Premier University
Department of Computer Science & Engineering

4th Semester Special Retake Exam, 2020

Course Title: Algorithms Design & Analysis

Course Code: CSE - 225

Total Marks: 35 **Time: 2 Hours**

Answer each question.

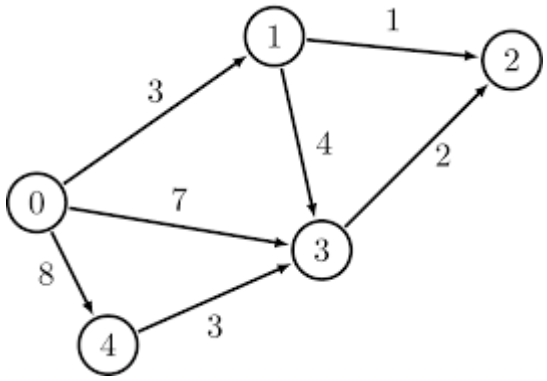
<table>
<tr><td>Q.1</td><td>a.</td><td>Find an optimal solution to the fractional knapsack problem, where the knapsack capacity is M = 10.
<table>
<tr><th>Item</th><th>Weight</th><th>Profit</th></tr>
<tr><td>1</td><td>3</td><td>12</td></tr>
<tr><td>2</td><td>4</td><td>8</td></tr>
<tr><td>3</td><td>6</td><td>15</td></tr>
<tr><td>4</td><td>8</td><td>17</td></tr>
</table></td><td>3</td></tr>
<tr><td></td><td>b.</td><td>Given the directed graph G in Fig. 1, find a shortest path from source 0 to each of the vertices using Dijkstra algorithm.


Fig. 1: A directed graph G</td><td>4</td></tr>
<tr><td>Q.2</td><td>a.</td><td>Suppose A,B,C,D,E,F,G and H are 8 items and suppose they are assigned weights as follows:
<table>
<tr><td>Data items:</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td></tr>
<tr><td>Weight :</td><td>15</td><td>24</td><td>10</td><td>5</td><td>12</td><td>10</td><td>8</td><td>7</td></tr>
</table></td><td>1.5+1.5</td></tr>
</table>

		Now construct the Tree by using Huffman Algorithm and encoding the each node.	
	b.	Solve the following instance of the 0/1 knapsack problem using dynamic programming approach. Assume that the knapsack capacity is 5.	**4**

Item	**Weight**	**Benefit**
1	2	$4
2	3	$5
3	4	$6
4	6	$10

Q.3	**a.**	Apply Floyd-Warshall algorithm to find all-pairs shortest-paths of the graph *F* given in Fig. 2.	**5**
	b.	What is the advantage of Bellman-Ford Algorithm over Dijkstra's Algorithm?	**2**

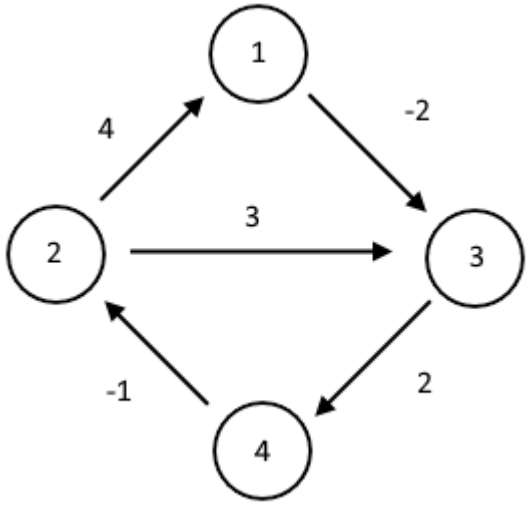
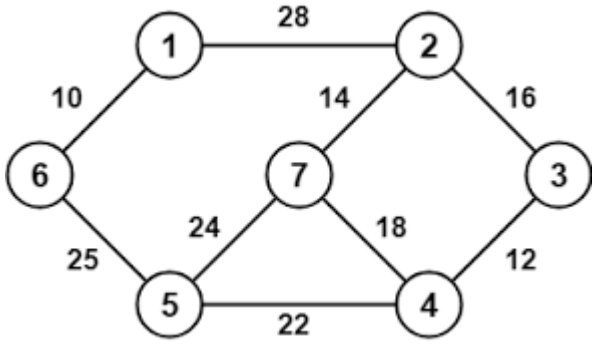
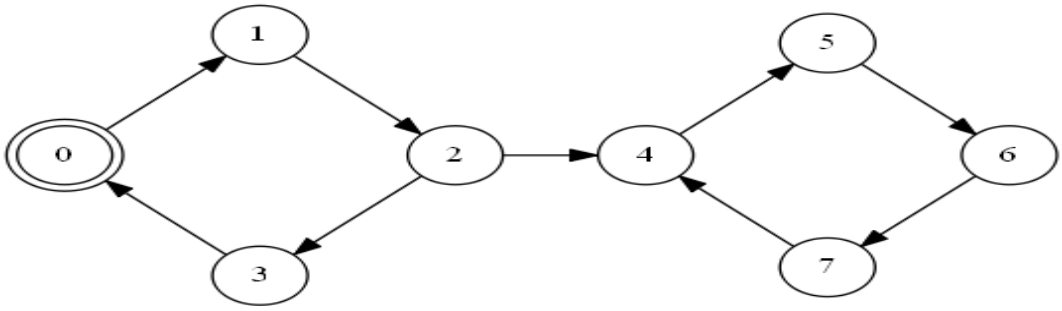
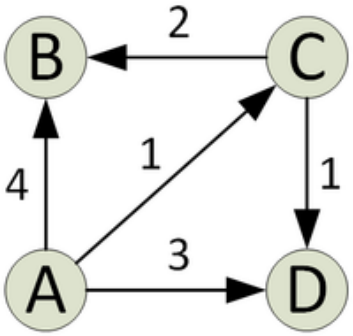


Fig. 2: A weighted graph *F*

Q.4	**a.**	Apply Prim's algorithm on the weighted graph *P* given in Fig. 3 to find a minimum spanning tree of *P.* Start node is 1.  **Fig. 3: A weighted graph *P***	**5**
	b.	Determine the space complexity of following code: **int sum(int d[], int b[] , int p)** **{** **int r = 0;** **for (int i = 0; i < p; ++i)** **{** **r += d[i]*b[i];** **}** **return r;** **}**	**2**
Q.5	**a.**	Find the strongly connected components of the directed graph S given in Fig. 4. Then draw the component graph of S  **Fig. 4: A directed graph S**	**4**

	b.	Apply DFS using topological sorting on the graph F in Fig. 5  **Fig. 5: A directed graph *F*** 	**3**