Design & Analysis of Algorithms - Spring 2012
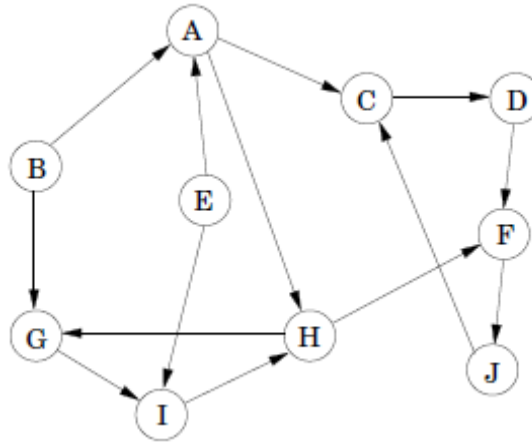# Mid Term 2

Q1.   (15)

Run the strongly connected components algorithm on the following directed graphs G. During DFS on G^R: whenever there is a choice of vertices to explore, always pick the one that is alphabetically first

In each case answer the following questions.
a.   In what order are the strongly connected components (SCCs) found?
b.   Which are source SCCs and which are sink SCCs?
c.   Draw the component graph.
d.   What is the minimum number of edges you must add to this graph to make it strongly connected?
e.   During the final phase of the algorithm, how it is determined, in linear time, without sorting, that the vertices are selected in reverse order of their finishing time.

Q2.   (5)

The BFS algorithm finds the shortest distance in terms of number of edges from source node to every other node in the give graph. Can the algorithm, with minor changes, be used to calculate shortest distances for a weighted graph with positive weights? Justify your answer.

Q3.   (10)

Design a linear-time algorithm which, given an undirected graph G and a particular edge **e** in it , determines whether G has a cycle containing **e**. If cycle containing **e** exists then it prints the cycle.

(Note: You have to give detailed pseudo-code. Precisely state your assumptions and don't make any unrealistic assumption for Q1 & Q3)

Q4.   (for section C only)   (10)

    i)   Convert the following 2-3 B-tree to 1-2 skip list
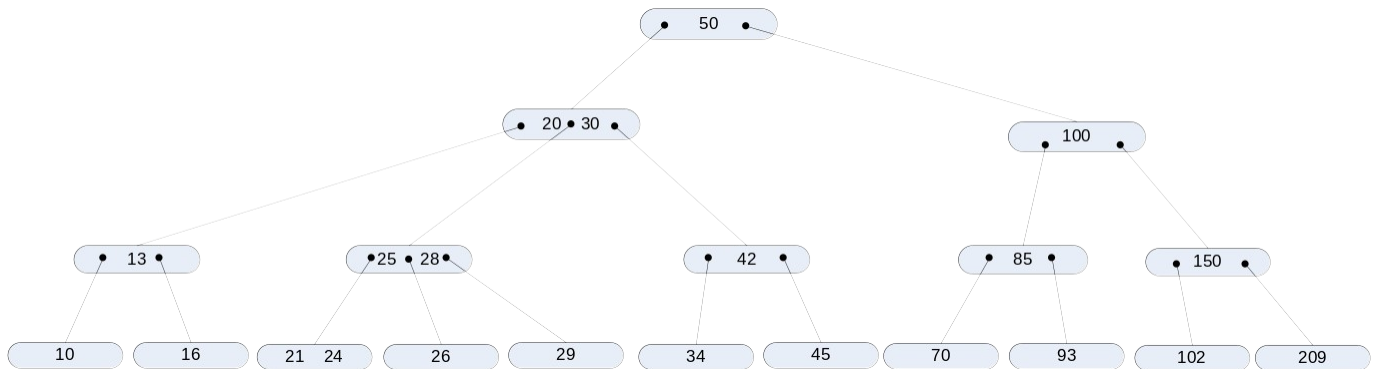
    ii)   Delete 100 from your skip list

    iii)   After 100 Delete 16 from the resulting skip list

Q4.   (for section A &B only)   (10)

    i.   Delete 150 from the following B-Tree of degree t=3

    ii.   Insert 148 in the resulting B-Tree of (i) above.

    iii.   Delete 43 from the following 1-2-3 skip list. (You should use "<=" comparison in the algorithm)
          algorithm)
          (Note: Please show all steps)