



School of Computer Science and Engineering

Winter Semester 2024-25

CAT 2

SLOT: A1+TA1

Course Name & Code: Design and Analysis of Algorithms, BCSE204L

Class Number (s): ALL

Faculty Name (s): ALL.

Exam Duration: 90 Min.

Maximum Marks: 50

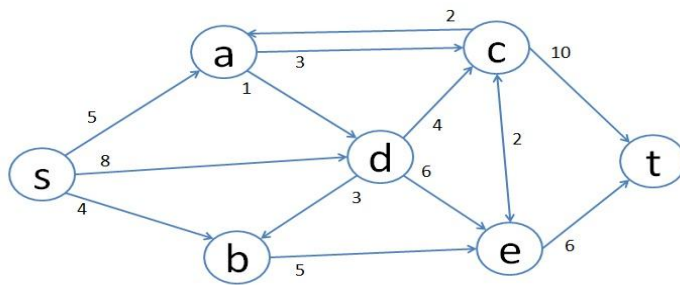
General instruction(s):

Open Book examination. Answer all the Questions.

Q.No.	Question	Max Marks	CO	BL																									
1.	<p>For the following job assignment problem, apply FIFO Branch and Bound (FIFOBB) and LIFO Branch and Bound (LIFOBB) separately and deduce the corresponding optimal assignments. The objective is to assign a unique pairing between a person and a job, such that the total time taken by all the persons to complete their assigned jobs is minimum. The rows correspond to the time taken by a person for each of the available jobs. Generate separate state space trees for FIFOBB and LIFOBB, devise an appropriate lower bound function and apply it for each of the states in the trees and finally deduce the optimal assignments.</p> <table border="1"> <thead> <tr> <th></th> <th>Job 1</th> <th>Job 2</th> <th>Job 3</th> <th>Job 4</th> </tr> </thead> <tbody> <tr> <td>Person A</td> <td>11</td> <td>9</td> <td>4</td> <td>10</td> </tr> <tr> <td>Person B</td> <td>9</td> <td>6</td> <td>5</td> <td>8</td> </tr> <tr> <td>Person C</td> <td>10</td> <td>7</td> <td>3</td> <td>10</td> </tr> <tr> <td>Person D</td> <td>8</td> <td>6</td> <td>11</td> <td>9</td> </tr> </tbody> </table> <p>Ans. Branch and Bound Lower bound (LB): take the minimum from each row. By FIFO method : Level 0 Root node <u>(1)</u> Start, LB=4+5+3+6=18</p> <p>Level 1 Allocation for Person A (4 nodes) <u>(2)</u> A=J1 → LB=11+5+3+6=25 <u>(3)</u> A=J2 → LB=9+5+3+8=25 <u>(4)</u> A=J3 → LB=4+6+7+6=23 <u>(5)</u> A=J4 → LB=10+5+3+6=24</p> <p>Level 2 Allocation for Person B (4 * 3 =) 12 nodes Under <u>(2)</u> → <u>(6)</u> B=J2 → LB=11+6+3+9=29 Under <u>(2)</u> → <u>(7)</u> B=J3 → LB=11+5+7+6=29</p>		Job 1	Job 2	Job 3	Job 4	Person A	11	9	4	10	Person B	9	6	5	8	Person C	10	7	3	10	Person D	8	6	11	9	10	CO2	BL3
	Job 1	Job 2	Job 3	Job 4																									
Person A	11	9	4	10																									
Person B	9	6	5	8																									
Person C	10	7	3	10																									
Person D	8	6	11	9																									

	<p>Under <u>(2)</u> → <u>(8)</u> B=J4 → LB=11+8+3+6=28</p> <p>Under <u>(3)</u> → <u>(9)</u> B=J1 → LB=9+9+3+9=30</p> <p>Under <u>(3)</u> → <u>(10)</u> B=J3 → LB=9+5+10+8=32</p> <p>Under <u>(3)</u> → <u>(11)</u> B=J4 → LB=9+8+3+8=28</p> <p>Under <u>(4)</u> → <u>(12)</u> B=J1 → LB=4+9+7+6=26</p> <p>Under <u>(4)</u> → <u>(13)</u> B=J2 → LB=4+6+10+8=28</p> <p>Under <u>(4)</u> → <u>(14)</u> B=J4 → LB=4+8+7+6=25</p> <p>Under <u>(5)</u> → <u>(15)</u> B=J1 → LB=10+9+3+6=28</p> <p>Under <u>(5)</u> → <u>(16)</u> B=J2 → LB=10+6+3+8=29</p> <p>Under <u>(5)</u> → <u>(17)</u> B=J3 → LB=10+5+7+6=28</p> <p>Similarly, Level 3 allocation for Person C (12*2=) 24 nodes</p> <p>And, Level 4 allocation for Person D (24*1 =) 24 nodes</p> <p>Pruning/killing of nodes start here, and the pruning happens at both Level 3 and Level 4</p> <p>Answer : A=J3, B=J4, C=J2, D=J1, Optimal cost = 27</p> <p>For LIFO, the same nodes are created and the same optimal answer is generated, but the order of creation changes, and more pruning occurs.</p>			
2.	<p>Apply a string matching algorithm (which applies hashing to reduce the time complexity of operations) to find out the shift locations where the pattern (P) matches with the text (T).</p> <p>T="ABDEFACBDE" P="ACE"</p> <p>Demonstrate the preprocessing operations as well as the actual shifting/matching operations. Deduce the locations of spurious hits (if any) as well as the actual matches. Use q=11 for the modulo operation, and an octal number system to represent the alphabet in both T and P.</p> <p>Ans. Rabin Karp algorithm</p> <p>T="ABDEFACBDE", P="ACE"</p> <p>Mapping- A=0, B=1,.....E=4, F=5</p> <p>T="01345024134", P="024", n=11, m=3, d=8, q=11, h=8² mod 11=9</p> <p>Preprocessing</p> <p>i=1 to 3, p=0, t0=0</p> <p>p=(d*p+P[i])mod 11, t0=(d*t0 + T[i]) mod 11</p> <p>p=(8*0+0)mod 11=0 → (8*0+2)mod 11=2 → (8*2+4)mod 11 = 9</p> <p>t0=(8*0+0)mod 11=0 → (8*0+1)mod 11=1 → (8*1+3)mod 11 = 0</p> <p>Processing part</p> <p>t(s+1)= (d (t(s) - T[s+1]*h) + T[s+m+1]) mod 11</p> <p>t1= (8(0-0*9)+4)mod 11=4</p>	10	CO3	BL3

	<p>$t_2 = (8(4-1*9)+5) \bmod 11 = 9$, $p=9$ but spurious hit $t_3 = (8(9-3*9)+0) \bmod 11 = 10$, Similarly, $t_4=3$, $t_5=9$ (actual match), $t_6=7$, $t_7=3$, $t_8=4$</p>																																																																																							
3.	<p>A network having six nodes (SABCDE), with its set of weighted edges will be provided. The objective is to find the shortest path from S to reach all the other nodes A, B, C, D and E, using an algorithm which can work with negative weighted edges. The solution should disclose the shortest distances from S to all other nodes AND the corresponding paths through which this is possible. Generate the solution by showing all the necessary intermediate steps. The order/sequence of edges and their corresponding weights are : $A \rightarrow B$ (10), $B \rightarrow C$ (30), $A \rightarrow C$ (50), $A \rightarrow D$ (30), $C \rightarrow D$ (-100), $D \rightarrow E$ (20), $S \rightarrow A$ (5), $S \rightarrow D$ (30), $E \rightarrow S$ (30)</p> <p>Ans. Bellman-Ford algorithm for single source shortest path</p> <p>Initialization</p> <table border="1"> <tr> <td></td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> </tr> <tr> <td>d/cost</td> <td>0</td> <td>Inf</td> <td>Inf</td> <td>Inf</td> <td>Inf</td> <td>Inf</td> </tr> <tr> <td>P</td> <td>Nil</td> <td>Nil</td> <td>Nil</td> <td>Nil</td> <td>Nil</td> <td>Nil</td> </tr> </table> <p>Relaxing the edges given in the same order as given in the question.</p> <p>After round 1</p> <table border="1"> <tr> <td></td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> </tr> <tr> <td>d/cost</td> <td>0</td> <td>5</td> <td>Inf</td> <td>Inf</td> <td>30</td> <td>Inf</td> </tr> <tr> <td>P</td> <td>Nil</td> <td>S</td> <td>Nil</td> <td>Nil</td> <td>S</td> <td>Nil</td> </tr> </table> <p>After Round 2</p> <table border="1"> <tr> <td></td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> </tr> <tr> <td>d/cost</td> <td>-5</td> <td>5</td> <td>15</td> <td>45</td> <td>-55</td> <td>-35</td> </tr> <tr> <td>P</td> <td>E</td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> </tr> </table> <p>And so on...</p> <p>After Round 5</p> <table border="1"> <tr> <td></td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> </tr> <tr> <td>d/cost</td> <td>-10</td> <td>-5</td> <td>10</td> <td>40</td> <td>-60</td> <td>-40</td> </tr> <tr> <td>P</td> <td>E</td> <td>S</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> </tr> </table> <p>For Round 6, change occurs at node B (10 becomes 5) Hence Bellman Ford outputs 'not possible' as there is a negative edge cycle.</p>		S	A	B	C	D	E	d/cost	0	Inf	Inf	Inf	Inf	Inf	P	Nil	Nil	Nil	Nil	Nil	Nil		S	A	B	C	D	E	d/cost	0	5	Inf	Inf	30	Inf	P	Nil	S	Nil	Nil	S	Nil		S	A	B	C	D	E	d/cost	-5	5	15	45	-55	-35	P	E	S	A	B	C	D		S	A	B	C	D	E	d/cost	-10	-5	10	40	-60	-40	P	E	S	A	B	C	D	10	CO3	BL2
	S	A	B	C	D	E																																																																																		
d/cost	0	Inf	Inf	Inf	Inf	Inf																																																																																		
P	Nil	Nil	Nil	Nil	Nil	Nil																																																																																		
	S	A	B	C	D	E																																																																																		
d/cost	0	5	Inf	Inf	30	Inf																																																																																		
P	Nil	S	Nil	Nil	S	Nil																																																																																		
	S	A	B	C	D	E																																																																																		
d/cost	-5	5	15	45	-55	-35																																																																																		
P	E	S	A	B	C	D																																																																																		
	S	A	B	C	D	E																																																																																		
d/cost	-10	-5	10	40	-60	-40																																																																																		
P	E	S	A	B	C	D																																																																																		
4.	<p>Deduce the maximum flow for the network given below by using an approach which allows an intermediate node to store the excess flow locally.</p>	10	CO3	BL3																																																																																				



The flow starts from the source, s, ends in the sink, t, and the values on the edges represent the corresponding edge capacities.

Ans.

Max Flow deduction by Push Relabel approach

Initialization

	s	a	b	c	d	e	t
u.e	0/Inf	5	4	0	8	0	0
u.h	7	0	0	0	0	0	0

After relabel a (h=1), push from a to c and d, relabel a 7 times to a.h=8, push from a to s

	s	a	b	c	d	e	t
u.e	0/Inf	0	4	3	9	0	0
u.h	7	8	0	0	0	0	0

Relabel c (h=1), push from c to t

	s	a	b	c	d	e	t
u.e	0/Inf	0	4	0	9	0	3
u.h	7	8	0	1	0	0	0

Relabel b (h=1), push from b to e

Relabel e (h=1), push from e to t

	s	a	b	c	d	e	t
u.e	0/Inf	0	0	0	9	0	7
u.h	7	8	1	1	0	1	0

Relabel d (h=2), push from d to b, c, e

Relabel c, e (h=2), push from c, e to t

	s	a	b	c	d	e	t
u.e	0/Inf	0	3	0	0	0	13
u.h	7	8	1	2	2	2	0

And so on ...

Total u.e deposited at t would be 15, and all the internal nodes e to be zero.

5. Starting and end points of six line segments are given below. The objective is to find out if any of the line segments intersect or not by applying an algorithm which sweeps the xy coordinate space from left

<p>to right by a vertical line. Execution of the necessary intermediate steps is required.</p> <p>Line segment A (3,4) to (9,5), B (5,9) to (23,9), C (7,9) to (19,15), D (22, 52) to (25,54), E (14,6) to (29,14) F(10,11) to (15,7)</p> <p>Ans. Line sweep technique There will be 12 event points, and they are : X=3 (Line A start), X=5 (Line B start), X=7 (Line C start), X=9 (Line A end), X=10 (Line F start), X=14 (Line E start), X=15 (Line F end), X=19 (Line C end), X=22 (Line D start), X=23 (Line B end), X=25 (Line D end), X=29 (Line E end).</p> <p>1st event point (x=3) only line A enters</p> <p>2nd Event point (x=5), line B enters, line B > line A. Does line B (5,9 to 23,9) and line A (3,4 to 9,5) intersect ? No. (Use two line segment intersection algorithm (time complexity O(constant)) to get the answer).</p> <p>3rd event point (x=7), line C enters, C>B, does C & B intersect? Yes (Use two line segment intersection algorithm (time complexity O(constant)) to get the answer).</p>			
--	--	--	--