Total number of printed pages: **Programme(UG /7th Semester/UCSE701**

2022

Advance Algorithms

Full Marks : 100

Time : Three hours

The figures in the margin indicate full marks for the questions.

*Answer **any five** questions.*

1.	a)	You are given a collection of *n* bolts of different widths and *n* corresponding nuts. You are allowed to try a nut and bolt together, from which you can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together or two bolts together. The problem is to match each bolt to its nut. Design an algorithm for this problem with average-case efficiency in $O(n \log n)$.	10
	b)	Prove that the average case time-complexity of quick sort is O(n log n)	10
2.	a)	Suppose we're doing a sequence of n operations (numbered 1, 2, 3,...) on a data structure in which the ith operations cost is as follows: $cost = \begin{cases} 1 \text{ if } i \neq \text{ power of } 2 \\ i \text{ if } i = \text{power of } 2 \end{cases}$ For example, the following table shows the costs for each of the first few operations: operation number: 1 2 3 4 5 6 7 8 9 ... cost: 1 2 1 4 1 1 1 8 1 ... Use aggregate analysis to determine amortized cost per operation.	10
	b)	Find the amortized cost per operation of augmented stack using potential analysis	10
3.	a)	Solve the following 0/1 knapsack problem by Dynamic programming (weight limit W=11):	10

Item	Weight(w_i)	Value(v_i)
1	1	1
2	2	6
3	5	18

4	6	22
5	7	28

<table>
<tr><td></td><td>b)</td><td>Suppose the letters a, b, c, d, e, f have probabilities ½, 1/4, 1/8, 1/16, 1/32, 1/32 respectively. Find the Huffman code for letter a, b, c, d, e, f.</td><td>10</td></tr>
<tr><td>4.</td><td>a)</td><td>Compute the time complexity of Ford-Fulkerson algorithm to find the maximum flow of the graph.</td><td>5</td></tr>
<tr><td></td><td>b)</td><td>State the Max-flow min-cut theorem for network flow analysis. Trace the execution of Ford-Fulkerson algorithm to find the maximum flow of the graph given in figure 1.

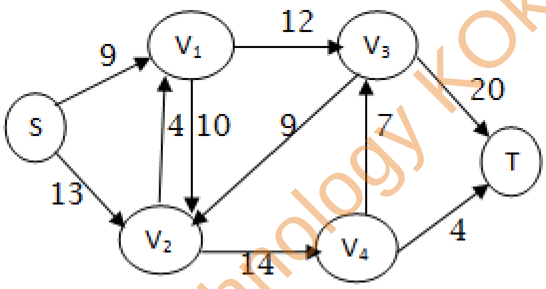


Figure 1</td><td>3+12</td></tr>
<tr><td>5.</td><td></td><td>Solve the APSP problem using Floyd-Warshall's algorithm for the graph given in figure 2:

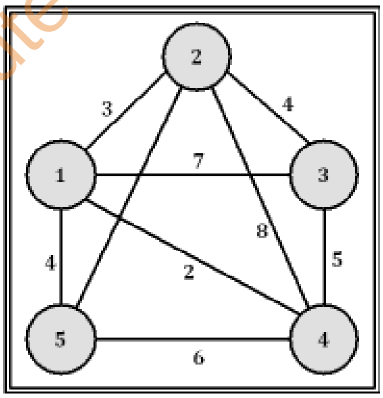


Figure 2</td><td>20</td></tr>
<tr><td>6</td><td>a)</td><td>Define the classes P and NP. Discuss diagrammatically the relations among P class, NP class, NP hard and NP complete.</td><td>10</td></tr>
<tr><td></td><td>b)</td><td>Trace the execution of vertex cover problem using Approximation algorithm of the graph given in figure 3. Also find the Approximation ratio (p(n)).</td><td>10</td></tr>
</table>

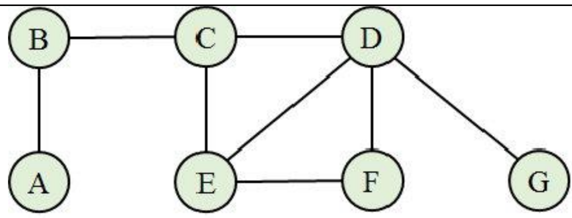
<table>
<tr><td></td><td></td><td>

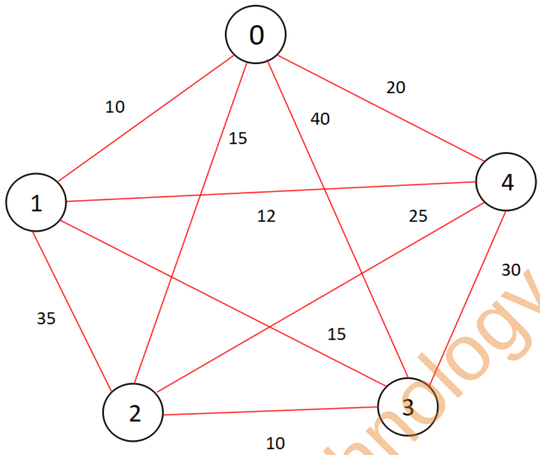
Figure 3</td><td></td></tr>
<tr><td>7.</td><td>a)</td><td>Apply the 2-approximate and 1.5-approximate algorithm for the Travelling Salesman Problem of the graph given in figure 4.


Figure 4</td><td>10+10=20</td></tr>
</table>