Homework #3

Due on Thursday, Feb. 24 before class starts.

1. The determinant of an n-by-n matrix

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

denoted *detA*, can be defined as a_{11} for n = 1 and, for n > 1, by the recursive formula

$$detA = \sum_{j=1}^{n} s_j a_{1j} detA_j$$

where $s_j = \begin{cases} 1 & j \text{ is odd} \\ -1 & j \text{ is even} \end{cases}$, a_{1j} is the element in row1 and column j, and A_j is the (n − 1)-by-(n − 1) matrix obtained from matrix A by deleting its row 1 and column j.

a. Set up a recurrence relation for the number of multiplications made by the algorithm implementing this recursive definition. You should give the initial conditions. (10pts)

b. Without solving the recurrence, what can you say about the solution's order of growth as compared to *n!* ? (10pts)

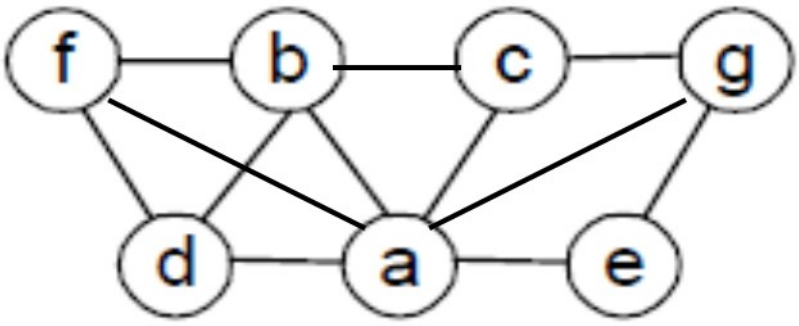
Hints:

a. Use the definition's formula to get the recurrence relation for the number of multiplications made by the algorithm.

b. Investigate the right-hand side of the recurrence relation. Compare the recurrence relation of this problem with the recurrence relation for computing *n!*. Computing the first few values of *M(n)* may be helpful, too.

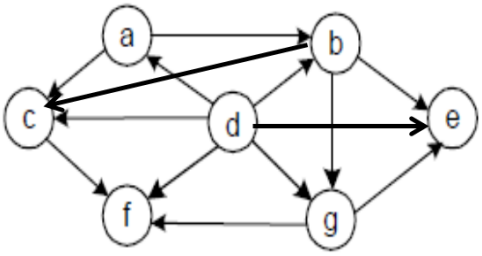
2. Sort the list **17, 7, 22, 33, 19, 2, 85** in an **ascending order** by **Insertion Sort**. You should show the process of sorting step by step -- **You must show the intermediate results after each insertion**. How many times of comparisons are used for this problem? (20pts)

3. Consider the graph

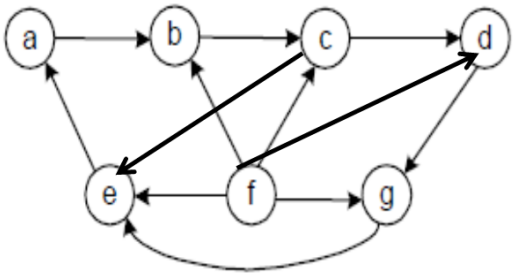


a. Write down the adjacency matrix and adjacency lists specifying this graph. (Assume that the matrix rows and columns and vertices in the adjacency lists follow in the alphabetical order of the vertex labels.) (10 pts)
b. Starting at vertex *a* and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresponding depth-first search forest including the tree edges and the other types of edges. **You should use solid edges to represent tree edges and dashed edges to represent other edges.** Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in which the vertices became dead ends (popped off the stack). (15 pts)
c. Starting the traversal at vertex *a* and resolve ties by the vertex alphabetical order, traverse the graph by breadth-first search and construct the corresponding breadth-first search tree. **You should use solid edges to represent tree edges and dashed edges to represent other edges.** Give the order in which the vertices were reached for the first time (enqueue) and the order in which the vertices became dead ends (dequeue). (15 pts)

4. Apply the DFS-based algorithm to solve the **topological sorting problem** for the following digraphs. You should also give the order of the vertices visited and the DFS forest including the tree edges and the other types of edges. (20 pts)



(a)



(b)