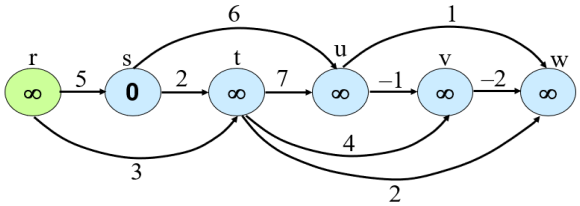
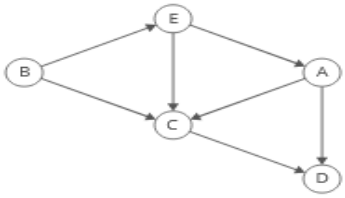
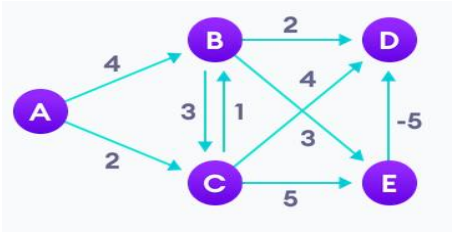
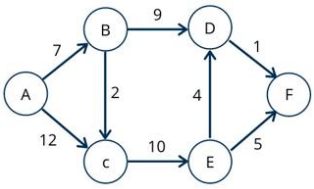
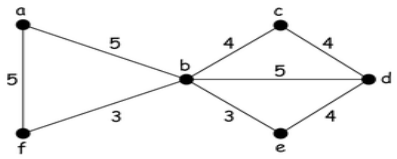
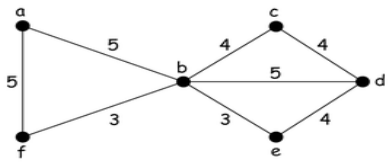
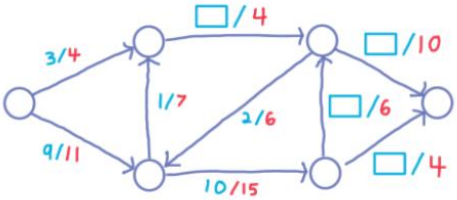
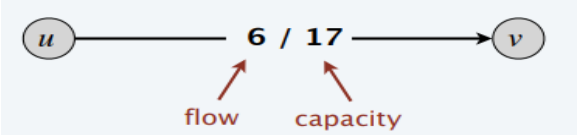
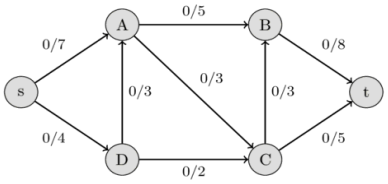
USN

**Department of Computer Science and Engineering**
**M.Tech in Computer Science and Engineering (CSE)**
**Continuous Internal Evaluation (CIE-II) Question Paper**

| | | | |
|---|---|---|---|
| Course: | Advanced Data Structures and Algorithms | Course Code: 22MCE12TL | Semester: 01 |
| 26.04.2023 | Duration : 90 minutes | Max Marks: 50 | Staff: RS |

| Sl. No. | Answer all questions | M,*L1-L6,CO |
|---|---|---|
| 1a. | Solve by applying the Shortest Path algorithm for Directed Acyclic Graph given below. Compute the time complexity for the same.   | 6,L3, CO1 |
| 1b. | Find the Topological Sort of the given graph. Calculate the Time and Space Complexity for the algorithm.   | 4,L3, CO1 |
| 2a. | Solve using Bellman-Ford Algorithm. Calculate the time complexity for the algorithm.   | 6,L4, CO2 |
| 2b. | Solve using Dijkstra's Algorithm. Calculate the time complexity for the algorithm.   | 4,L4, CO2 |

| | | |
|---|---|---|
| 3a. | Solve the below given problem by applying Prim's Algorithm to find the Minimum Cost Spanning Tree. Calculate the time complexity for the algorithm.   | 6,L4, CO4 |
| 3b | Compare the working of Bellman Ford algorithm with Dijkstra's algorithm. | 4,L2, CO2 |
| 4a. | Solve the below given problem by applying Kruskal's Algorithm to find the Minimum Spanning Tree. Calculate the time complexity for the algorithm.   | 6,L4, CO4 |
| 4b. | Fill up the flow values in the graph given below:   | 4,L3, CO3 |
| 5a. | For the Graph given below, find the Residual Graph   | 2,L2, CO1 |
| 5b. | Calculate the max flow for the following network   | 8, L5, CO3 |

****Course Outcome and Marks Distribution *(L1-L6)**

CO1: Analyze the efficiency of programs based on time complexity.

CO2: Critically think and apply appropriate design paradigm and algorithm for a specific problem.

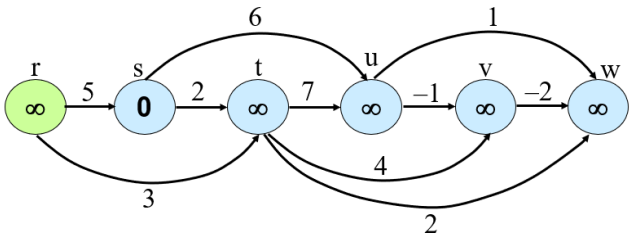
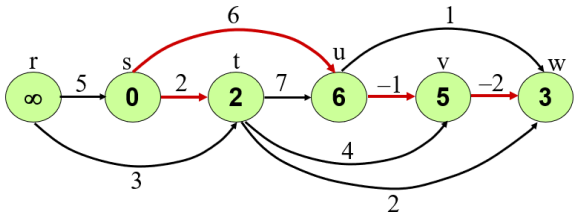
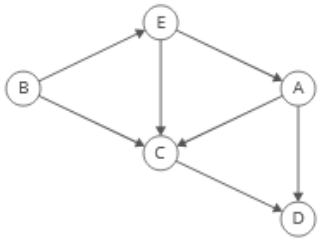
CO3: Apply knowledge of computing and mathematics to algorithm design.

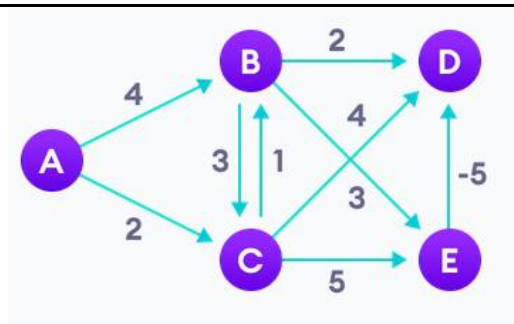
CO4: Design, implement and evaluate algorithms to solve real world problems.

| L1 | L2 | L3 | L4 | L5 | L6 | CO1 | CO2 | CO3 | CO4 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 10 | 26 | 0 | 0 | 12 | 14 | 12 | 12 |

**Department of Computer Science and Engineering**
**M.Tech in Computer Science and Engineering (CSE)**
**Continuous Internal Evaluation (CIE-II)- Scheme and Solution**

| | | | |
|---|---|---|---|
| Course: | Advanced Data Structures and Algorithms | Course Code: 22MCE12TL | Semester: 01 |
| 26.04.2023 | Duration : 90 minutes | Max Marks: 50 | Staff: RS |

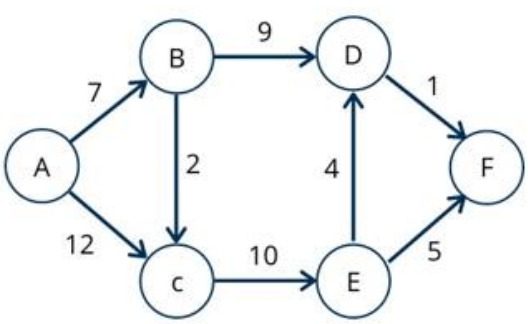
| | | |
|---|---|---|
| 1a. | Solve by applying the Shortest Path algorithm for Directed Acyclic Graph given below. Compute the time complexity for the same.<br>  <br><br>  <br><br>Time Complexity: Time Complexity: O (V + E) | 5+1 |
| 1b. | Find the Topological Sort of the given graph. Calculate the Time and Space Complexity for the algorithm.<br>  <br><br>Topological Ordering:<br>B E A C **D**<br>Time Complexity: $O(V+E)$<br>Space Complexity: O(V) | 3+0.5+0.5 |
| 2a. | Solve using Bellman-Ford Algorithm. Calculate the time complexity for the algorithm. | 5+1 |

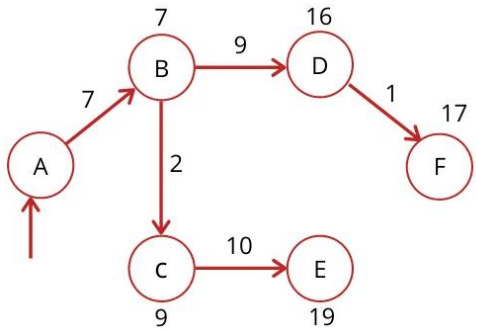



Time Complexity : O(V*E)

| | B | C | D | E |
|---|---|---|---|---|
| 0 | ∞ | ∞ | ∞ | ∞ |
| 0 | 4 | 2 | ∞ | ∞ |
| 0 | 3 | 2 | 6 | 6 |
| 0 | 3 | 2 | 1 | 6 |
| 0 | 3 | 2 | 1 | 6 |

2b. Solve using Dijkstra's Algorithm. Calculate the time complexity for the algorithm. 3+1

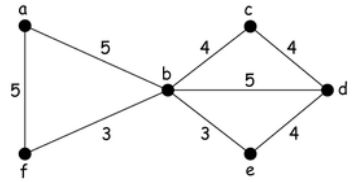


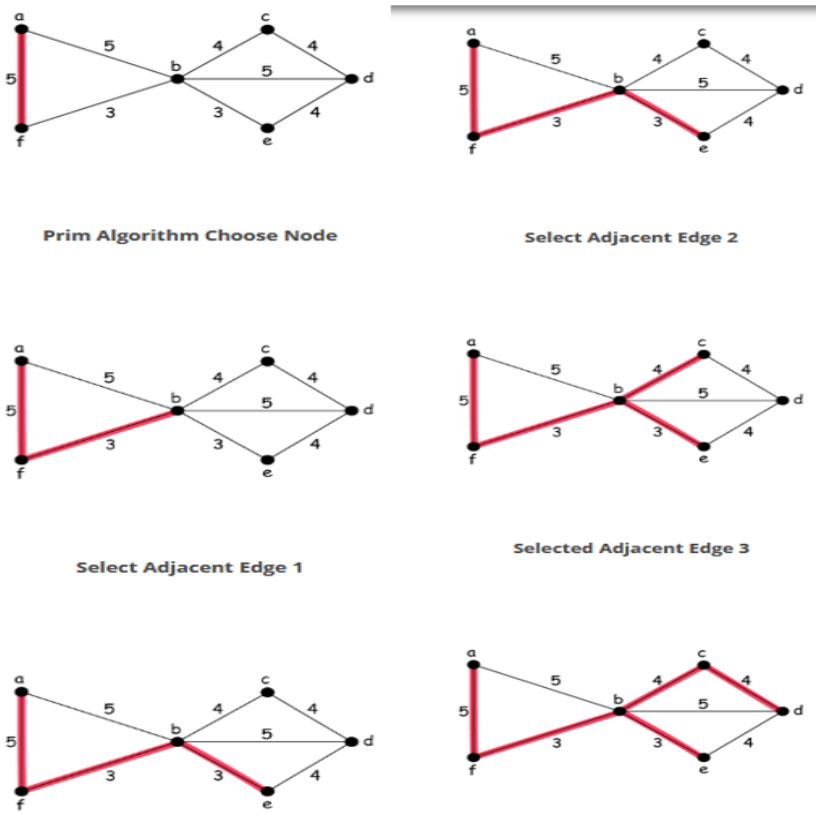




Time Complexity: O(|V|^2 + |E|)

For Sparse Graphs : $O((|E|+|V|) \log |V|)$

3a. Solve the below given problem by applying Prim's Algorithm to find the Minimum Spanning Tree. Calculate the time complexity for the algorithm. 5+1



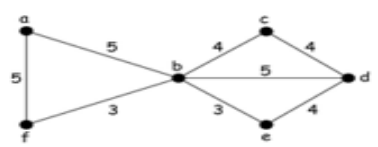




Prim's algorithm starts with a single node and grows a spanning tree by adding edges until every vertex is selected and the minimum spanning tree is built with a Minimal Total Cost = 5 + 3 + 3 + 4 + 4 = 19.

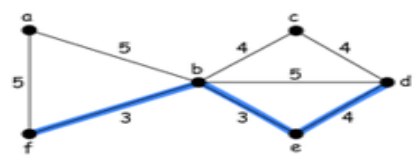
| | | |
|---|---|---|
| 3b | **Bellman Ford's algorithm** and **Dijkstra's algorithm** both are single-source shortest path algorithm, i.e. both determines the shortest distance of each vertex of a graph from a single source vertex. However, there are some key differences between them. We follow the Dynamic Programming approach in Bellman Ford's algorithm and Greedy approach in Dijkstra's algorithm. | 4x1=4 |

| Bellman Ford's Algorithm | Dijkstra's Algorithm |
|---|---|
| Bellman Ford's Algorithm works when there is negative weight edge, it also detects the negative weight cycle. | Dijkstra's Algorithm may or may not work when there is negative weight edge. But will definitely not work when there is a negative weight cycle. |
| The result contains the vertices which contains the information about the other vertices they are connected to. | The result contains the vertices containing whole information about the network, not only the vertices they are connected to. |
| It can easily be implemented in a distributed way. | It can not be implemented easily in a distributed way. |
| It is more time consuming than Dijkstra's algorithm. Its time complexity is O(VE). | It is less time consuming. The time complexity is O(E logV). |
| Dynamic Programming approach is taken to implement the algorithm. | Greedy approach is taken to implement the algorithm. |
| Bellman Ford's Algorithm have more overheads than Dijkstra's Algorithm. | Dijkstra's Algorithm have less overheads than Bellman Ford's Algorithm. |
| Bellman Ford's Algorithm have less scalability than Dijkstra's Algorithm. | Dijkstra's Algorithm have more scalability than Bellman Ford's Algorithm. |

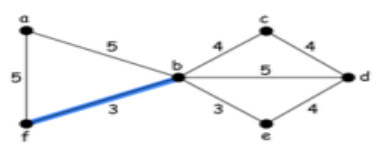
**4a.** Solve the below given problem by applying Kruskal's Algorithm to find the Minimum Spanning Tree. Calculate the time complexity for the algorithm. **6**



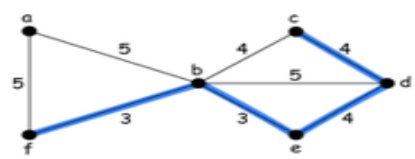

Undirected Graph Kruskal Algorithm



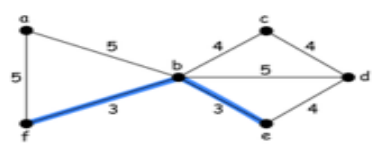

Min Weight Kruskal 3



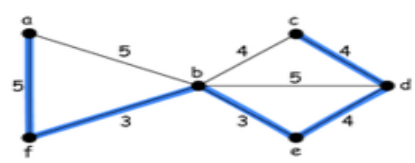

Min Weight Kruskal 1




Min Weight Kruskal 4

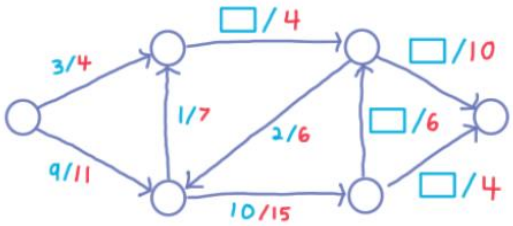
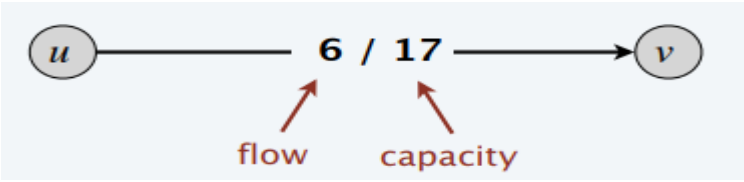
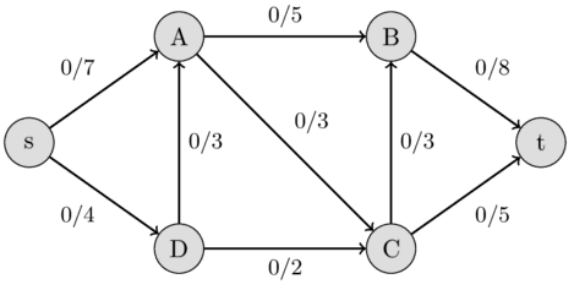



Min Weight Kruskal 2




Kruskal's algorithm keeps adding edges of minimum weight until a minimum spanning tree is created — thus, giving us a Minimal Total Cost of 3 + 3 + 4 + 4 + 5 = 19

**4b.** Fill up the flow values in the graph given below: **4**

| | | |
|---|---|---|
| |  <br>4/4, 6/6, 4/4, 8/10 | |
| 5. | (a) For the Graph given below, find the Residual Graph<br>  <br>11 in forward direction and 6 in reverse direction<br>(b) Calculate the max flow for the following network:<br>  <br>max flow = 10 | 2+8 |

****Course Outcome**

| |
|---|
| CO1: Analyze the efficiency of programs based on time complexity. |
| CO2: Critically think and apply appropriate design paradigm and algorithm for a specific problem. |
| CO3: Apply knowledge of computing and mathematics to algorithm design |
| CO4: Design, implement and evaluate algorithms to solve real world problems |

**Marks Distribution *(L1-L6)**

| L1 | L2 | L3 | L4 | L5 | L6 | CO1 | CO2 | CO3 | CO4 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 8 | 18 | 24 | 0 | 0 | 10 | 20 | 10 | 10 |