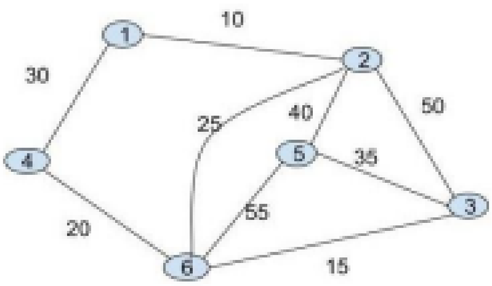
USN | | | | | | | | | |

Internal Assessment Test 3 – August 2024

Sub: **ANALYSIS & DESIGN OF ALGORITHMS**		Sub Code: **BCS401**		Branch: **AIML/CSEAIML**
Date:	Duration: **90 min**	Max Marks: **50**	Sem/Sec: **IV -A, B, C**	**OBE**

	Answer any FIVE FULL Questions	**MARKS**	**CO**	**RBT**
1	What is a heap? Explain what is Max heap? How is it used in the Sorting technique? Explain with a neat diagram.	10	CO3	L1
2	How many bits may be required for encoding the message 'mississippi'? Find how many bits will be transmitted using fixed sized codes and variable sized codes.	10	CO4	L2
3	Write and explain the iterative backtracking algorithm. Draw the state space tree for the 4-queens problem and give the solution tuples.	10	CO5	L3
4	Give the formulation knapsack problem using branch and bound and find the optimal solution using branch and bound with n=4, m=15, (p1…p4) = (15, 15, 17, 23) & (w1…w4) = (3, 5, 6, 9).	10	CO5	L2
5	Find the cost of the Minimum Spanning Tree using Kruskal's algorithm? 	10	CO4	L3
6	Formulate the Knapsack problem with greedy method and find the optimal solution for n=7, m=15, (p1-p7)=(10,5,15,7,6,18,3), (w1-w2)=(2,3,5,7,1,4,1).	10	CO4	L3

CI **CCI** **HOD** --------------

------------------------------All the Best---

USN | | | | | | | | | | |

Internal Assessment Test 3 – August 2024

Sub: **ANALYSIS & DESIGN OF ALGORITHMS**		Sub Code: **BCS401**		Branch: **AIML/CSEAIML**
Date:	Duration: **90 min**	Max Marks: **50**	Sem/Sec: **IV -A, B, C**	**OBE**

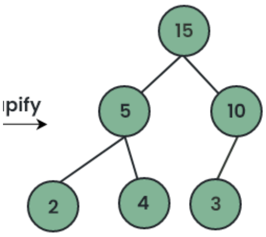
	<u>Answer any FIVE FULL Questions</u>	**MARKS**	**CO**	**RBT**
1	What is a heap? Explain what is Max heap? How is it used in the Sorting technique? Explain with a neat diagram.	10	CO3	L1

Ans:

A heap is defined as a binary tree with keys assigned to its nodes provided the following conditions are met:

- The binary tree is a complete binary tree. All its levels are full. Rightmost leaves may be missing.
- The key at each node is either greater than or equal to the keys of its children (or) the key at each node is either lesser than or equal to the keys of its children.

A max heap is defined as a heap where every internal node is greater than or equal to its child nodes.



		MARKS	CO	RBT
2	How many bits may be required for encoding the message 'mississippi'? Find how many bits will be transmitted using fixed sized codes and variable sized codes.	10	CO4	L2

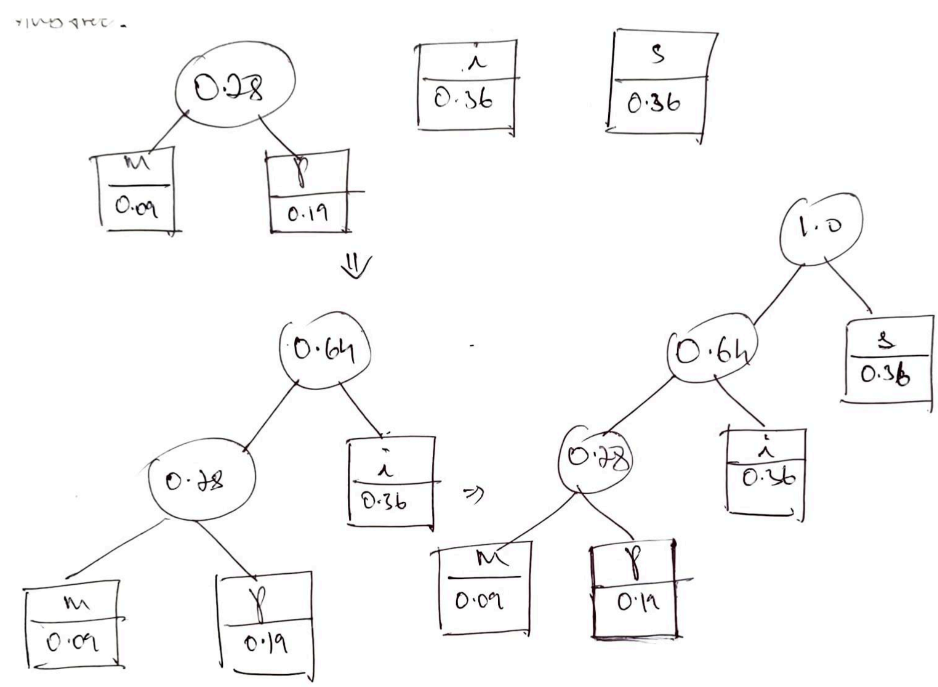
Ans:

Consider the following characters 'mississippi'.

The frequency and probabilities of the characters are as follows:

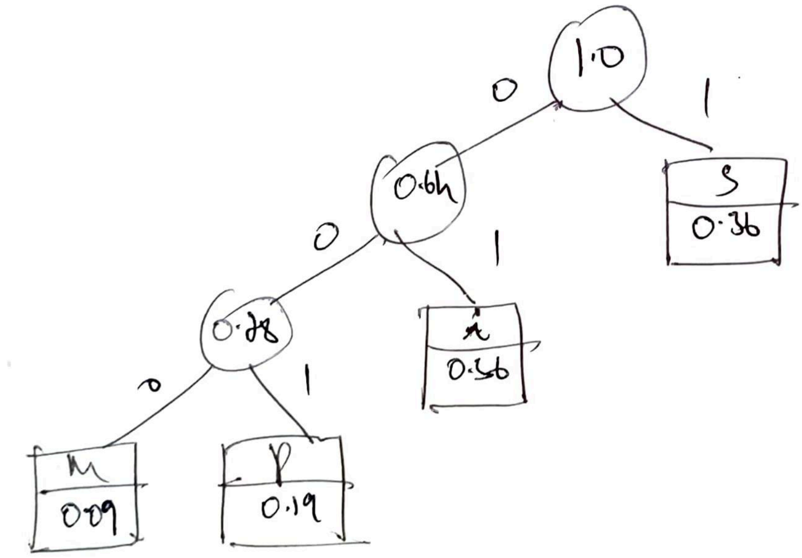
Character	m	i	s	p
Frequency	1	4	4	2
Probability	0.09	0.36	0.36	0.19

Merge the smallest weights and make them the left and right subtrees. The root will hold the sum of the weights.



Repeat the same until a single tree is obtained.

Allocate 0 for left subtree and 1 for right subtree.



These are the following variable codes that are obtained for the characters:

Character	Code
m	000
i	01
s	1
p	001

<table>
<tr><td></td><td>
The no. of bits per character in these codes are:

(3*0.09)+(2*0.36)+(1*0.36)+(3*0.19) = 1.92

The compression ratio (3-1.92)/3*100 = 36%

Huffman encoding will use 36% less memory.
</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>
Write and explain the iterative backtracking algorithm. Draw the state space tree for the 4-queens problem and give the solution tuples.

Ans;

9.1 Backtracking

Many problems which deal with searching for a set of solutions satisfying some constraints can be solved using backtracking. Now, let us see "What is backtracking?"

Definition: Backtracking is a systematic method of searching for the solution to a combinatorial problem by means of an algorithm. Some of the problems that have exponential time complexity can be best solved using backtracking method. The main idea is to create solutions by selecting one component at a time and evaluate such partially constructed candidates as follows:
• If a partially constructed solution can be developed further without violating the problem's constraints, it is done by taking the first remaining legitimate option for the next component.
• If there is no legitimate option for the next component, then remaining alternate components need not be considered. In this case, the algorithm backtracks to replace the last component of the partially constructed solution with its next option.

Ex: The various problems that can be solved using backtracking method are:
• N-queens.
• Sum of subset
• Hamiltonian circuit
• Knapsack etc.

Now, let us see "What is N-queens problem?"

Definition: The N-queen's problem is stated as follows: "Given N x N chess board, it is required to place all N-queens on the chessboard such that no two queens attach each other". That is, two or more queens should not be placed in the same row, same column or same diagonal. The placing of N-queens on the chessboard using the above constraints is called N-queens problem.

For example, all four queens can be placed on the chessboard such that no two queens are attacking each other is shown on right hand side using a 2-dimensionay array.

<table>
<tr><td></td><td>Q</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>Q</td></tr>
<tr><td>Q</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>Q</td><td></td></tr>
</table>
4 x 4 chessboard
</td><td>10</td><td>CO5</td><td>L3</td></tr>
</table>

First, let us see "What is a state? What is a state space? What is state space tree?"

Definition: In any of the problem that can be solved using backtracking, the solution can be expressed as n-tuple:

$$(x_1, x_2, x_3, \ldots\ldots x_n)$$

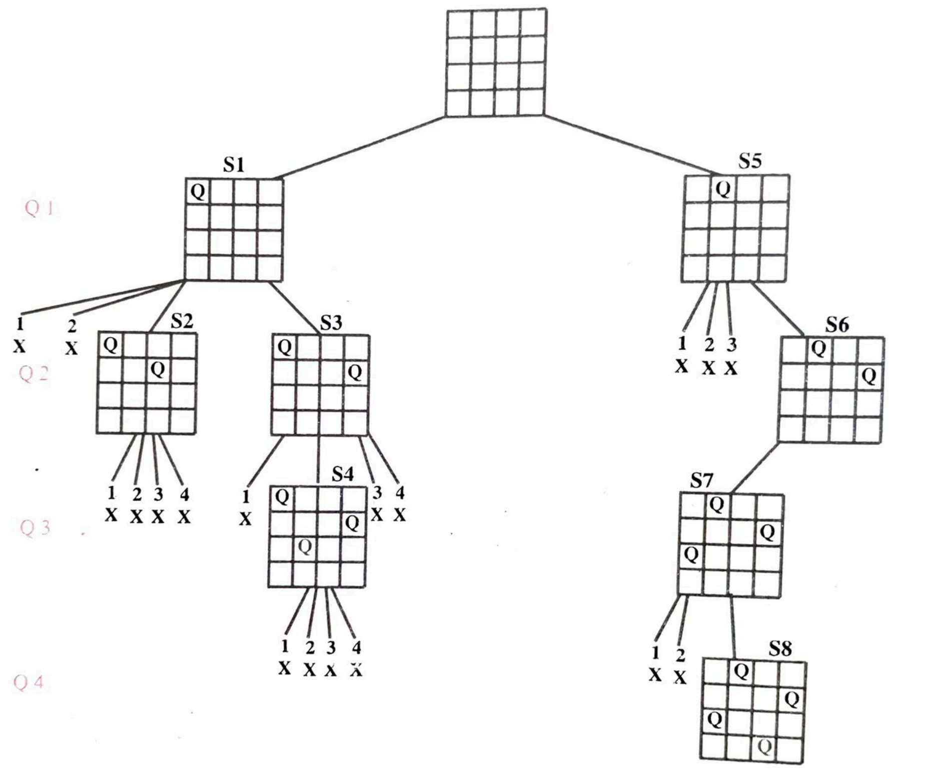
The n-tuple $(x_1, x_2, \ldots\ldots, x_n)$ is called a *state/solution* of the problem where as a set of all such possible states/solutions is called *state/solution space* and it is denoted by *S*.

Definition: The tree corresponding to the *state/solution space* is called *state space tree* or *solution space tree* which is normally constructed using *depth first search* method. The state/solution space tree can be constructed as follows:

- The root represent the initial state before the search for a solution begins
- The nodes of the first level in the tree indicate the choices made for the first component of a solution. The details such as whether the component is considered or not considered is clearly given. Sometimes, only the selected component is shown.
- The nodes of the second level in the tree indicate the choices made for the second component of a solution and so on.
- A node in the *state space tree* may be promising node or non-promising node. A promising node is the one whose partially constructed solution moves towards the solution. If it is moving away from the solution, the node is called non-promising node.
- Leaves may represent either nonempty dead nodes or complete solutions found by the algorithms
- If the current node is promising then, its child is generated by adding the first remaining legitimate option for the next component of a solution and processing moves to the new child.
- If the current node turns out to be non-promising node, the algorithm backtracks to its parent node to consider the next possible option for its last component. If there is no such option, then it backtracks one more level up the tree and so on.
- Finally, if the algorithm reaches a complete solution to the problem, it either stops (if only one solution is required) or backtracks to continue searching for other possible solutions.

Now, let us "Give the state space tree for solving 4-queens problem for at least one solution" The state space tree can be obtained as shown below:

- Start with an empty board and then place queen 1 in row 1 and column 1
- Then place queen 2 in row 2 after trying unsuccessfully in columns 1 and 2 but placing in column 3 i.e., in square(2, 3).
- From the current configuration, we cannot place queen 3 and we reach the dead end. So, algorithm backtracks and puts queen 2 in next position (2, 4). Then queen 3 is placed at (3, 2) which proves to be another dead end. Proceeding in this way, we get the following state space tree.

	 Note: In the above diagram, Q1, Q2, Q3 and Q4 represent first queen, second queen, third queen and fourth queen respectively. The symbol X denote an unsuccessful attempt to place a queen in the specified column. The names S1, S2, S3, S4, S5, S6, S7 and S8 indicate the order in which the nodes are generated.			
4	Give the formulation knapsack problem using branch and bound and find the optimal solution using branch and bound with n=4, m=15, (p1…p4) = (15, 15, 17, 23) & (w1…w4) = (3, 5, 6, 9). Ans:	10	CO5	L2

$n = 4$, $M = 15$

Weight	Value	v_i/w_i
3	15	5
5	15	3
6	17	2.8
9	23	2.5

The capacity of the knapsack = 15.
The upperbound is calculated by

$ub = v + (M - w)\, v_{i+1}/w_{i+1}$

$v \to$ profit of all the chosen objects
$w \to$ weight of all the chosen objects.

Item 0:

$v = 0$, $w = 0$,
$(M - w) = 15 - 0 = 15$
$v_{i+1}/w_{i+1} = v_1/w_1 = 5$.
$ub = 0 + (15) * 5$
$\boxed{ub = 75}$

Item 1:

$v = 15$, $w = 3$
$(M - w) = 15 - 3 = 12$
$v_{i+1}/w_{i+1} = v_2/w_2 = 3$
$ub = 15 + (12) * 3$
$\boxed{ub = 51}$

Without item:

$v = 0$, $w = 0$
$(M - w) = (15 - 0) = 15$
$v_{i+1}/w_{i+1} = v_2/w_2 = 3$
$ub = 0 + (15) * 3$
$\boxed{ub = 45}$

Item 2:

$V = 15 + 15 = 30$, $W = 3 + 5 = 8$.

$(M - W) = (15 - 8) = 7$

$v_{i+1}/w_{i+1} = v_3/w_3 = 2.8$

$ub = 30 + (7)\,2.8$

$= 30 + 19.6$

$\boxed{ub = 49.6}$

Without Item 2:

$V = 15 + 0 = 15$, $W = 3 + 0 = 3$

$(M - W) = (15 - 3) = 12$

$v_{i+1}/w_{i+1} = v_3/w_3 = 2.8$

$ub = 15 + (12) * 2.8$

$= 15 + 33.6$

$\boxed{ub = 48.6}$

Item 3:

$V = 30 + 17 = 47$, $W = 8 + 6 = 14$

$(M - W) = (15 - 14) = 1$

$v_{i+1}/w_{i+1} = v_4/w_4 = 2.5$

$ub = 47 + 2.5$

$\boxed{ub = 49.5}$

Without Item 3:

$V = 30 + 0 = 30$, $W = 8 + 0 = 8$

$(M - W) = (15 - 8) = 7$

$v_4/w_4 = 2.5$

$ub = 30 + (7) * 2.5$

$= 30 + 17.5$

$\boxed{ub = 47.5}$

Item 4:

$V = 47 + 23 = 60$, $W = 14 + 9 = 23$

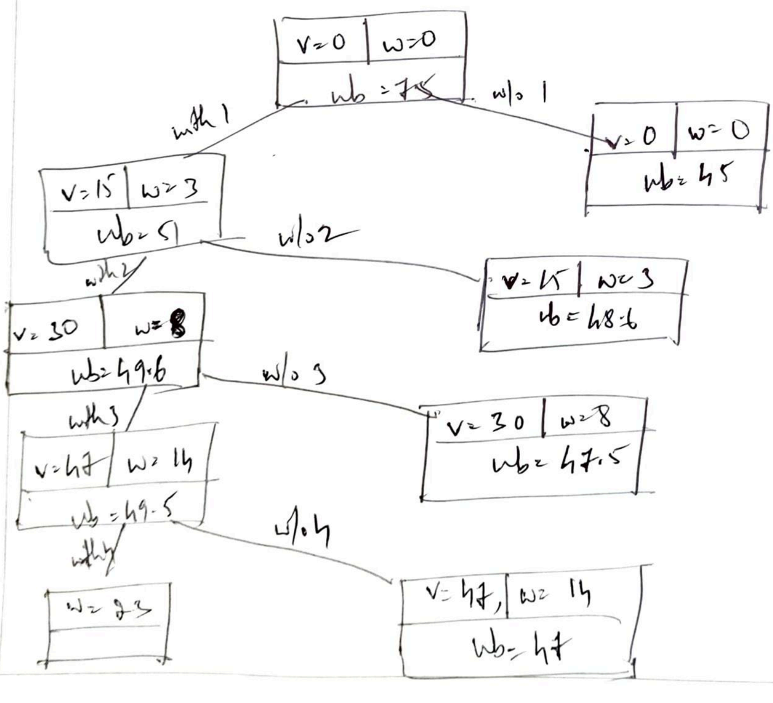
Without item 4:

$V = 47 + 0 = 47$, $W = 14$

$(M - W) = (15 - 14) = 1$, $v/w = 0$

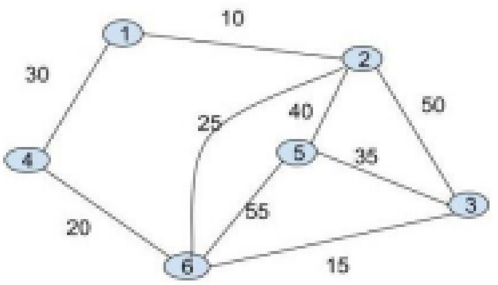
$ub = 47 + 0$

$\boxed{ub = 47}$



Final solution vector is {1, 2, 3}.

5	Find the cost of the Minimum Spanning Tree using Kruskal's algorithm?	10	CO4	L3



Ans:

ment No.

Page No.:

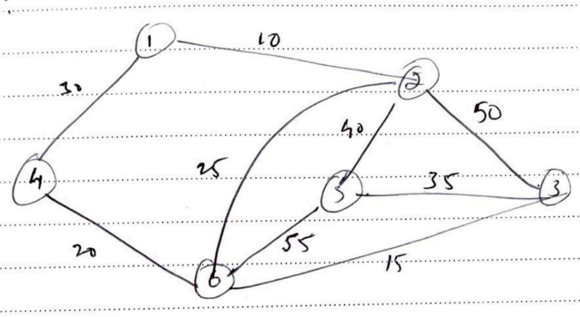
5) Step 1: Arrange the edges in NON-DECREASING order.

Step 2: Select the least edge.

Step 3: Add to Minimum Spanning Tree & if it does not form a cycle.

Step 4: Repeat for (n-1) edges.

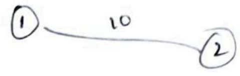
Graph:



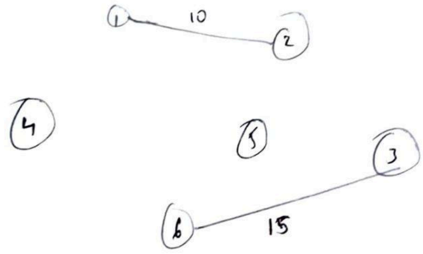
Edges	Cost
1-2	10
3-6	15
4-6	20
2-6	25
1-4	30
3-5	35
2-5	40
2-3	50
5-6	55

ent

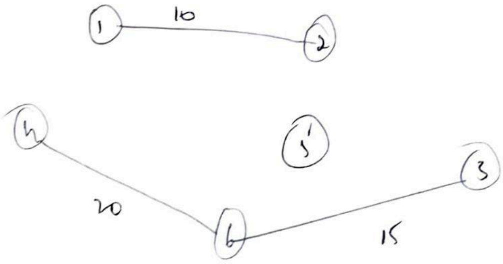
Add edge : 1-2



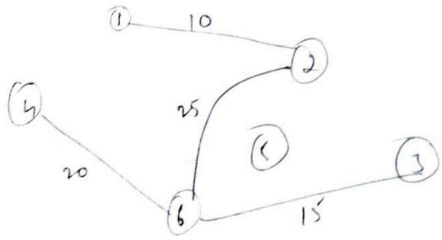
Add edge 3-6 :



Add edge : 4-6



Add edge 2-6

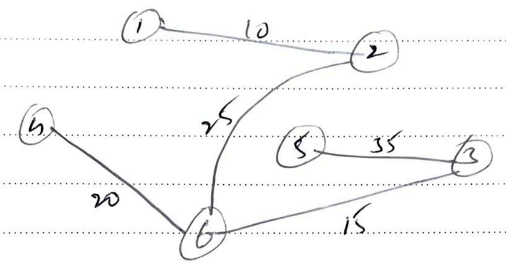


nt No. Page No.:

~~Add~~ edge

* Cannot add edge 1-4 since it forms a cycle.

Add edge 3-5 :



(n-1) edges have been added.

6	Formulate the Knapsack problem with greedy method and find the optimal solution for n=7, m=15, (p1-p7)=(10,5,15,7,6,18,3), (w1-w2)=(2,3,5,7,1,4,1).	10	CO4	L3

Ans:

P	W	P_i/W_i
10	2	5
5	3	1.6
15	5	3
7	7	1
6	1	6
18	4	4.5
3	1	3

n = 7
m = 15.

Rearrange the items based on non-decreasing order of Profit/Weight ratio.

Profit	Weight	P_i/W_i
6	1	6
10	2	5
18	4	4.5
15	5	3
3	1	3
5	3	1.6
7	7	1

M = 15.

Select item 1: $P_1 = 6$, $W_1 = 1$

Total profit = 6

Total weight = 1 . $TW \le M$

Select item 2: $P_2 = 10$, $W_2 = 2$

Total profit = 6+10 = 16

Total weight = 1+2 = 3 . $TW \le M$

Select item 3: $P_3 = 18$, $W_3 = 4$

Total profit = 16+18 = 34

Total weight = 7 $(7 \le 15)$

Select item 4: $P_4 = 15$, $W_4 = 5$

Total profit = 34+15 = 49

Total weight = 7+5 = 12 $(12 \le 15)$

Select item 5: $P_5 = 3$, $W_5 = 1$

Total profit = 49+3 = 52

Total weight = 12+1 = 13 $(13 \le 15)$

Select item 6: $P_6 = 5$, $W_6 = 3$

Total profit = 52+5 = 57

Total weight = 13+3 = 16

16 does not follow the constraint

$\therefore (M - TW)/W_i * P_i$ as $(15-13)/3 * 5 = 3.33$

$\boxed{\text{Total profit} = 52 + 3.3 = 55.3}$

Object (i)	w_i	P_i	$x: 1$ or rc/w_i	$P_i * x$	$rc = rc - w_i * x$
1	1	6	1	6	$rc = 15 - 1 = 14$
2	2	10	1	10	$rc = 14 - 2 = 12$
3	4	18	1	18	$rc = 12 - 4 = 8$
4	5	15	1	15	$rc = 8 - 5 = 3$
5	1	3	1	3	$rc = 3 - 1 = 2$
6	3	5	2/3	$\frac{2}{3} \times 5 = 3.3$	$rc = 2 - (\frac{2}{3} \times 3) = 0$
			Total =	55.3	

The total profit is 55.3

CI **CCI** **HOD**

--All the Best--