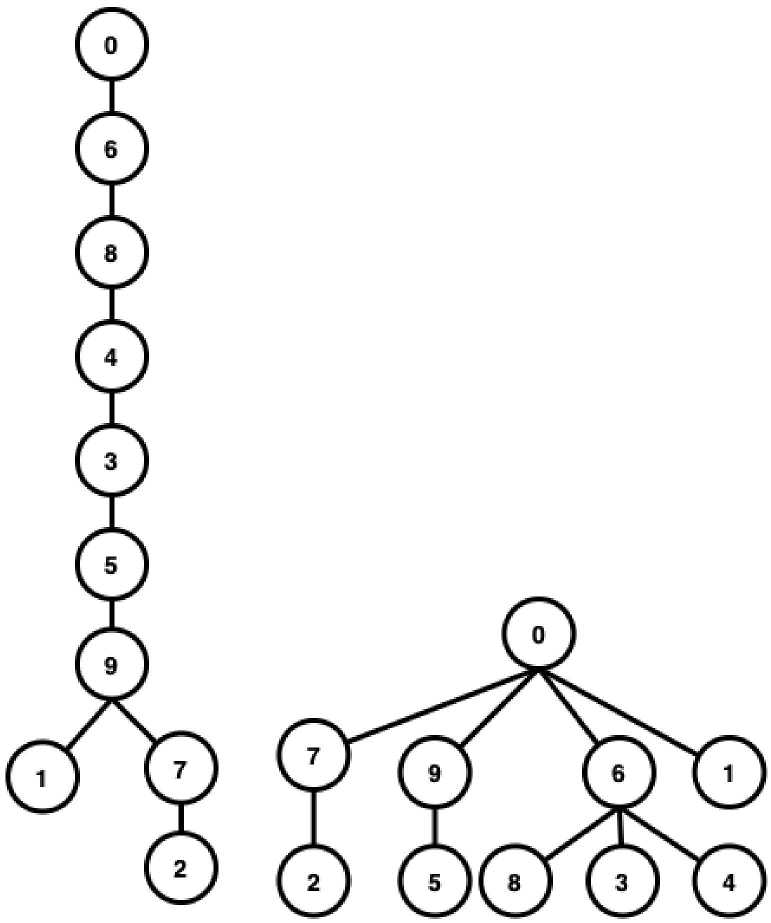
1. (10 Points) Given adjacency list representation of an undirected graph below with 10 vertices from 0 to 9 and source vertex as 0, use algorithms provided in the class slides

a) Draw path tree using Depth First Search
b) Draw path tree using Breadth First Search

```
0: 6 1 7 9
1: 9 3 8 4 0
2: 5 7 4
3: 4 5 6 8 1
4: 8 6 3 2 1
5: 9 3 2
6: 8 4 3 0
7: 9 2 0
8: 6 4 3 1
9: 7 5 1 0
```

Answer:

DFS Search Result	BFS Search Result
index marked edgeTo	index marked edgeTo
0: true 0	0: true 0
1: true 9	1: true 0
2: true 7	2: true 7
3: true 4	3: true 6
4: true 8	4: true 6
5: true 3	5: true 9
6: true 0	6: true 0
7: true 9	7: true 0
8: true 6	8: true 6
9: true 5	9: true 0



DFS Path Tree BFS Path Tree

2. (5 Points) Given adjacency list representation of a digraph below with 10 vertices from 0 to 9, does it have a topological order? If so, provide one. Otherwise, explain why.

0: 4 2 1 3
1: 2
2: 3
3:
4: 2
5: 1 4 3 8 9
6: 3 2 1 9
7: 2 1
8: 2 1 4 6
9: 3 1 4

Answer:
The digraph is a DAG, so it has topological order and here is one:
7, 5, 8, 6, 9, 0, 1, 4, 2, 3,

3. (10 Points) The following is an adjacency list representation of an edge weighted graph, where contents inside each pair of parentheses represent an edge and its associated weight. For instance, (0-5 77) represents edge from 0 to 5 with weight 77. For the given edge weighted graph, provide MST (Minimum Spanning Tree) weight and MST edges in the order as they are added to the MST, using

a) Lazy Prim's MST algorithm starting from vertex 0

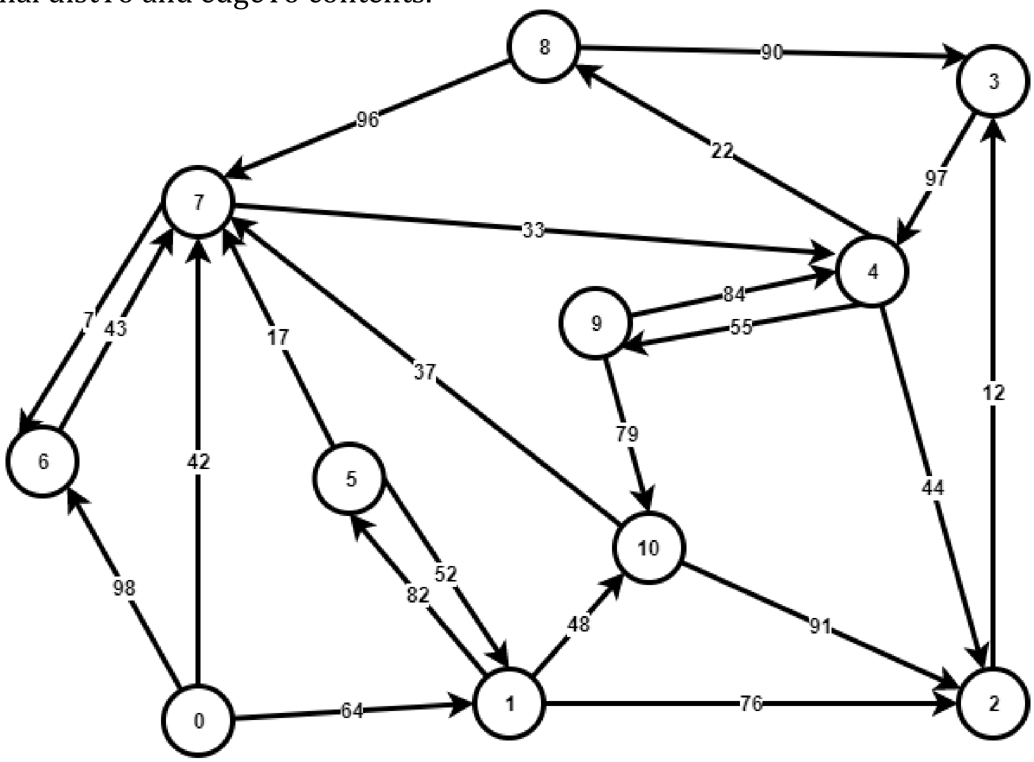
b) Kruskal's MST algorithm

0: (0-5 77.00) (0-2 60.00)
1: (1-5 74.00) (1-3 24.00) (1-2 31.00)
2: (0-2 60.00) (1-2 31.00) (2-8 41.00) (2-9 72.00) (2-5 81.00) (2-6 71.00) (2-7 57.00)
3: (1-3 24.00) (3-5 38.00) (3-10 37.00)
4: (4-8 58.00) (4-7 98.00) (4-5 88.00)
5: (0-5 77.00) (1-5 74.00) (2-5 81.00) (3-5 38.00) (4-5 88.00) (5-6 17.00) (5-8 28.00)
6: (2-6 71.00) (5-6 17.00) (6-9 99.00) (6-10 1.00) (6-7 66.00)
7: (2-7 57.00) (4-7 98.00) (6-7 66.00)
8: (2-8 41.00) (4-8 58.00) (5-8 28.00) (8-10 39.00)
9: (2-9 72.00) (6-9 99.00) (9-10 30.00)
10: (3-10 37.00) (6-10 1.00) (8-10 39.00) (9-10 30.00)

Answer:

Lazy Prim's MST	Kruskal's MST
0-2 60.00 1-2 31.00 1-3 24.00 3-10 37.00 6-10 1.00 5-6 17.00 5-8 28.00 9-10 30.00 2-7 57.00 4-8 58.00	6-10 1.00 5-6 17.00 1-3 24.00 5-8 28.00 9-10 30.00 1-2 31.00 3-10 37.00 2-7 57.00 4-8 58.00 0-2 60.00
MST Weight: 343	MST Weight: 343

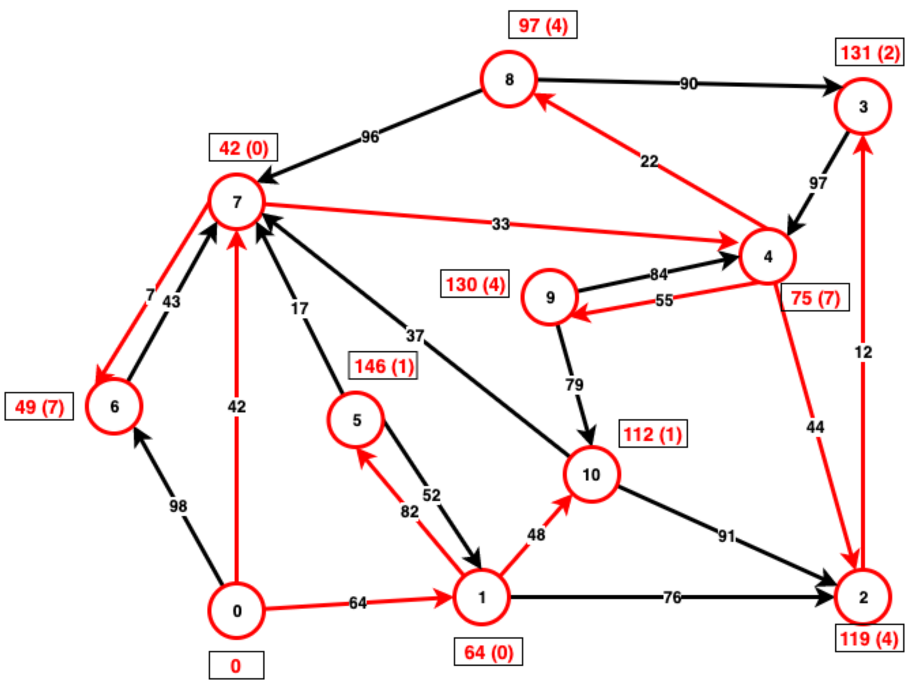
4. (10 Points) Given the following edge weighted digraph and adjacency list representation, find the shortest path tree from source vertex 0 and provide the final distTo and edgeTo contents.

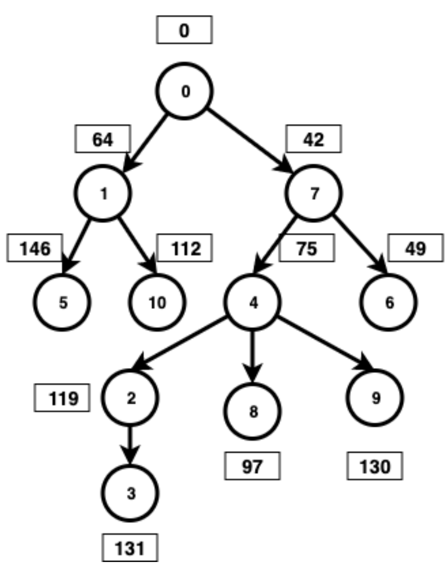


0: (0->6 98.00) (0->7 42.00) (0->1 64.00) 1: (1->5 82.00) (1->10 48.00) (1->2 76.00) 2: (2->3 12.00) 3: (3->4 97.00) 4: (4->8 22.00) (4->9 55.00) (4->2 44.00) 5: (5->1 52.00) (5->7 17.00) 6: (6->7 43.00) 7: (7->4 33.00) (7->6 7.00) 8: (8->7 96.00) (8->3 90.00) 9: (9->10 79.00) (9->4 84.00) 10: (10->7 37.00) (10->2 91.00)

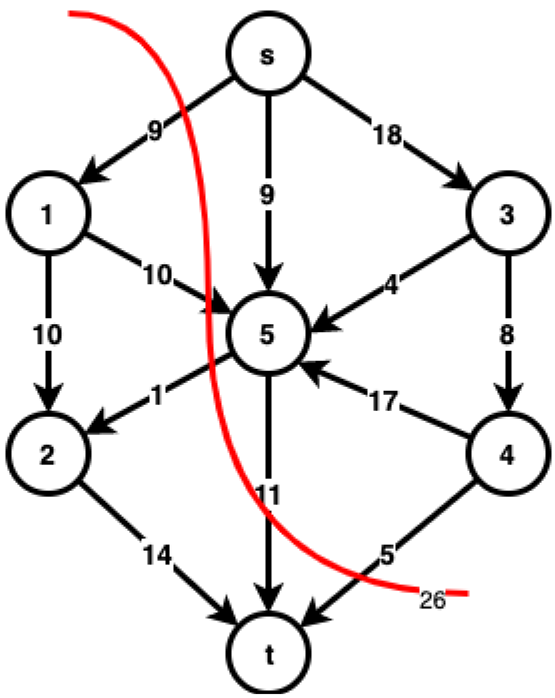
	distTo										
	0	1	2	3	4	5	6	7	8	9	10
Initial	0	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
Relax 0	0	64	INF	INF	INF	INF	98	42	INF	INF	INF
Relax 7		64	INF	INF	75	INF	49	42	INF	INF	INF
Relax 6		64	INF	INF	75	INF	49		INF	INF	INF
Relax 1		64	140	INF	75	146			INF	INF	112
Relax 4			119	INF	75	146			97	130	112
Relax 8			119	187		146			97	130	112
Relax 10			119	187		146				130	112
Relax 2			119	131		146				130	
Relax 9				131		146				130	
Relax 3				131		146					
Relax 5						146					

	edgeTo										
	0	1	2	3	4	5	6	7	8	9	10
Initial	null	null	null	null	null	null	null	null	null	null	null
Relax 0	null	0->1	null	null	null	null	0->6	0->7	null	null	null
Relax 7	null	0->1	null	null	7->4	null	7->6	0->7	null	null	null
Relax 6	null	0->1	null	null	7->4	null	7->6		null	null	null
Relax 1	null	0->1	1->2	null	7->4	1->5			null	null	1->10
Relax 4	null		4->2	null	7->4	1->5			4->8	4->9	1->10
Relax 8	null		4->2	8->3		1->5			4->8	4->9	1->10
Relax 10	null		4->2	8->3		1->5				4->9	1->10
Relax 2	null		4->2	2->3		1->5				4->9	
Relax 9	null			2->3		1->5				4->9	
Relax 3	null			2->3		1->5					
Relax 5	null					1->5					



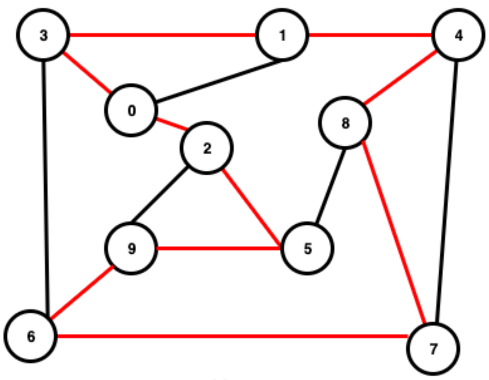


5. (5 Points) Given the network flow digraph below, what is the maximum flow?
Answer:
The maximum flow is 26, because the minimum capacity of any cuts is 26.

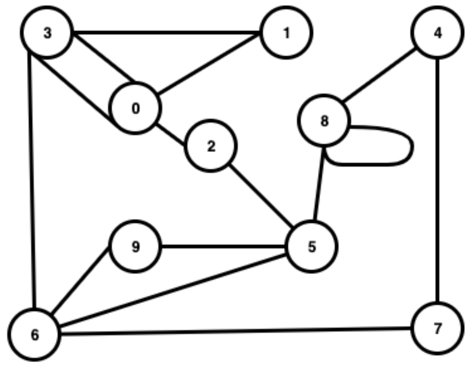


6. (5 Points) Eulerian and Hamiltonian cycles. Consider the graphs defined by the following four sets of edges:
a) 0-1 0-2 0-3 1-3 1-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8
b) 0-1 0-2 0-3 1-3 0-3 2-5 5-6 3-6 4-7 4-8 5-8 5-9 6-7 6-9 8-8
c) 0-1 1-2 1-3 0-3 0-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8
d) 4-1 7-9 6-2 7-3 5-0 0-2 0-8 1-6 3-9 6-3 2-8 1-5 9-8 4-5 4-7
1) Which of these graphs have Euler cycles (cycles that visit each edge exactly once)?
Answer:
b

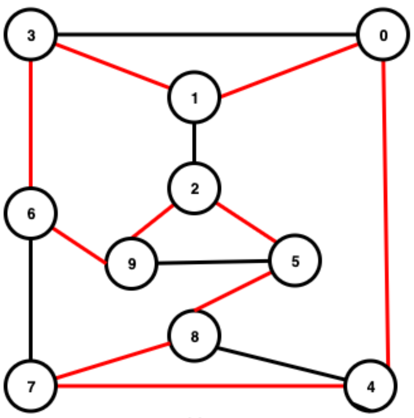
2) Which of them have Hamilton cycles (cycles that visit each vertex exactly once)?
Answer:
a, c, d. Please see the image on the next page for Hamilton cycles.



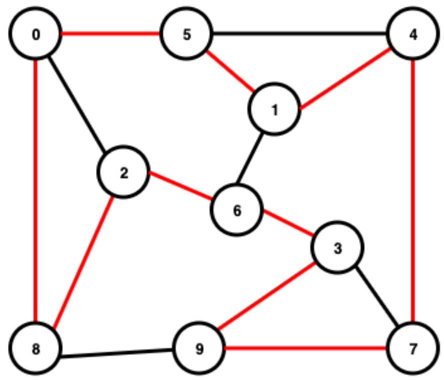
(a)



(b)



(c)

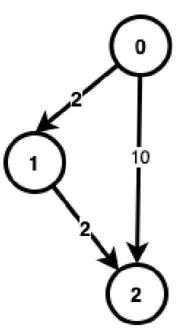


(d)

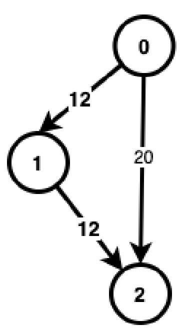
7. (5 Points) If the following statement true? If so, explain why, if not provide a counter example.
Adding a constant to every edge weight does not change the solution to the single-source shortest-paths problem.

Answer:

The statement is false. For the original graph below, the shortest path from 0 to 2 is 0->1->2. After adding 10 to each edge weight, the shortest path from 0 to 2 is 0->2, which is different from the original solution.



Original Graph



After adding 10 to each edge weight

8. (10 Points) Indicate true or false for each of the following statements regarding complexity classes.

a) Finding shortest unweighted paths in graph is a P problem.
b) There are some problems in P do not belong to NP.
c) If Traveling salesman problem can be solved in polynomial time using deterministic Turing machine, then every problem in NP can be solved in polynomial time using deterministic Turing machine.
d) NP stands for Non-Polynomial.
e) There are some problems cannot be solved by a Turing machine.

Answer:
a: true, b: false, c: true, d: false, e: true

9. (10 Points) For each of the algorithms listed in the table, indicate which of the following algorithm design techniques is used: Greedy algorithm, divide and conquer, dynamic programming, randomized algorithms, and backtracking.

Algorithms	**Algorithm Design Technique Used**
Merge Sort	Divide and conquer
Quick sort that randomly picks pivot element	Divide and conquer or Randomized algorithm
Dijkstra's SPT	Greedy algorithm
Bellman-Ford SPT	Dynamic programming
Tic-tac-toe game using minimax strategy with α-β pruning	Backtracking

10. (Extra Credit 10 Points) The eccentricity of a vertex v is the length of the shortest path from that vertex to the furthest vertex from v. Implement the methods in the following class (extra helper methods can be added). After the constructor is called, a client should be able to call getEccentricity() method to get the result.

Answer:

One solution is to modify BFS find path algorithm by adding the following:

```
private int eccentricity;

@SuppressWarnings("unused")
public Eccentricity(MatrixGraph graph, int vertex) {
    this.s = vertex;

    int numVertices = graph.V();
    marked = new boolean[numVertices];
    edgeTo = new int[numVertices];
    int maxLength = 0;

    bfs(graph, vertex);
    for (int v=0; v<numVertices; v++) {
        if (v!=vertex && marked[v]) {
            int count = -1;
            Iterable<Integer> it = pathTo(v);
            for (Integer n: it) {
                count++;
            }

            if (count > maxLength) {
                maxLength = count;
            }
        }
    }

    eccentricity = maxLength;
}

public int getEccentricity() {
    return eccentricity;
}
```

Submission Note

1) For written part of the questions:
   a) Write your answers inside a text document (in plain text, MS Word, or PDF format)
   b) Name the file as csc130.firstname.lastname.assignment5.txt(doc, docx, or pdf) with proper file extension
3) Submit both of your text document via Canvas course web site.
4) Due Dec 3, 11:59 PM