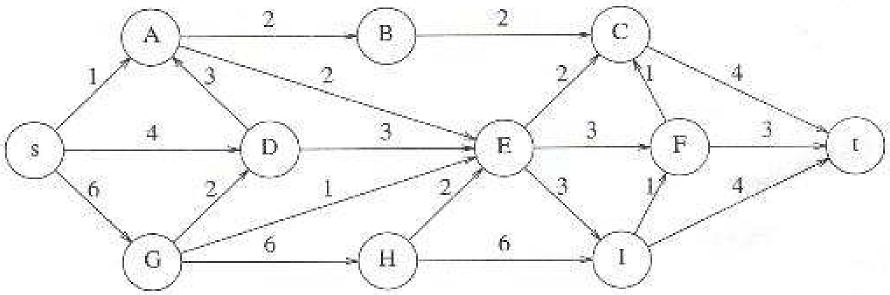
CSE 373 Data Structures WI15 HW06

Problem 1 (5 pts)

Show how to insert the following keys into a B+-tree. The tree is a 2-3 tree, meaning that each internal node has 1-2 keys and up to 3 children. Each leaf node should also have at most 2 keys for this problem.

```
Keys to insert in this order:
82   96   53   46   91
```

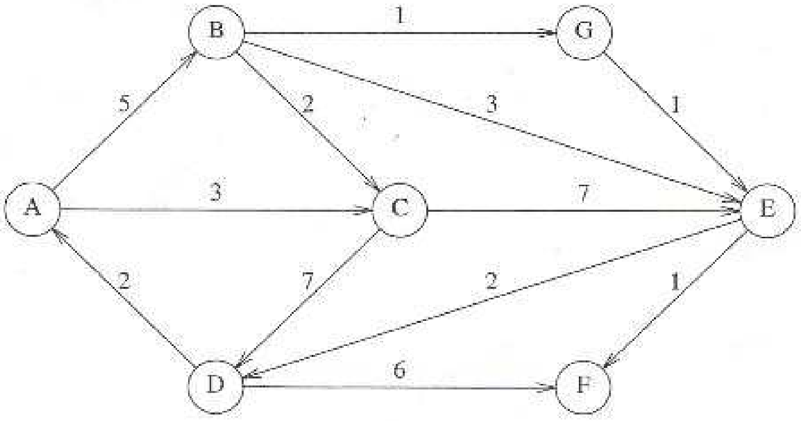
Show all 5 steps in construction of the B+-tree. And note that when a leaf has to split, because it would contain 3 keys, move ONE key to the left leaf and TWO to the right to obey the algorithm I gave in class. If you do it the other way, it will be wrong, even though it is reasonable.



Problem 2 (10 pts)

Find a topological ordering for the graph of Figure 9.81 (reproduced above) using the topological sort algorithm of Slide 27, lecture 17. Show the indegree changes, and the output at each node in the table we provide.

nodes	s	A	B	C	D	E	F	G	H	I	t	Queue Contents	Output
indegree	0	2	1	3	2	4	2	1	1	2	3	s	
Step 1													s
Step 2													
Step 3													
Step 4													
Step 5													
Step 6													
Step 7													
Step 8													
Step 9													
Step 10													
Step 11													



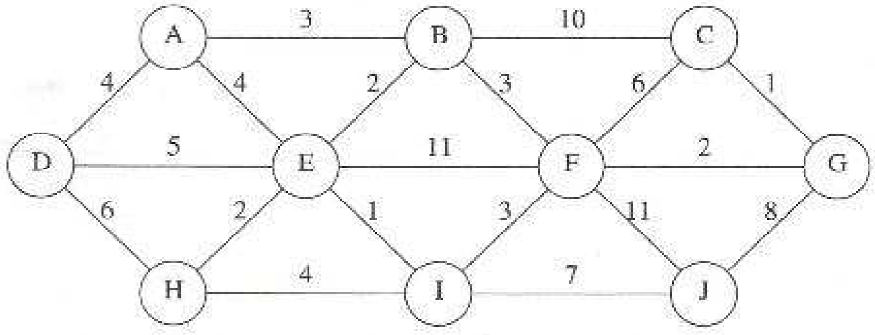
Problem 3 (15 pts)

Find the shortest parth in the weighted digraph of Figure 9.82 (reproduced above) from node A to all other nodes using Dijkstra's algorithm. Do this by **writing code** for the algorithm and running it. Use an an adjacency matrix, since $|V|$ is small. **Just code the brute force method, not the fancy one with the priority queue.** Turn in your code with the assignment.
At each step of the algorithm, output the current distance from A for each vertex. Number the vertices as follows: A-0, B-1, C-2, D-3, E-4, F-5, G-6.

Extra credit: (5 pts) Also keep track of the path to each vertex and at the end print for each vertex both the shortest distance from A to that vertex and the path from A to that vertex. You can do this with one extra field per node in which you keep track of the previous vertex in the shortest path.

Form of output for distances at each step:

Step	0	1	2	3	4	5	6
1	0	9999	9999	9999	9999	9999	9999
2	0						



Problem 4

(10 pts)

Find the minimal spanning tree of the graph of Figure 9.84 (reproduced above) and show each step of your work as little partial graphs with labeled nodes and weighted edges, using

1. Prim's method starting with vertex A

```
      3
1. A-----B
```

2. Kruskal's method starting with edge (E,I)

```
      1
1. E-----I
```

Problem 5 (10 pts)

Construct (draw) a graph from the following relation:

$$(8,6),(8,5),(6,4),(6,10),(6,3),(5,3),(5,1)$$

Then list the nodes as they would be traversed, starting at node 8 in the two kinds of traversals: depth-first (with stack) and breadth-first (with queue). Note: answers will depend partly on which order you take the neighbors of a node.

1. draw graph

2. depth-first

3. breadth-first