K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
III SESSIONAL TEST QUESTION PAPER 2018 – 19 EVEN SEMESTER

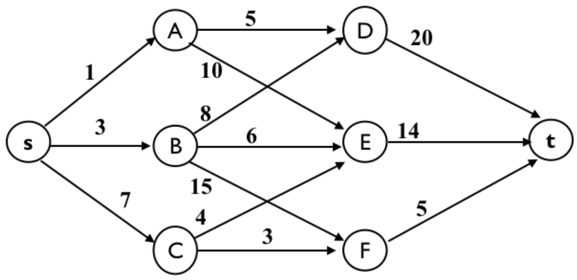
Scheme and Answers

SET – A/~~B~~

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

Degree	:	**B.E**	**Semester :**	**IV**
Branch	:	**Computer Science and Engineering**	**Course Code :**	**17CS43**
Course Title	:	**Design and Analysis of Algorithms**	**Date :**	**21-May-2019**
Duration	:	**90 Minutes**	**Max Marks :**	**30**

Note: Answer ONE full question from each part.

Q No.	Question	Marks
1(a)	**Compare** the principle of optimality with Greedy approach. Consider the problem of optimal merge pattern to merge N files, and determine which pair of files should be merged at each. **Analyze** if decision sequences follows the principle of optimality.	5
Sch & Ans	Sch: 3 marks for comparison, 2 marks for merging N files Ans: With principle of optimality, the next decision in sequence of decisions is always optimal, whereas in Greedy approach, the next decision may not be optimal. For example, using Dijkstra's algo with negative weights, using greedy approach it may not give optimal results. The optimal merge pattern is always pick two smallest size files and merge them. Put the merge file back and this become another file. Thus each step, number of files decreases by 1, and merging will be complete after N-1 merges.	
(b)	**Calculate** the shortest path from source `s` to destination `t` for the following multi-stage graph using backward approach (forward reasoning). 	5

Sch & Ans	Sch: 3 marks for defining the right steps, and 2 marks for computation.
	
	Ans:
	`d(s,t)=min{d(s,D)+20, d(s,E)+14, d(s,F)+5}`
	`d(s,D)=min{d(s,A)+5, d(s,B)+8}`
	`d(s,E)=min{d(s,A)+10, d(s,B)+6, d(s,C)+4}`
	`d(s,F)=min{d(s,B)+15, d(s,C)+3}`
	
	Computation:
	`d(s,A)=1, d(s,B)=3, d(s,C)=7`
	`d(s,D)=min{1+5,3+8}=6`
	`d(s,E)=min{1+10,3+6,7+4}=9`
	`d(s,F)=min{3+15,7+3}=10`
	`d(s,t)=min{6+20,9+14,10+5}=15`	

<table>
<tr><td></td><td></td><td></td></tr>
<tr><td>(c)</td><td>Identify the reasons that are used to terminate a search path in a state-space tree of a branch-and-bound algorithm.</td><td>5</td></tr>
<tr><td>Sch
&
Ans</td><td>Sch: 2 marks each for first 2 reason, 1 mark for remaining

Ans:
A search in state space tree using Branch and Bound terminates, when any of the following conditions are met
i. The value of a node’s bound value is worse than the best solution seen so far
ii. The node represents a situation which is not feasible due to constraints imposed on the problem.
iii. The node has become a leaf node and reached the end of solution. At this point, if the node’s objective function is better than the best solution so far, update the value of best solution.</td><td></td></tr>
<tr><td>2(a)</td><td>List the OptimalBST algorithm and analyze its space and time complexity.</td><td>5</td></tr>
<tr><td>Sch
&
Ans</td><td>Sch: 3 marks for algo, and 2 marks for time complexity.

Ans:
Algo optimalBST(P[1:n])
# P[1:n] represents the probabilities of occurrence of n keys.
For i=1 to n do
   C[i,i-1]=0
   C[i,i]=P[i]
   R[i,i]=i
C[n+1,n]=0
For d=1 to n-1 do # diagonal count
   For i=1 to n-d to
      j=i+d
      minval = inf
      for k=i to j do
         if [Ci,k-1]+C[k+1,j] <minval then
            minval = C[I,k-1]+C[k+1,j]
            kmin = k
      R[i,j] = kmin
      Sum = P[i]
      For s=i+1 to j do
         Sum = sum+P[s]
      C[i,j]=minval + sum
Return C[1,n],R</td><td></td></tr>
<tr><td>(b)</td><td>Construct an OptimalBST for following 3 keys.
Pa=0.2, Pb=0.5, Pc=0.3</td><td>5</td></tr>
<tr><td>Sch
&
Ans</td><td>Sch: 0.5 mark each for diagonal computations C[1,1], C[2,2],C[3,3]: total 1.5 marks
1 mark each for computing C[1,2], and C[2,3], total : 2 marks
1.5 marks for computing C[1,3]
Thus, total = 1.5+2+1.5=5 marks

Ans:
C(1,0)=0, C(2,1)=0, C(3,2)=0
C(1,1)=p_1=0.2, R(1,1)=1</td><td>2</td></tr>
</table>

$C(2,2)=p_2=0.5,\ R(2,2)=2$

$C(3,3)=p_1=0.3,\ R(3,3)=3$

$C(1,2)=\Sigma_{1\le s\le 2}p_s+\min\{C(1,0)+C(2,2),C(1,1)+C(3,2)\}$
$\quad =0.7+\min\{0+0.5,0.2+0\}=0.9$

$R(1,2)=2$

$C(2,3)=\Sigma_{2\le s\le 3}p_s+\min\{C(2,1)+C(3,3),C(2,2)+C(4,3)\}$
$\quad =0.8+\min\{0+0.3,0.5+0\}=1.1$

$R(2,3)=2$

$C(1,3)=\Sigma_{1\le s\le 3}p_s+\min\{C(1,0)+C(2,3),C(1,1)+C(3,3),C(1,2)+C(4,3)$
$\quad =1.0+\min\{0+1.1,0.2+0.3,0.9+0\}=1.0+0.5=1.5$

$R(1,3)=2$

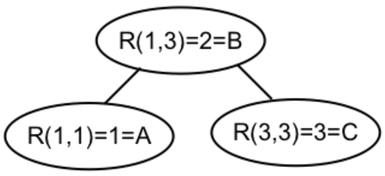
Comparison table and Root table

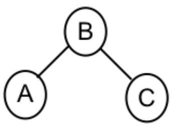
C(i,j)

	0	1	2	3
1	0	0.2	0.9	1.5
2		0	0.5	1.1
3			0	0.3
4				0

Root table

	0	1	2	3
1		1	2	2
2			2	2
3				3
4				





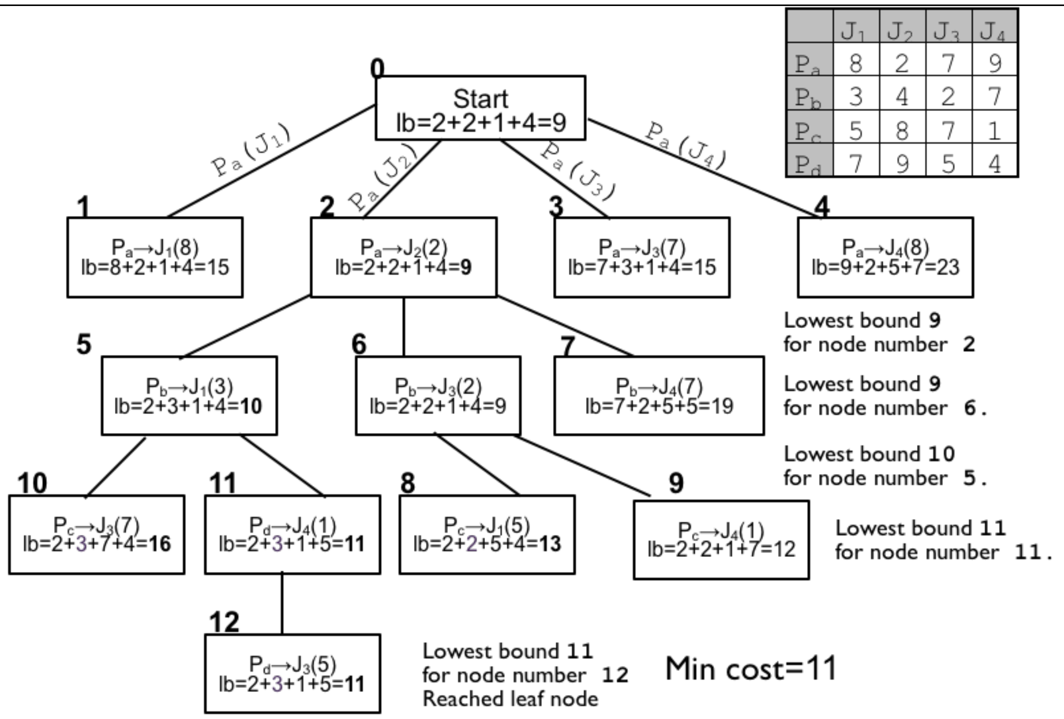
(c) **Apply** the branch and bound approach to **solve** the instance of following assignment problem among 4 persons for 4 jobs. **5**

	J_1	J_2	J_3	J_4
P_a	8	2	7	9
P_b	3	4	2	7
P_c	5	8	7	1
P_d	7	9	5	4

Sch & Ans

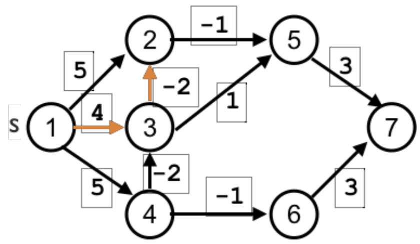
Sch: 1 mark for each level in state space tree. Total 5 levels→5 marks

Ans:



3(a)

Calculate the shortest path from node 2 to all the nodes for the graph below using Bellman Ford algorithm and **list** output after each of the iteration step.



5

Sch & Ans

Sch: 1 mark each for each iteration step

Ans:

With source as 1

Source as Node 1 (most may have done it)

k↓	1	2	3	4	5	6	7
1	0	5	4	5	∞	∞	∞
2	0	2	3	5	4	4	∞
3	0	1	3	5	1	4	7
4	0	1	3	5	0	4	4
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

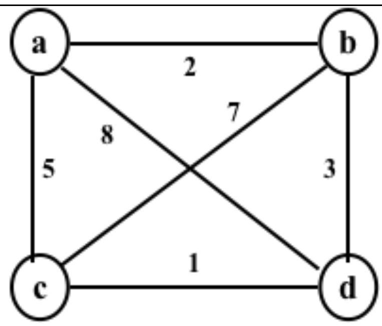
Source as node 2 (as per Question)

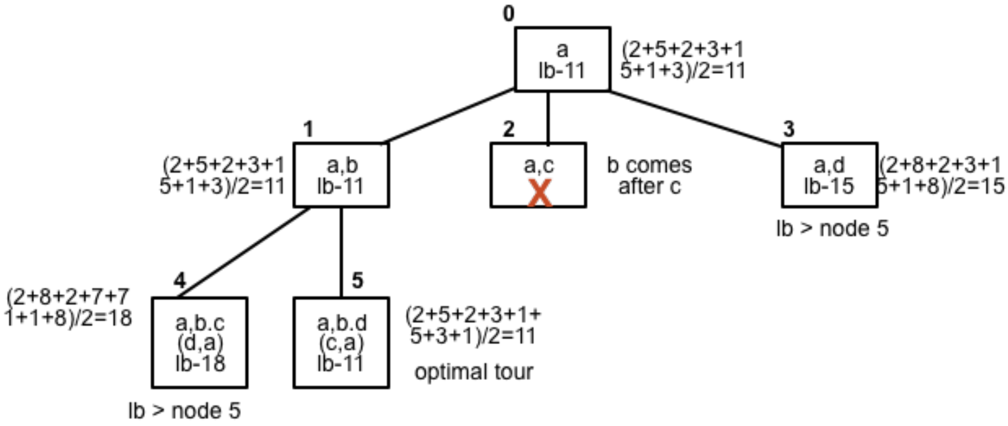
k↓	1	2	3	4	5	6	7
1	∞	0	∞	∞	**-1**	∞	∞
2	∞	0	∞	∞	**-1**	∞	**2**
3	∞	0	∞	∞	**-1**	∞	**2**

(b)

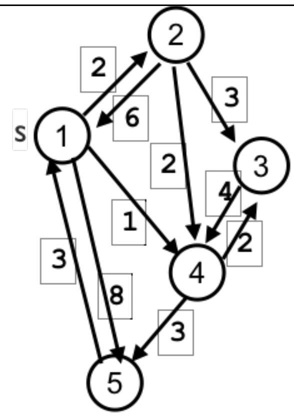
Establish the shortest tour for the Traveling Salesperson Problem using Dynamic Programming for the following graph. Discover all the applicable `g(i,S)` values in your analysis of TSP steps.

5

	 	
Sch & Ans	Sch: 1 mark each computing g(i,S) for |S|-0, 1, 2. 1 mark for computing corresponding J values, and 1 marks for working out the min cost tour. Ans: Goal:g(1,V-{1}) Power set of {2,3,4} Ø,{2},{3},{4},{2,3},{2,4},{3,4},{2,3,4} g(1,Ø)=c_{11}=0 g(2,Ø)=c_{21}=2 g(3,Ø)=c_{31}=5 g(4,Ø)=c_{41}=8 Compute g(i,S), for |S|=1 g(2,{3})=c_{23}+g(3,Ø)=7+5=12 g(2,{4})=c_{24}+g(4,Ø)=3+8=11 g(3,{2})=c_{32}+g(2,Ø)=7+2=9 g(3,{4})=c_{34}+g(4,Ø)=1+8=9 g(4,{2})=c_{42}+g(2,Ø)=3+2=5 g(4,{3})=c_{43}+g(3,Ø)=1+5=6 J(2,{3})=3,J(2,{4})=4,J{3,{2}=2 J(3,{4})=4,J(4,{2})=2,J(4,{3}=3 Compute g(i,S), for |S|=2 g(2,{3,4})=min{c_{23}+g(3,{4}),c_{24}+g(4,{3})=min{7+9,3+6}=9, J(2,{3,4})=4 g(3,{2,4})=min{c_{32}+g(2,{4}),c_{34}+g(4,{2})=min{7+11,1+5}=6,J(3,{2,4})=4 g(4,{2,3})=min{c_{42}+g(2,{3}),c_{43}+g(3,{2})=min{3+12,1+9}=10,J(2,{3,4})=3 g(1,{2,3,4})= min{c_{12}+g(2,{3,4}),c_{13}+g(3,{2,4}),c_{14}+g(4,{2,3})} =min{2+9, 5+6, 8+10} = 11 J(1,{2,3,4})=2 (can be 3 as well) The optimal tour length is 11, path is 1,2,4,3 J(1,{2,3,4})=2, J(2,{3,4})=4, J(4,{3})=3 If J(1,{2,3,4})=3, then J(3,{2,4})=4, J(4,{2})=2 And thus path is 1,3,4,2.	
(c)	**Apply** the branch and bound algorithm to solve Traveling Salesperson Problem for the graph Q3b.	**5**
Sch & Ans	Sch: 1 mark each for each node of state space tree. Ans:	



4(a)	**Analyze** the problem of designing a system that is composed of n devices connected in series and maximizing its reliability using device duplication. **Examine** the maximization expression along with applicable constraints to be used for solving this problem using dynamic programming.	**5**
Sch & Ans	Sch: 2 marks for problem analysis, 3 marks for maximization expression Ans: Designing a system that is composed of n devices connected in series so as to maximize the reliability of the system with a upper limit c of cost. r_i: reliability of device D_i functioning properly c_i: cost of device D_i m_i: number devices D_i that are arranged in duplicate Thus, the probability that all devices D_i with duplicate setting will malfunction is $(1-r_i)^{mi}$ Let reliability at i^{th} stage with m_i duplicate devices is given by $\varnothing_i(m_i)$ Then the problem of reliability is given is specified as Maximize $\Pi_{1\leq i\leq n}\ \varnothing_i(m_i)$ Subject to $\Sigma_{1\leq i\leq n}\ c_i m_i \leq C$ Where $m_i >= 1$ is an integer, and $1\leq i\leq n$	
(b)	**Calculate** the shortest distance between all pairs for the diagraph with the below weight matrix. $$\begin{bmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$$	**5**
Sch & Ans	Sch: 1 mark each for each of first 3 iteration (total 4 iterations), 2 marks for last iteration Ans:	



k↓	1	2	3	4	5
1	0	2	∞	1	8
2	6	0	3	2	∞
3	∞	∞	0	4	∞
4	∞	∞	2	0	3
5	3	∞	∞	∞	0

D(0)

k↓	1	2	3	4	5
1	0	2	∞	1	8
2	6	0	3	2	14
3	∞	∞	0	4	∞
4	∞	∞	2	0	3
5	3	5	∞	4	0

D(1)

k↓	1	2	3	4	5
1	0	2	5	1	8
2	6	0	3	2	14
3	∞	∞	0	4	∞
4	∞	∞	2	0	3
5	3	5	8	4	0

D(2)

k↓	1	2	3	4	5
1	0	2	5	1	8
2	6	0	3	2	14
3	∞	∞	0	4	∞
4	∞	∞	2	0	3
5	3	5	8	4	0

D(3)

k↓	1	2	3	4	5
1	0	2	3	1	4
2	6	0	3	2	5
3	∞	∞	0	4	7
4	∞	∞	2	0	3
5	3	5	6	4	0

D(4)

k↓	1	2	3	4	5
1	0	2	3	1	4
2	6	0	3	2	5
3	10	12	0	4	7
4	6	8	2	0	3
5	3	5	6	4	0

D(5)

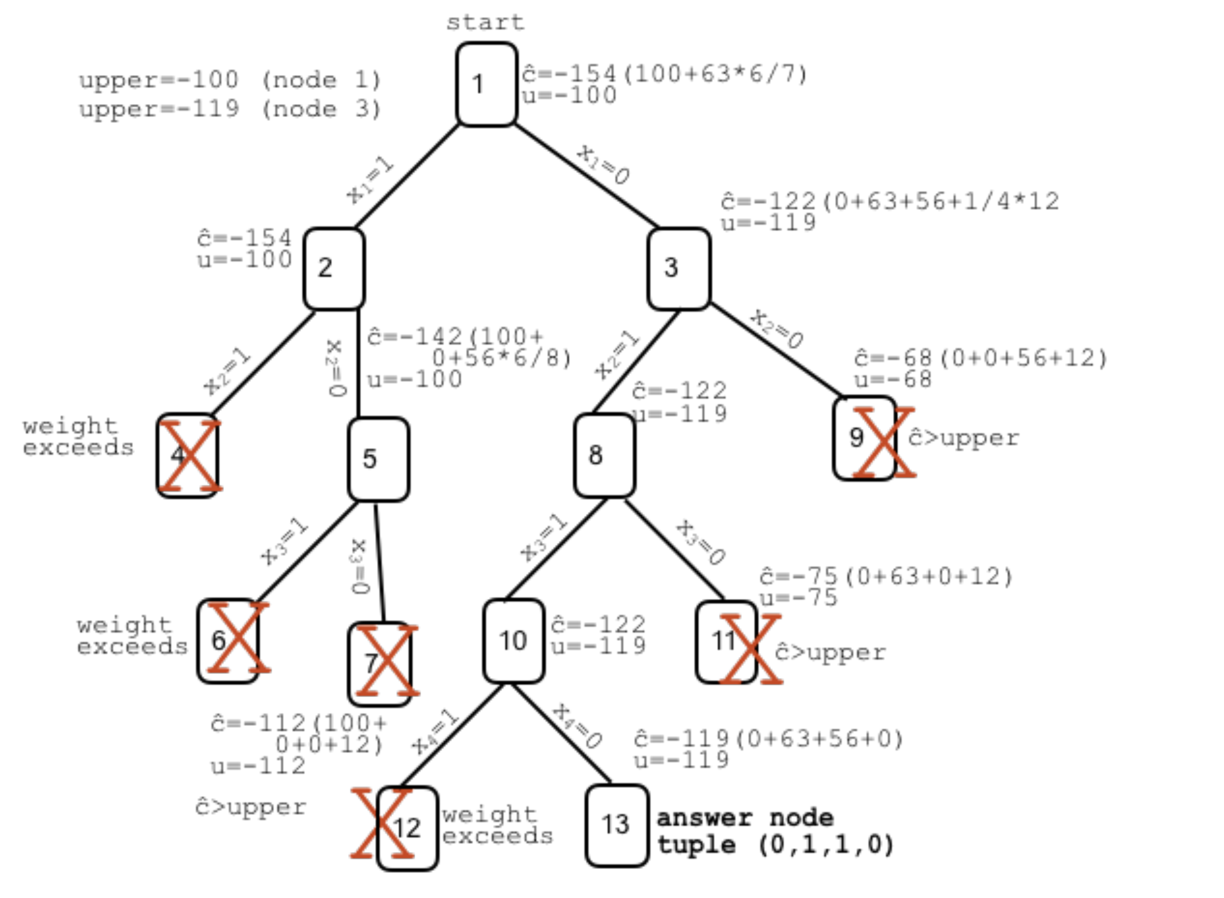
(c) **Apply** the branch and bound technique to solve the following knapsack problem with knapsack weight of 16, with weights and values for 4 items are given below. **5**

Item	Weight	Value
1	10	Rs 100
2	7	Rs 63
3	8	Rs 56
4	4	Rs 12

Sch & Ans

Sch: 1 mark each or each level of state space tree

Ans:



Signature of Course in charge **Signature of HOD-CSE**