-first traversal** starting at vertex 1. What is the traversal order? Include screenshots of the program running and submit your java code in a separate file.

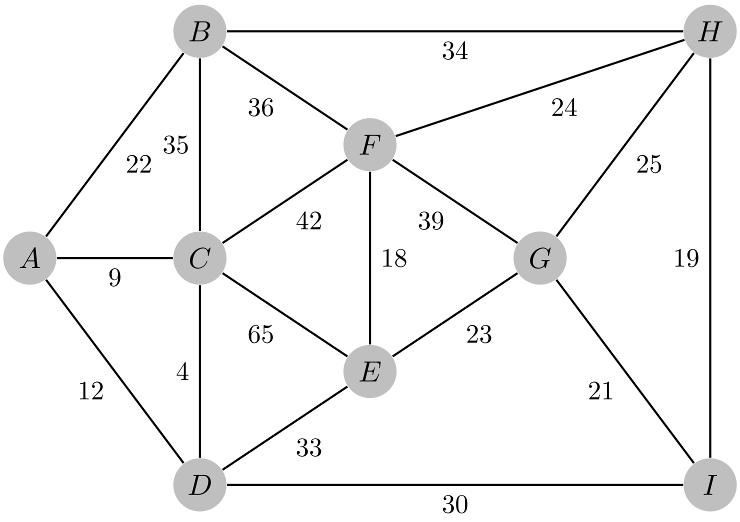
Solution:



0	1	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	1	0	0	0
1	0	1	0	0	0	0	0
0	1	1	0	0	1	0	0
0	0	0	0	1	0	1	1
0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0

B) 1,2,3,4,5,6,7,8
C) 1,2,4,3,5,6,7,8

2. **(20 points)** Consider the following undirected weighted graph.



(a) Give the list of edges in the minimum spanning tree in the order that **Kruskal's algorithm** inserts them. Please implement the Kruskal's algorithm. Include screenshots of the program running and submit your java code in a separate file.

(b) Give the list of edges in the minimum spanning tree in the order that **Prim's algorithm** inserts them, assuming that it starts at vertex F. Please implement the Prim's algorithm. Include screenshots of the program running and submit your java code in a separate file.

Solution:

A) C-D, C-A, E-F, H-I, G-I, A-B, E-G, D-I
B) F-E, E-G, G-I, I-H, I-D, D-C, C-A, A-B

3. **(20 points)** For each of the six questions in parts $(a) - (c)$, answer in terms of big-Oh and the number of vertices in the graph $|V|$.

   (a) Suppose a graph has no edges.

      i. What is the asymptotic space cost of storing the graph as an adjacency list?

      ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

   (b) Suppose a graph has every possible edge.

      i. What is the asymptotic space cost of storing the graph as an adjacency list?

      ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

   (c) Suppose an undirected graph has one node A that is connected to every other node and the graph has no other edges.

      i. What is the asymptotic space cost of storing the graph as an adjacency list?

      ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

   (d) Is an adjacency list faster or slower than an adjacency matrix for answering queries of the form, "is edge (u, v) in the graph"?

   (e) Is an adjacency list faster or slower than an adjacency matrix for answering queries of the form, "are there any directed edges with u as the source node"?

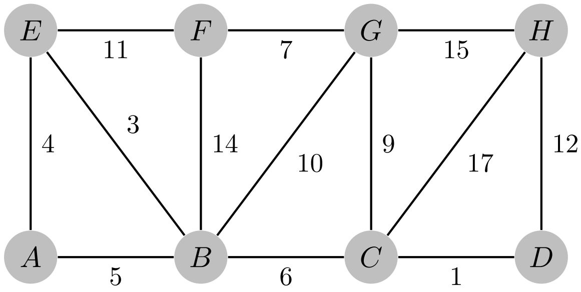
<u>**Solution:**</u>

A) i) n
   ii) n^2
B) i) n + (n(n-1))/2
   ii) n^2
C) i) n + (n-1)
   ii) n^2
D) Slower
E) Faster

n is the number of vertices just like the slides in class

4. **(30 points)**

(a) Run **Dijkstra's algorithm** on the weihgted graph below, using vertex A as the source. Write the vertices in the order which they are marked and compute all distances at each step.



(b) Write a program with Java to implement **Dijkstra's algorithm**. Test your program with the above example. Include screenshots of the program running and submit your java code in a separate file.

Solution:

Final: A, E, B, C, D, G, F, H

Step 1:
Marked: A=0

E=4
B=5
F, G, C, H, D = infinity

Step 2:
Marked: A=0, E=4

B=5
F=15
G, C, H, D = infinity

Step 3:
Marked: A=0, E=4,B=5

F=15
G=15
C=11
H, D = infinity

Step 4:
Marked: A=0, E=4,B=5
C=11

F=15
G=15
D=12
H=28

Step 5:
Marked: A=0, E=4,B=5
C=11,D=12

F=15
G=15
H=24

Step 6:
Marked: A=0, E=4,B=5
C=11,D=12,G=15

F=15
H=24

Step 7:
Marked: A=0, E=4,B=5
C=11,D=12,G=15,F=15

H=24

Step 8:
Marked: A=0, E=4,B=5
C=11,D=12,G=15,F=15,
H=24