

Graphs Continued

Recitation 14

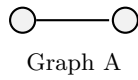
It's your last official recitation! Congratulations!! Let's get started. In this recitation you'll be covering:

- More questions covering graphs
- A coding exercise covering basic concepts with graphs

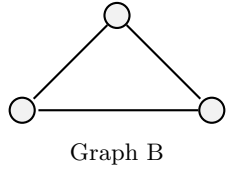
Graphs

1. Draw a graph models, starting the type of graph used, to represent airline routes where every day there are four flights from Boston to Newark, two flights from Newark to Boston, three flights from Newark to Miami, two flights from Miami to Newark, one flight from Newark to Detroit, two flights from Detroit to Newark, three flights from Newark to Washington, two flights from Washington to Newark, and one flight from Washington to Miami, with:
 - (a) an edge between vertices representing cities that have a flight between them (in either direction).
 - (b) an edge between vertices representing cities for each flight that operates between them) in either direction.
 - (c) an edge between vertices representing cities for each flight that operates between them (in either direction) plus a loop for a special sightseeing trip that takes off and lands in Miami
2. Describe a graph model that represents whether each person at a class knows the name of each other person at the class. Should the edges be directed or undirected? Should multiple edges be allowed? Should loops be allowed?
3. Show that in a simple graph with at least two vertices there must be two vertices that have the same degree.

4. How many subgraphs with at least one vertex does Graph A have?



5. How many subgraphs with at least one vertex does Graph B have?

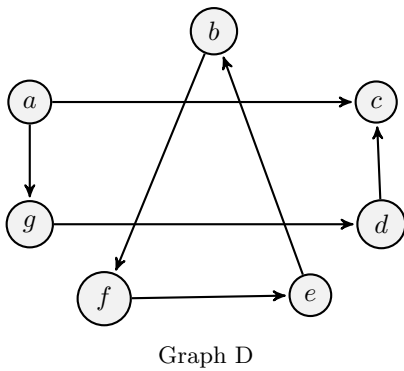
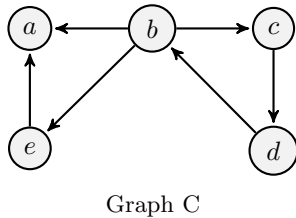


6. Are the simple graphs with the following adjacency matrices isomorphic?

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

7. Determine whether each of these graphs is strongly connected and if not, whether they are weakly connected

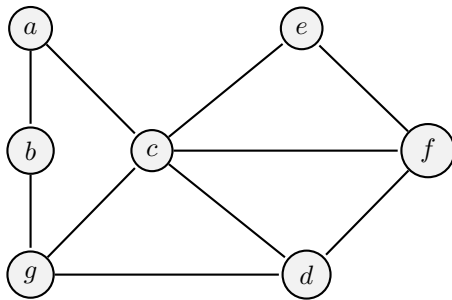


8. Consider Graph E, and answer the following

- (a) Will the adjacency matrix be symmetrical
- (b) Provide the adjacency matrix

(c) What does the sum of each row of the adjacency matrix represent?

(d) Provide the adjacency list

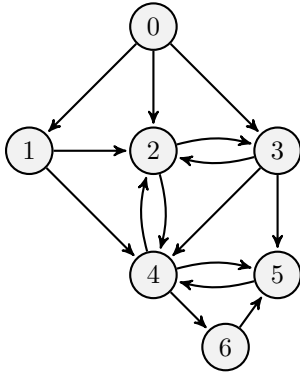


Graph E

9. Describe an adjacency matrix for a complete undirected graph

10. Do any complete graphs have Euler circuits? If so, describe the characteristics of such graphs

11. Trace the shortest-path algorithm for the graph in Graph H, letting vertex 0 be the origin

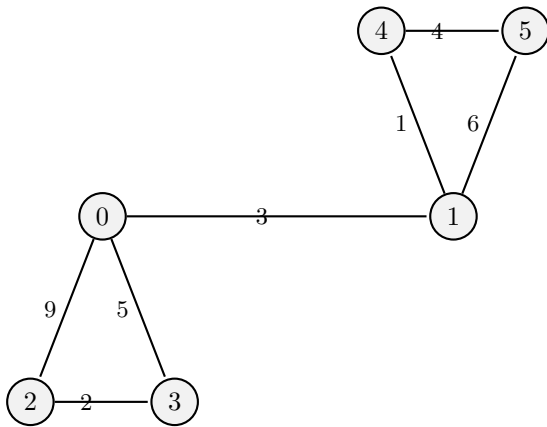


(a) Graph H

12. How many edges are there in a graph with 20 vertices each degree of 10?

Programming Exercise:

1. Create a new project in eclipse for this recitation
2. Import R14.jar from `~cs200/recit14/R14.jar`
3. The Graph class that you have been given is a representation of an undirected weighted graph using an adjacency list. Look at the class and understand what each method does.
4. Look at the provided code. We are going to manipulate the code in two ways:
 - (a) Create a main method in the Graph class. Let's test the methods that are already there. Go ahead and create this graph then create a `toString()` to print our your results.

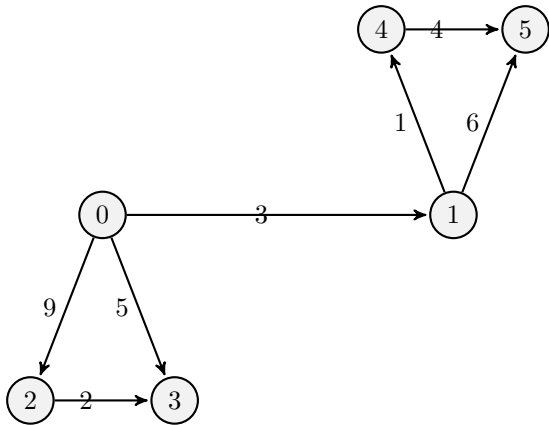


Your results should look like this:

```
0 => 1[3] => 2[9] => 3[5]
1 => 0[3] => 4[1] => 5[6]
2 => 0[9] => 3[2]
3 => 0[5] => 2[2]
4 => 1[1] => 5[4]
5 => 1[6] => 4[4]
```

- (b) Change the existing methods so that it represents a directed or undirected graph. At minimum you're going to have to modify the methods that you used to create the first Graph.
 - Since this lab allows for quite a bit of flexibility, keep track of your changes so that if you have questions you can provide a clear explanation for the TA.
- (c) After you have changed the Graph class to accommodate for a directed or undirected graph, you should not be able to switch between the two types. If you haven't already and your code needs it, add exception handling that would cover these situations.

(d) Last, create this directed graph and print it out:



Your results should look like this:

0 => 1[3] => 2[9] => 3[5]
1 => 4[1] => 5[6]
2 => 3[2]
3
4 => 5[4]
5

We are done!! Good luck studying for your finals :)