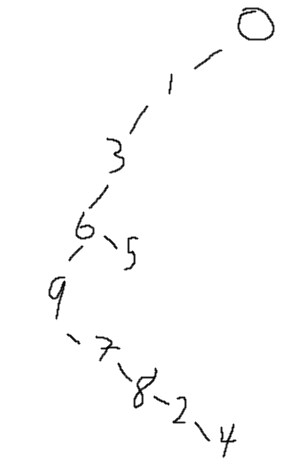
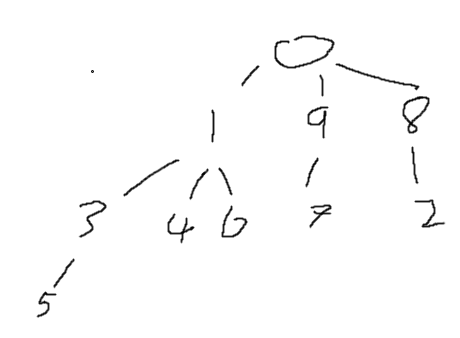
1. (10 Points) Given adjacency list representation of an undirected graph below and source vertex as 0, use algorithms provided in the class slides

1. Draw path tree using Depth First Search



1. Draw path tree using Breadth First Search



0: 1 9 8

1: 3 4 6 0

2: 4 8 6

3: 6 5 9 1

4: 7 9 6 2 1

5: 6 3

6: 9 7 5 4 3 2 1

7: 9 8 6 4

8: 7 2 0

9: 7 6 4 3 0

2. (5 Points) Given adjacency list representation of a digraph below, does it have cycles? If so, provide a cycle.

0: 1 2 3 4 8

1: 2 3 6 8

2: 3 4 8

3: 1 6 8 9

4: 5 7 9

5: 4 6 9

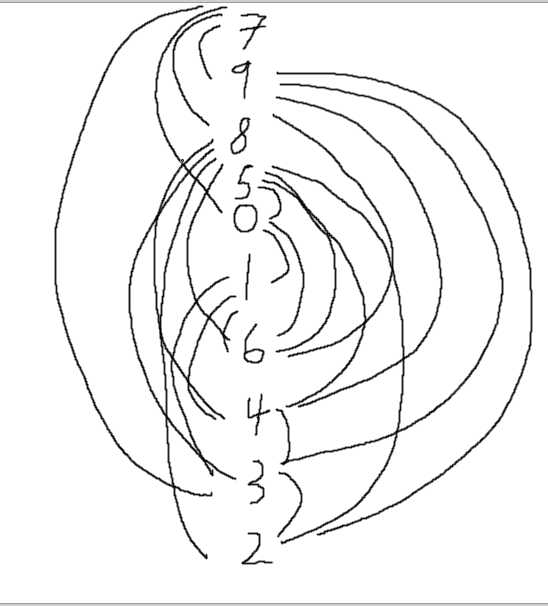
6: 1 5 8

7: 8 9

8: 0 9

9: 3 6 7

Yes. 3🡪 1 🡪 2 🡪 3 🡪 1, etc.

3. (5 Points) Given adjacency list representation of a digraph below, does it have a topological order? If so, provide one. Otherwise, explain why.  Yes

0: 6 1

1: 3 4 6

2:

3: 2

4: 3

5: 6 2 4 0

6: 2 4

7: 3 0 9 8

8: 2 4 3 6

9: 4 3 6 2

4. (10 Points) Given the following edge weighted graph, provide MST edges in the order as they are added to the MST and MST weight, using

1. Lazy Prim’s MST algorithm starting from vertex 0

0-6(0.09), 6-10(0.20), 10-3(0.08), 10-7(0.12), 7-4(0.04),

4-8(0.13), 8-5(0.14), 10-2(0.28), 8-9(0.33), 10-1(0.86)

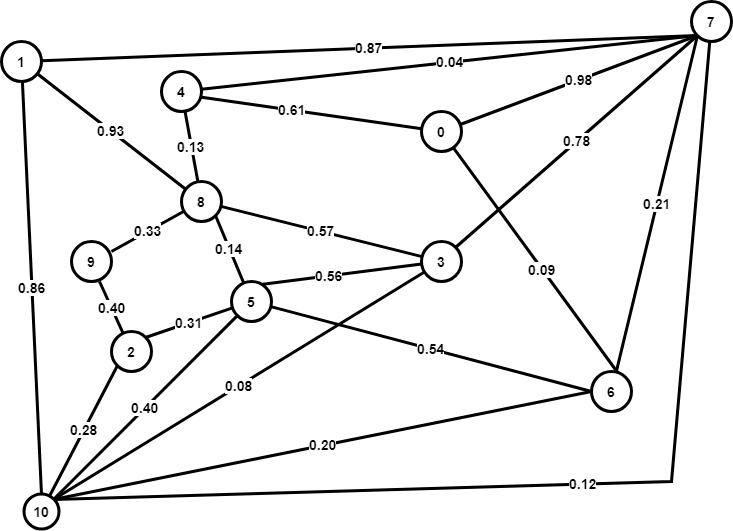
MST Total: 2.27

1. Kruskal’s MST algorithm

4-7(0.04), 3-10(0.08), 0-6(0.09), 10-7(0.12), 4-8(0.13),

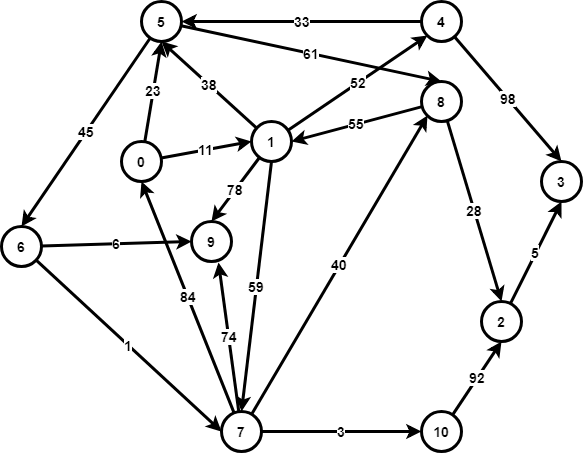
8-5(0.14), 6-10(0.20), 10-2(0.28), 8-9(0.33), 10-1(0.86)

MST Total: 2.27



5. (10 Points) Given the following edge weighted digraph, find the shortest path tree from source vertex 0, including the trace of distTo and edgeTo contents similar to slide#18 from module 11.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | distTo | | | | | | | | | | | edgeTo | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 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~~ | 11 | INF | INF | INF | 23 | INF | INF | INF | INF | INF |  | 0🡪1 |  |  |  | 0🡪5 |  |  |  |  |  |
| Relax 5 | ~~0~~ | 11 | INF | INF | INF | ~~23~~ | 68 | INF | 84 | INF | INF |  | 0🡪1 |  |  |  | 0🡪5 | 5🡪6 |  | 5🡪8 |  |  |
| Relax 6 | ~~0~~ | 11 | INF | INF | INF | ~~23~~ | ~~68~~ | 69 | 84 | 74 | INF |  | 0🡪1 |  |  |  | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 |  |
| Relax 9 | ~~0~~ | 11 | INF | INF | INF | ~~23~~ | ~~68~~ | 69 | 84 | ~~74~~ | INF |  | 0🡪1 |  |  |  | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 |  |
| Relax 7 | ~~0~~ | 11 | INF | INF | INF | ~~23~~ | ~~68~~ | ~~69~~ | 84 | ~~74~~ | 72 |  | 0🡪1 |  |  |  | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 1 | ~~0~~ | ~~11~~ | INF | INF | 63 | ~~23~~ | ~~68~~ | ~~69~~ | 84 | ~~74~~ | 72 |  | 0🡪1 |  |  | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 8 | ~~0~~ | ~~11~~ | 112 | INF | 63 | ~~23~~ | ~~68~~ | ~~69~~ | ~~84~~ | ~~74~~ | 72 |  | 0🡪1 | 8🡪2 |  | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 10 | ~~0~~ | ~~11~~ | 112 | INF | 63 | ~~23~~ | ~~68~~ | ~~69~~ | ~~84~~ | ~~74~~ | ~~72~~ |  | 0🡪1 | 8🡪2 |  | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 2 | ~~0~~ | ~~11~~ | ~~112~~ | 117 | 63 | ~~23~~ | ~~68~~ | ~~69~~ | ~~84~~ | ~~74~~ | ~~72~~ |  | 0🡪1 | 8🡪2 | 2🡪3 | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 4 | ~~0~~ | ~~11~~ | ~~112~~ | 117 | ~~63~~ | ~~23~~ | ~~68~~ | ~~69~~ | ~~84~~ | ~~74~~ | ~~72~~ |  | 0🡪1 | 8🡪2 | 2🡪3 | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |
| Relax 3 | ~~0~~ | ~~11~~ | ~~112~~ | ~~117~~ | ~~63~~ | ~~23~~ | ~~68~~ | ~~69~~ | ~~84~~ | ~~74~~ | ~~72~~ |  | 0🡪1 | 8🡪2 | 2🡪3 | 1🡪4 | 0🡪5 | 5🡪6 | 6🡪7 | 5🡪8 | 6🡪9 | 7🡪10 |



6. (5 Points) To find shortest paths for an edge weighted graph with negative edge weights, 1) find the minimum edge weight, 2) add the absolute value of the minimum edge weight to every edge, 3) run Dijkstra’s SPT algorithm to find the shortest path from the modified graph.

Would the above algorithm work for? If so, explain why. If not, provide a counter example.

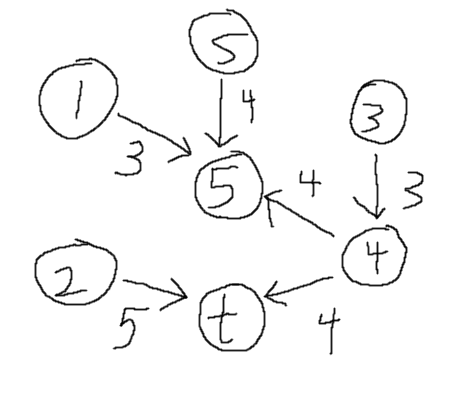
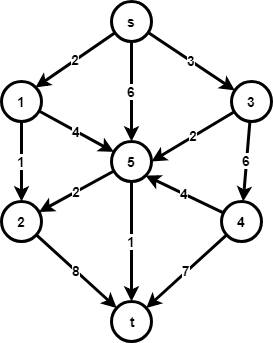
No it wouldn’t. Say you have two edges valued at 0 and one edge valued at 3. you have a minimum edge weight of -45. Before running this algorithm you would see the 0, 0 path as the lightest path. Afterwards, taking a path of weight 48 is certainly lighter than a path of 45+45=90. Basically, the added weight of the minimum edge is doubly potent when considering multiple paths instead of directly comparing individual paths.

7. (5 Points) Given the network flow digraph below,

1. What is the maximum flow?

7

1. Provide the flow graph that shows the maximum flow paths.



8. (5 Points) Indicate true or false for each of the following statements regarding complexity classes.

1. There are some problems in P but not in NP
2. There is no overlap between NP and undecidable problems
3. NP-Complete problems are harder than P problems
4. It has been proven that P!=NP
5. NP stands for Non-Polynomial

9. (5 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.

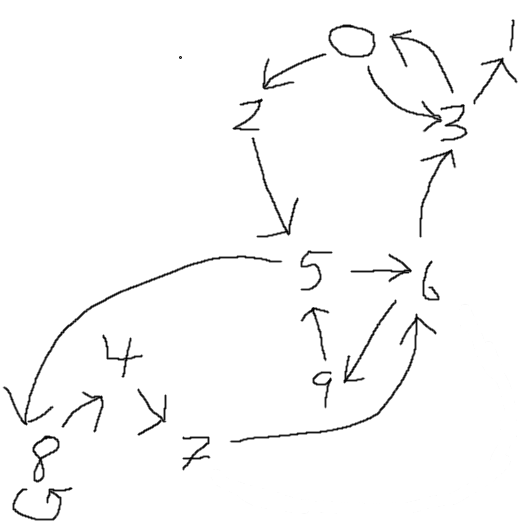
|  |  |
| --- | --- |
| **Algorithms** | **Algorithm Design Technique Used** |
| Merge Sort | Divide and Conquer Algorithm |
| Prim’s MST | Greedy Algorithm |
| Bellman-Ford SPT | Dynamic Programming |
| Skip Lists | Randomized Algorithms |
| Tic-tac-toe game using minimax strategy with α-β pruning | Backtracking |

**Extra Credit Questions**

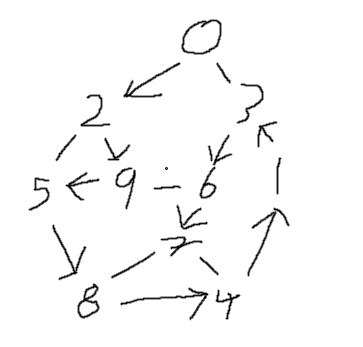
10. (4 Points) Consider the graphs defined by the following

four sets of edges:

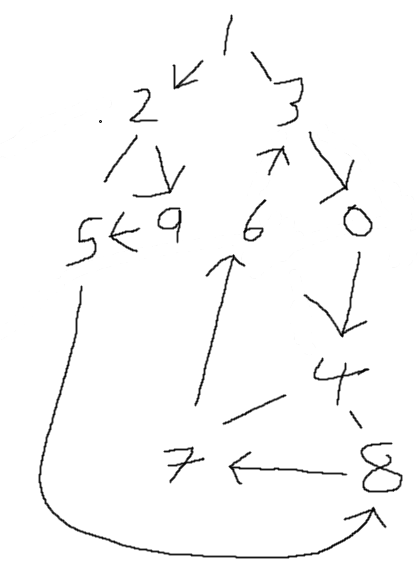
1. 0-2 0-3 1-3 0-3 2-5 5-6 3-6 4-7 4-8 5-8 5-9 6-7 6-9 8-8

Euler 

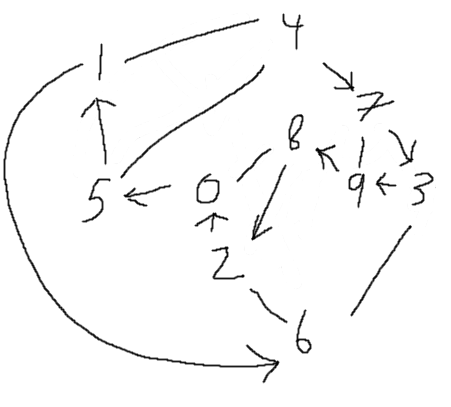
1. 0-2 0-3 1-3 1-4 2-5 2-9 3-6 4-7 4-8 5-8 5-9 6-7 6-9 7-8

Hamilton 

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

Hamilton 

1. 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

Hamilton 

Which of these graphs have Euler cycles (cycles that visit each edge exactly once)?

Which of them have Hamilton cycles (cycles that visit each vertex exactly once except the starting and ending vertex)?

11. (3 Points) Give a counterexample that shows why the following strategy does not necessarily find the MST: ‘Start with any vertex as a single-vertex MST, then add V-1 edges to it, always taking next a min-weight edge incident to the vertex most recently added to the MST.’

You could end up in a closed loop of min weighted edges, which means you would never complete the MST. You could also end up taking a path with a higher total weight than optimal because you path exclusively down min-weight edges rather than checking available edge weights at the next vertex.

12. (3 Points) Suppose that we convert an EdgeWeightedGraph into an EdgeWeightedDigraph by creating two DirectedEdge objects in the EdgeWeightedDigraph (one in each direction) for each Edge in the EdgeWeightedGraph with the same edge weight for each direction and then use the Bellman-Ford algorithm to find a SPT. Explain why this approach fails.

If you have a negative edge weight value, it frequently makes sense to backtrack on that doubled up negative edge in order to reduce the total weight of the digraph, which would not be possible on an edge weighted graph. You could very likely end up with a new "lowest weight path" if you have any negative edge weights.

**Submission Note**

1) For written part of the questions:

1. Write your answers inside a text document (in plain text, MS Word, or PDF format)
2. Name the file as 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 7, 11:59 PM