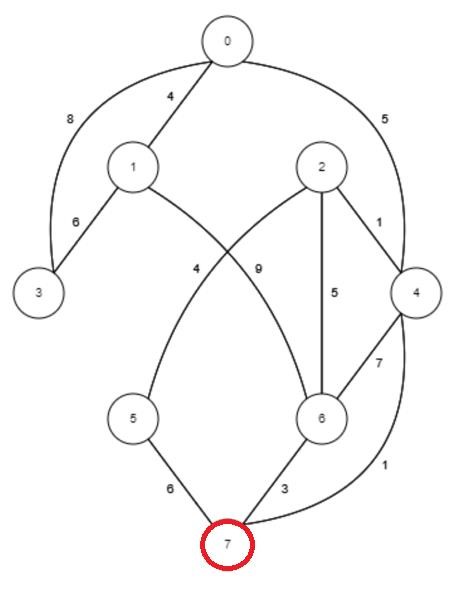
**COSC 3304 – Algorithms Design and Analysis**

**Assignment 11**

**Due: 11:59pm, 04/30/2024**

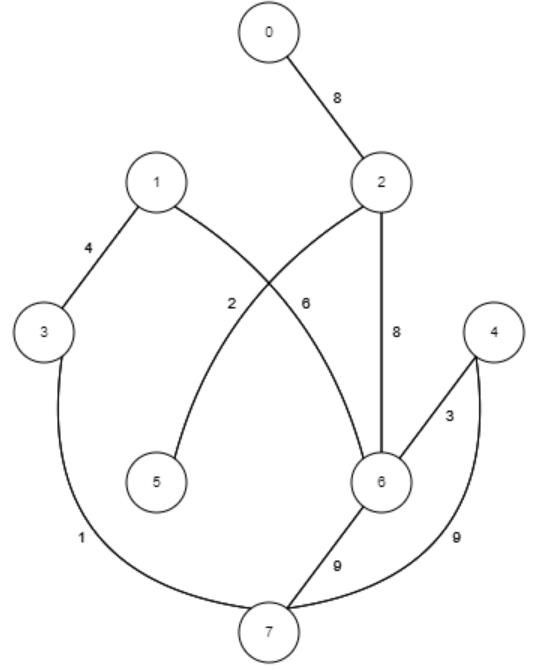
**Andrew Kalathra**

1. Please apply Prim’s method to find the Minimum Spanning Tree of the graph below starting from the **vertex 7**. (15 points)



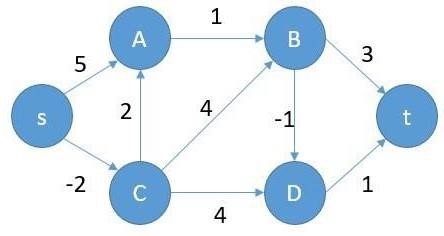
1. 7 🡪 4 (1)
2. 4 🡪 2 (1)
3. 7 🡪 6 (3)
4. 2 🡪 5 (4)
5. 4 🡪 0 (5)
6. 0 🡪 1 (4)
7. 1 🡪 3 (6)

1. Please apply Kruskal method to find the Minimum Spanning Tree of the graph below. (15 points)



1. List of edges in order: 1, 2, 3, 4, 6, 8, 8, 9, 9
2. 1: 3 🡪 7
3. 2: 2 🡪 5
4. 3: 4 🡪 6
5. 4: 3 🡪 1
6. 6: 1 🡪 6
7. 8: 2 🡪 6
8. 8: 0 🡪 2
9. 9: X, loop
10. 9: X, loop

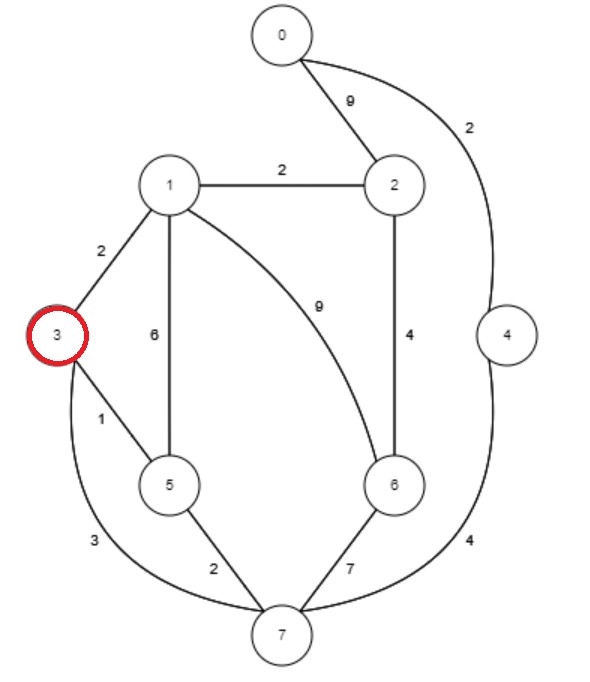
1. Please show the **Bellman-Ford Algorithm** to find the shortest parths from vertex s to other vertices. Assume the each pass relax’s the edges in the order of **sA, AB, Bt, CA, CB, BD, sC, CD,** and **Dt**. Please use the table below to show how the distance to each vertex is updated (You can use as many rows as you need). (20 Points)



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| s | A | B | C | D | t |
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 0 | 5 | 6 | -2 | 2 | 3 |
| 0 | 0 | 2 | -2 | 1 | 2 |
| 0 | 0 | 1 | -2 | 0 | 1 |
| 0 | 0 | 1 | -2 | 0 | 1 |
| 0 | 0 | 1 | -2 | 0 | 1 |

5 loops because 6 vertices, so 6-1 = 5.

1. Please use the **Dijkstra Algorithm** to find the shortest paths starting from vertex 0 to other vertices. You need to show the cost and vertices on each shortest path. (20 points)



1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | ∞ | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| Parent |  |  |  | -1 |  |  |  |  |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | ∞ | 2 | ∞ | 0 | ∞ | 1 | ∞ | 3 |
| Parent |  | 3 |  | -1 |  | 3 |  | 3 |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | ∞ | 2 | ∞ | 0 | ∞ | 1 | ∞ | 3 |
| Parent |  | 3 |  | -1 |  | 3 |  | 3 |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | ∞ | 2 | 4 | 0 | ∞ | 1 | 11 | 3 |
| Parent |  | 3 | 1 | -1 |  | 3 | 1 | 3 |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | ∞ | 2 | 4 | 0 | 7 | 1 | 10 | 3 |
| Parent |  | 3 | 1 | -1 | 7 | 3 | 7 | 3 |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | 13 | 2 | 4 | 0 | 7 | 1 | 8 | 3 |
| Parent | 2 | 3 | 1 | -1 | 7 | 3 | 2 | 3 |

1. :

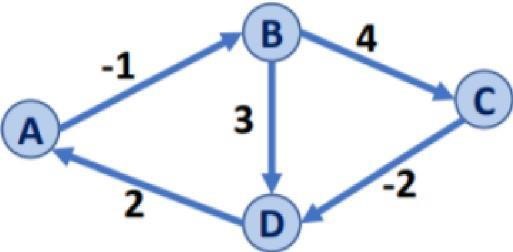
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | 9 | 2 | 4 | 0 | 7 | 1 | 8 | 3 |
| Parent | 4 | 3 | 1 | -1 | 7 | 3 | 2 | 3 |

1. :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Cost | 9 | 2 | 4 | 0 | 7 | 1 | 8 | 3 |
| Parent | 4 | 3 | 1 | -1 | 7 | 3 | 2 | 3 |

1. Please use **Floyd-Warshall** method to compute the all pairs shortest paths of the graph below. Please show the **Distance Matrix** after every iteration D(0) to D(4) (30 points)

For D1, I will assume that is A, D2 as B, and so on so forth.



1. D0:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | 0 | -1 | ∞ | ∞ |
| **B** | ∞ | 0 | 4 | 3 |
| **C** | ∞ | ∞ | 0 | -2 |
| **D** | 2 | ∞ | ∞ | 0 |

P0:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | NIL | A | NIL | NIL |
| **B** | NIL | NIL | B | B |
| **C** | NIL | NIL | NIL | C |
| **D** | D | NIL | NIL | NIL |

1. D1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | 0 | -1 | ∞ | ∞ |
| **B** | ∞ | 0 | 4 | 3 |
| **C** | ∞ | ∞ | 0 | -2 |
| **D** | 2 | 1 | ∞ | 0 |

P1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | NIL | A | NIL | NIL |
| **B** | NIL | NIL | B | B |
| **C** | NIL | NIL | NIL | C |
| **D** | D | A | NIL | NIL |

1. D2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | 0 | -1 | 3 | 2 |
| **B** | ∞ | 0 | 4 | 3 |
| **C** | ∞ | ∞ | 0 | -2 |
| **D** | 2 | 1 | 5 | 0 |

P2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | NIL | A | B | B |
| **B** | NIL | NIL | B | B |
| **C** | NIL | NIL | NIL | C |
| **D** | D | A | B | NIL |

1. D3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | 0 | -1 | 3 | 1 |
| **B** | ∞ | 0 | 4 | 2 |
| **C** | ∞ | ∞ | 0 | -2 |
| **D** | 2 | 1 | 5 | 0 |

P3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | NIL | A | B | C |
| **B** | NIL | NIL | B | C |
| **C** | NIL | NIL | NIL | C |
| **D** | D | A | B | NIL |

1. D4:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | 0 | -1 | 3 | 1 |
| **B** | 4 | 0 | 4 | 2 |
| **C** | 0 | -1 | 0 | -2 |
| **D** | 2 | 1 | 5 | 0 |

P3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** |
| **A** | NIL | A | B | C |
| **B** | D | NIL | B | C |
| **C** | D | D | NIL | C |
| **D** | D | A | B | NIL |