**LAST NAME , FIRST NAME**

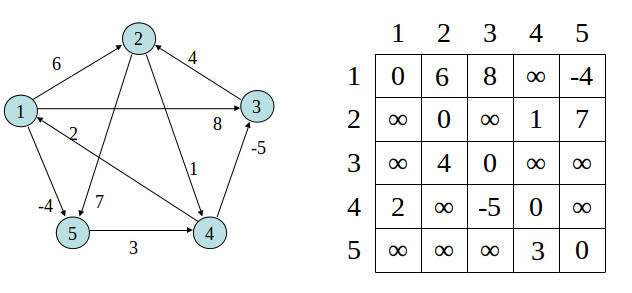
CS 4520/6520 Spring 2020

Please keep 1 problem per 1 page, and submit your work typed and in pdf format.

**Homework #3**

**Problem 1.**

For the following *directed* graph, what are alternative representations?



1. the adjacency matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 6 | 8 |  | -4 |
| 2 |  | 0 |  |  | 7 |
| 3 |  | 4 | 0 |  |  |
| 4 | 2 |  | -5 | 0 |  |
| 5 |  |  |  | 3 | 0 |

1. adjacency list

1->2,3,5

2->7

3->2

4->1,3

5->4

**Problem 2.**

1. Explain (in words is Ok, no code necessary) the *procedure* how representation of a graph in adjacency matrix could be transformed into representation in list of edges. You can assume the graph is a simple graph.

arrayList a = new arrayList(); l

inkedList <> l =new linkedList<>()

for i to length of row

linkedList <> k =new linkedList<>()

k.add(i) a.add(k)

for j to length of column

if (column(j) != 0 or null)

a.get(i).add(j)

1. From adjacency list to list of edges? (on simple graph)

hashmap <> h =new hashmap<>()

for i to length of array

for j to length of linkedlist

h.put(j,linkedlist(j))

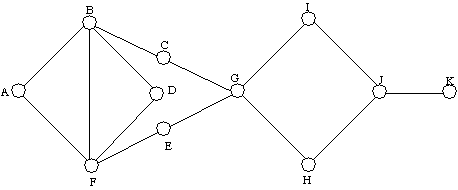
**Problem 3.**

Apply **Fleury’s** algorithm, beginning with vertex K, to find an Eulerian path in the following graph. In applying the algorithm, at each stage chose the edge (from those available) which visits the vertex which comes first in *alphabetical* order.

Does the graph have Eulerian **cycle** (circuit)? \_\_\_\_No\_\_\_\_\_\_\_ Eulerian **path**? \_yes\_\_\_\_\_\_

Which of the edges are **bridges**? \_\_\_\_(J,K)\_\_\_\_\_\_\_

Now apply Fleury’s algorithm, and explain how you go.



Resulting Eulerian path is: **K** -J,H,G,C,B,A,F,B,D,F,E,G,I,J

**Problem 4.**

Given a graph G, give the order in which nodes are traversed with BFS. After listing a node, add its BFS-label in parenthesis.

from source c: c(0), \_\_ d(1),e(1), f(2),g(3),a(4),b(5)\_\_\_\_\_\_\_\_\_

from source e: e(0),\_\_\_ d(1),e(1), f(2),g(3),a(4),b(5) \_\_\_

a

e

f

g

b

d

c

Is the graph connected? \_YES\_\_  
How BFS algorithm helps you determine that?

It can use BFS to reach out all vertices

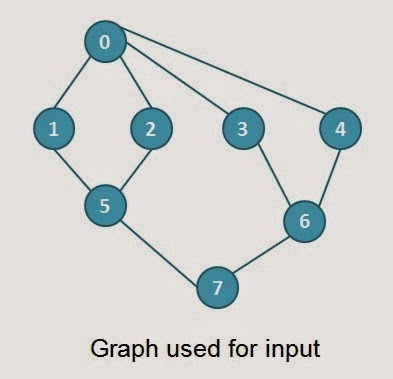
Is the graph bipartite? \_\_YES\_\_

Explain how BFS helps you determine that.

**You can divide it Into 2 independent sets U,V and each vertex in U,V have a vertex that connect to them.**

**Problem 5.** Traverse given graph by DFS, starting at vertex “0”.

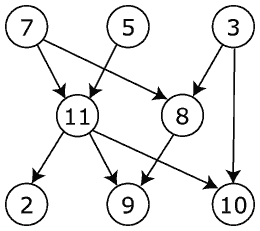
Mark which edges are forward and which are back. Give each vertex a timestamp.



**0(1|16) , 1(2|15),5(3|14),2(4|5),7(6|13),6(7|12),4(8|9),3(10|11)**

**Back (0,3) (0,2) (0,1)**

**Problem 6.** Perform topological sort for the given graph. Be careful about directions on edges.



**Sorted sequence of vertices:**

**7 5 11 2 3 8 9 10**