USN | | | | | | | | | |

Internal Assessment Test 2 – Aug 2022

Sub:	Design and Analysis of Algorithms				Sub Code:	18CS42	Branch:	CSE
Date:	03/08/2022	Duration:	90 mins	Max Marks: 50	Sem/Sec:	IV/A,B&C		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	a. Define i) Optimal solution ii) Feasible solution. iii)Will greedy method yield an optimal solution always?	[2+2+1]	CO3	L2

Ans: Most of the optimization problems have n inputs and require us to obtain a subset that satisfies some constraints. Any subset that satisfies these constraints is called a feasible solution.
A feasible solution that either maximizes or minimizes a given objective function is known as optimal solution.
Although greedy method provides an efficient algorithm for many optimization problems, there are some problems where greedy approach does not guarantee an optimal solution. Example 0-1 knapsack

		MARKS	CO	RBT
	b. Solve the greedy knapsack problem where m=10, n=4, P=(40,42,25,12), W=(4,7,5,3)	[5]	CO3	L2

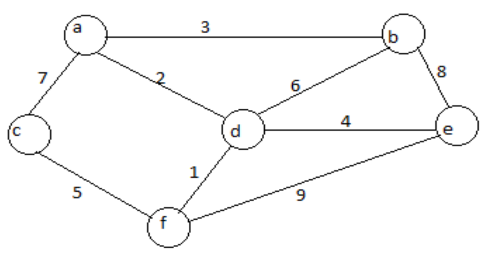
Ans:
Step1: Calculate per unit value of each item
Per unit value of item1 = 40/4 = 10
Per unit value of item2 = 42/7 = 6
Per unit value of item1 = 25/5 = 5
Per unit value of item1 = 12/3= 4

Step2: Sort the items as per per unit value and take the items in that order

Items #	Amount taken	Value	Remaining sack capacity
1	4	40	6
2	6	36	0

4 units of item1 and 6 units of item2 are taken to obtain a total value of 76

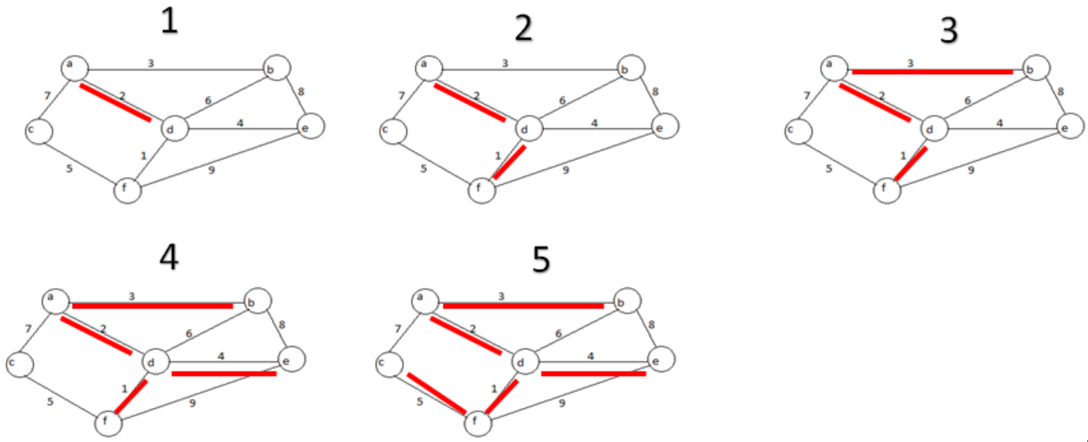
		MARKS	CO	RBT
2	Define minimum spanning tree. Find minimum cost spanning tree for the following graph using Prim's algorithm. Compare it with Kruskal's algorithm	[2+6+2]	CO2	L2



Ans: A spanning tree of an undirected connected graph is its connected acyclic subgraph (i.e., a tree) that contains all the vertices of the graph. A minimum spanning tree is a spanning tree whose weight is smallest where the weight of a tree is defined as the sum of the weights on all its edges.

Create minimum spanning tree starting with node a. In each of the subsequent steps, identify the edge with minimum weight that is going from a node in the partial tree already created and to a node outside the tree. Add that edge and the node to MST.

The steps can be shown by drawing the growth of the tree (shown below) or using a table.



Weight of the MST is 2+1+3+4+5 = 15

3	What is job sequencing with deadlines problem? Let n=5, profits [10,3,33,11,40] and deadlines[3,1,1,2,2] Find the optimal sequence of jobs with deadlines.	[10]	CO2	L2

Ans: We are given a set of n jobs. Associated with job i is an integer deadline di >= 0 and a profit pi> 0.
For any job i, the profit pi is earned iff the job is completed by its deadline. To complete a job, one has to process the job on a machine for one unit of time (i.e. all jobs are 1 unit long)
A feasible solution for this problem is a subset J of all jobs such that each job in this subset can be completed by its deadline. The value of a feasible solution J is the sum of the profits of the jobs in J. An optimal solution is a feasible solution with maximum value (2 marks)

Solution to the given job sequencing with deadlines problem (8 marks)

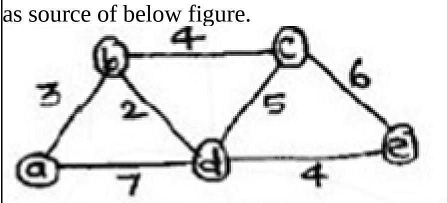
Sort the jobs in decreasing order of profits and schedule them in that order. Schedule the job as late as possible. If there is no free slot, discard it.

Sorted order of Job Numbers with profit and deadline in backets:
5(40,2), 3(33,1), 4(11,2), 1(10,3), 2(3,1).

1. Schedule 5 at slot 2
2. Schedule 3 at slot 1
3. 4 cannot be schedule as all slots up to 2 are full
4. Schedule 1 at slot 3
5. 2 cannot be schedule as all slots up to 1 are full

Total profit earned: 40+33+10 = 83

4	Write Dijkstra’s algorithm and apply the same to find the single source shortest path for graph taking vertex ‘a’ as source of below figure.	[10]	CO3	L2



```
ALGORITHM Dijkstra(G, s)
    //Dijkstra's algorithm for single-source shortest paths
    //Input: A weighted connected graph G = ⟨V, E⟩ with nonnegative weights
    //       and its vertex s
    //Output: The length d_v of a shortest path from s to v
    //        and its penultimate vertex p_v for every vertex v in V
    Initialize(Q)   //initialize priority queue to empty
    for every vertex v in V
        d_v ← ∞;   p_v ← null
        Insert(Q, v, d_v)   //initialize vertex priority in the priority queue
    d_s ← 0;   Decrease(Q, s, d_s)   //update priority of s with d_s
    V_T ← ∅
    for i ← 0 to |V| − 1 do
        u* ← DeleteMin(Q)   //delete the minimum priority element
        V_T ← V_T ∪ {u*}
        for every vertex u in V − V_T that is adjacent to u* do
            if d_u* + w(u*, u) < d_u
                d_u ← d_u* + w(u*, u);   p_u ← u*
                Decrease(Q, u, d_u)
```

Ans:

Step #	Node which is added to the list of nodes to which shortest path is already calculated	Current shortest distance to other nodes
1	a(-,0)	b(a,3), d(a,7), c(-,∞), e(-,∞)
2	b(a,3)	d(b,5), c(b,7),e(-,∞)
3	d(b,5)	c(b,7), e(d,9)
4	c(b,7)	e(d,9)
5	e(d,9)	

Q.No	Question	Marks	CO	Level
5	a. Compare Greedy approach with Dynamic Programming.	[4]	CO3	L2
	b. Describe the various steps involved in bottom up heap construction with example. Sort the given list of numbers using heap sort 2, 9, 7, 6, 5, 8	[6]	CO1	

Ans:

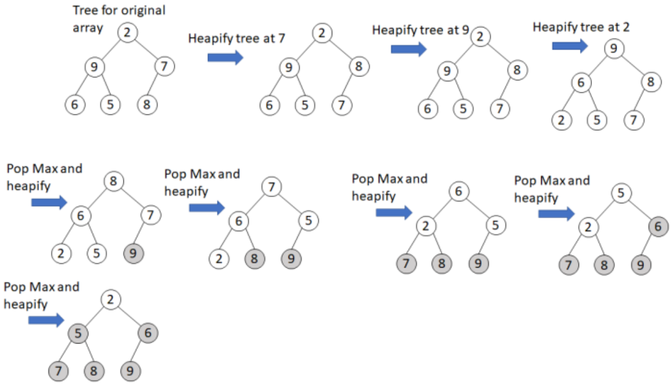
FEATURE	GREEDY METHOD	DYNAMIC PROGRAMMING
Feasibility	In a greedy Algorithm, we make whatever choice seems best at the moment in the hope that it will lead to global optimal solution.	In Dynamic Programming we make decision at each step considering current problem and solution to previously solved sub problem to calculate optimal solution .
Optimality	In Greedy Method, sometimes there is no such guarantee of getting Optimal Solution.	It is guaranteed that Dynamic Programming will generate an optimal solution as it generally considers all possible cases and then choose the best.
Recursion	A greedy method follows the problem solving heuristic of making the locally optimal choice at each stage.	A Dynamic programming is an algorithmic technique which is usually based on a recurrent formula that uses some previously calculated states.
Memorization	It is more efficient in terms of memory as it never look back or revise previous choices	It requires dp table for memorization and it increases it's memory complexity.
Time complexity	Greedy methods are generally faster. For example, Dijkstra's shortest path algorithm takes O(ELogV) time.	Dynamic Programming is generally slower. For example, Bellman Ford algorithm takes O(VE) time.
Fashion	The greedy method computes its solution by making its choices in a serial forward fashion, never looking back or revising previous choices.	Dynamic programming computes its solution bottom up or top down by synthesizing them from smaller optimal sub solutions.

Ans: b

Step 0: Initialize the structure with keys in the order given

Step 1: Starting with the last (rightmost) parental node, fix the heap rooted at it, if it doesn't satisfy the heap condition: keep exchanging it with its larger child until the heap condition holds

Step 2: Repeat Step 1 for the preceding parental node

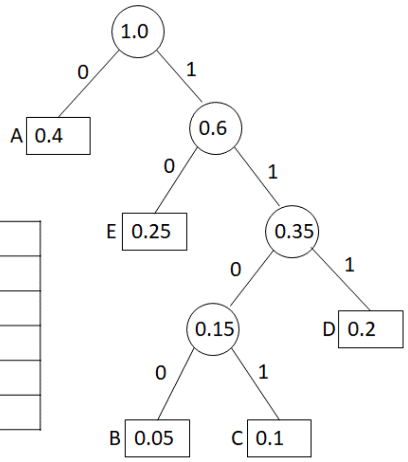


6 What are Huffman codes? Construct a Huffman tree and code for the following data. When a parent is created for two nodes, the edge from parent to the node with lower frequency should get label “0” and the other one “1”. [2+6+2] CO3 L3

Character	A	B	C	D	E
Probability	0.4	0.05	0.1	0.2	0.25

Encode the text ABC and decode the text 1100100111, using the above code.

Ans: Huffman code is a prefix free variable length coding scheme used to represent characters. Using this scheme, we can reduce the number of bits required to store text documents.



Character	Code
A	0
B	1100
C	1101
D	111
E	10

Code for ABC: 011001101

Decoded text for 1100100111: BEAD

BEST OF LUCK

CO-PO and CO-PSO Mapping																			
Course Outcomes		**Blooms Level**	**Modules covered**	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	Describe the computational solution to well-known problems like searching and sorting	**L1**	1,2	2	3	2	2	-	2	-	-	2	-	-	-	2	-	-	2
CO2	Estimate computational complexity of various algorithms.	**L2**	1,2,3,4,5	3	3	2	2	-	2	-	-	2	-	-	-	2	-	-	2
CO3	Devise an algorithm using appropriate design strategies for computation problems.	**L3**	2,3,4,5	3	3	2	2	-	2	-	-	2	-	-	-	2	-	-	2

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		