VIT
Vellore Institute of Technology

School of Computer Science and Engineering

Winter Semester 2023-2024

Continuous Assessment Test – II
SLOT: A2+TA2

Programme Name & Branch : B.Tech (BCB/BCE/BCI/BCT/BDS/BKT)

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

Class Number (s): ALL

Faculty Name (s): ALL

Exam Duration: 90 Min. **Maximum Marks: 50**

Answer All

<table>
<tr><th>Q.No.</th><th>Question</th><th>Max Marks</th></tr>
<tr><td>1.</td><td>A bank has 14 million dollars, which can be invested into stocks of four companies (1, 2, 3, and 4). The following table shows the net revenue of each company and the amount that must be invested into each company.
<table>
<tr><td>Company</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Revenue (Million $)</td><td>16</td><td>22</td><td>12</td><td>8</td></tr>
<tr><td>Investment Amount (Million $)</td><td>5</td><td>7</td><td>4</td><td>3</td></tr>
</table>
The objective for the bank is to select a set of companies for investment, so as to maximize the total revenue with the condition that no partial investment can be done i.e., for each company we can either invest into it or not. Solve the problem with the algorithm whose exponential worst-case time complexity can be improved by employing better techniques for efficient pruning.</td><td>10</td></tr>
<tr><td>2.</td><td>i) Find out the longest proper prefix that is also the proper suffix for the pattern “ababa”.
ii) Calculate the LPS table of the pattern “ababaa”.

Using the table constructed in (ii), answer the following.
iii) What is the LPS value for the 4th character ‘b’? What does that value infer?
iv) Let ‘i’ and ‘j’ be used to denote the index of the characters to be compared in text and pattern respectively. For the text “abababababaabb”, if the comparison starts from the beginning of the text and if all the characters of the pattern matched with those of the text, what will be the new values of i and j, to proceed with the next comparison?
v) For the text “abababababaabb”, if all the characters of the pattern except the last, matched with those of the text, what will be the new values of i and j, to proceed with the next comparison?</td><td>10</td></tr>
<tr><td>3.</td><td>A)
RELAX(u, v, w)
1 if $v.d > u.d + w(u, v)$
2 $v.d = u.d + w(u, v)$

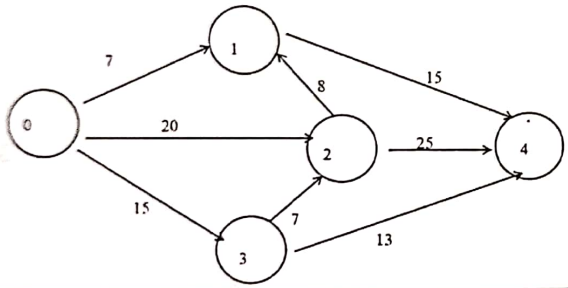
BELLMAN-FORD(G, w, s)
1 INITIALIZE-SINGLE-SOURCE(G, s)
2 for $i = 1$ to $|G.V| - 1$
3 for each edge $(u, v) \in G.E$
4 RELAX(u, v, w)
5 for each edge $(u, v) \in G.E$
6 if $v.d > u.d + w(u, v)$
7 return FALSE
8 return TRUE

In the above Bellman-Ford algorithm, make the required modifications to merge the two loops in line numbers 2 and 5 into a single loop and to implement early termination of the loop, if no edges are relaxed in a particular iteration.

B) The capacity matrix for a flow network is given below. The first augmenting path found using Edmonds-Karp algorithm for the graph is 0-1-2-4-5. Is this correct or not? Justify your answer.</td><td>5</td></tr>
</table>

		5

	0	1	2	3	4	5
0	0	16	13	0	0	0
1	0	0	0	12	0	0
2	0	0	0	0	14	0
3	0	0	9	0	0	20
4	0	0	0	7	0	4
5	0	0	0	0	0	0

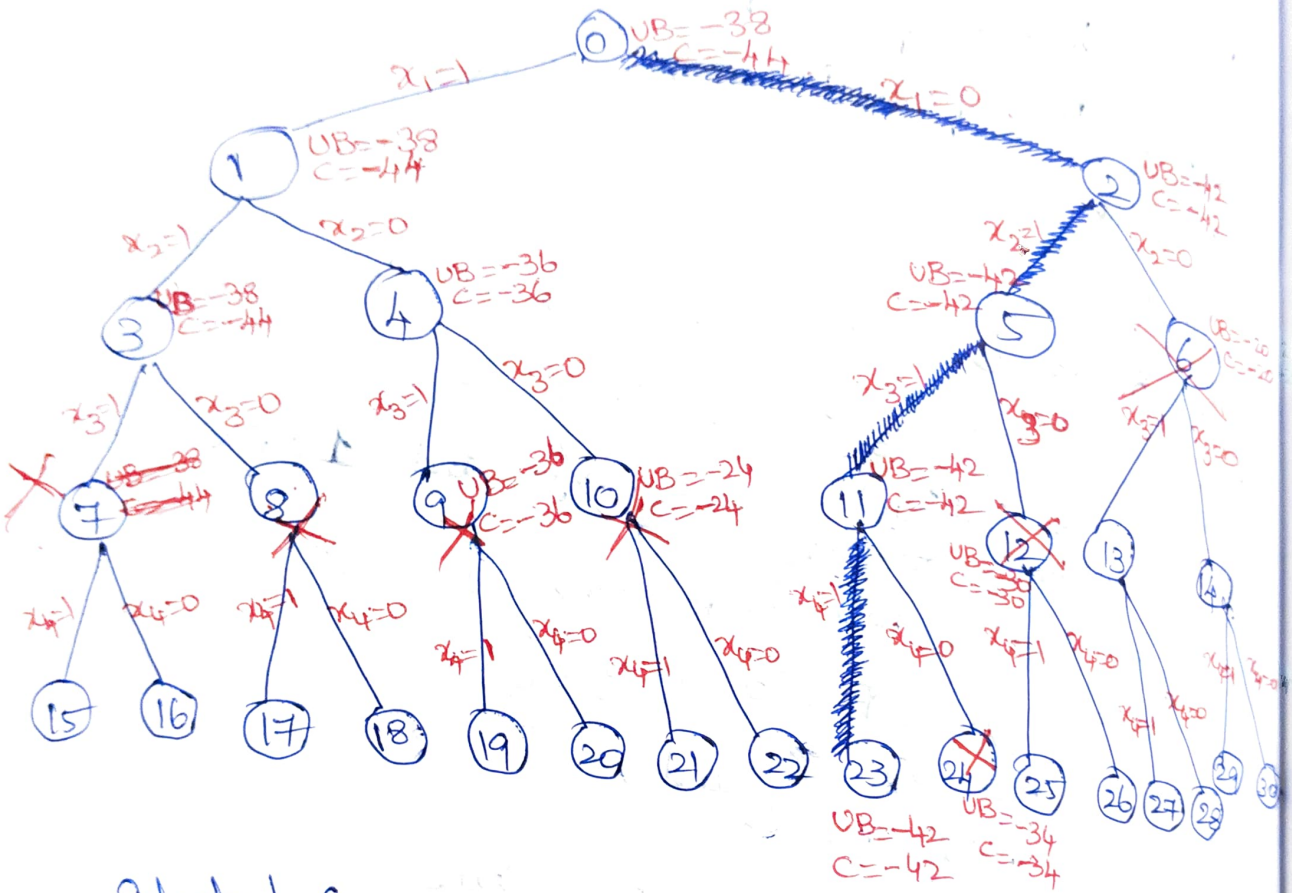
4.	Apply Push-Relabel algorithm for the flow network given below to find the maximum flow that can pass from the source to sink.  	10
5.	Two line-segments intersect if and only if either (or both) of the following conditions holds: 1. Each segment straddles the line containing the other. 2. An endpoint of one segment lies on the other segment. (This condition comes from the boundary case.) SEGMENTS-INTERSECT(p_1, p_2, p_3, p_4) 1 d_1 = DIRECTION(p_3, p_4, p_1) 2 d_2 = DIRECTION(p_3, p_4, p_2) 3 d_3 = DIRECTION(p_1, p_2, p_3) 4 d_4 = DIRECTION(p_1, p_2, p_4) 5 if (($d_1 > 0$ and $d_2 < 0$) or ($d_1 < 0$ and $d_2 > 0$)) and (($d_3 > 0$ and $d_4 < 0$) or ($d_3 < 0$ and $d_4 > 0$)) 6 return TRUE 7 elseif $d_1 == 0$ and ON-SEGMENT(p_3, p_4, p_1) 8 return TRUE 9 elseif $d_2 == 0$ and ON-SEGMENT(p_3, p_4, p_2) 10 return TRUE 11 elseif $d_3 == 0$ and ON-SEGMENT(p_1, p_2, p_3) 12 return TRUE 13 elseif $d_4 == 0$ and ON-SEGMENT(p_1, p_2, p_4) 14 return TRUE 15 else return FALSE DIRECTION(p_i, p_j, p_k) 1 return $(p_k - p_i) \times (p_j - p_i)$ ON-SEGMENT(p_i, p_j, p_k) 1 if $\min(x_i, x_j) \le x_k \le \max(x_i, x_j)$ and $\min(y_i, y_j) \le y_k \le \max(y_i, y_j)$ 2 return TRUE 3 else return FALSE	

	Given a line-segment L1 with two end-points (2,1) and (6,5), select the other appropriate line-segment among the following to justify your answer for the statements (A), (B) and (C) given below. L2 with two end-points (3,5) and (5,2) L3 with two end-points (3,5) and (4,4) L4 with two end-points (7,6) and (8,7) L5 with two end-points (7,6) and (5,8) L6 with two end-points (5,4) and (7,3)	
	Statement A: The condition in line number 5 of the algorithm is "if ((d1 > 0 and d2 < 0) or (d1 < 0 and d2 > 0)) **and** ((d3 > 0 and d4 < 0) or (d3 < 0 and d4 > 0)) return TRUE" Can the "and" operator which is underlined and highlighted in bold, be replaced with "or" operator to prove that one segment straddles the line containing the other?	3
	Statement B: To prove intersection, is it a must to check the condition "ON_SEGMENT(P_i, P_j, P_k)" in line numbers 7 to 14 of the algorithm?	3
	Statement C: The else-if conditions in line numbers 7 to 14 of the algorithm show that any one of the directions d1, d2, d3, d4 can be zero to prove intersection. Is there any line-segment in the list for which only one direction is zero and does not intersect with L1? Is there any line-segment in the list for which more than one direction is zero and still does not intersect with L1?	4

A2 - CAT-II - Key

1) B & B:

UB = ~~-38~~ -42



Selected Companies: 2, 3, 4.

Total revenue: 22 + 12 + 8 = 42 million $

2) i) ababa

Proper prefix	Proper suffix
a	a
ab	ba
aba	aba
abab	baba

The longest proper prefix that is also the proper suffix is "aba"

(ii) LPS table for "ababaa"

0	1	2	3	4	5
a	b	a	b	a	a
0	0	1	2	3	0

(iii) LPS value of "b" = 2.

It means that the longest proper prefix - suffix for the substring "abab" is "ab" whose length is stored in this cell.

(iv)

	0	1	2	3	4	5	6	7	8	9	10	11	12
Text	a	b	a	b	a	a	b	a	b	a	a	b	b
Pattern	a	b	a	b	a	a							

If array indexing starts from '0' after all the characters of the pattern match with the text, i = 6 and j = 0 to start comparing the encircled letter 'b' in text and 'a' in pattern.

(v)

	0	1	2	3	4	5	6	7	8	9	10	11
text	a	b	a	b	a	b	a	b	a	a	b	b
pattern	a	b	a	b	a	a						

5th characters do not match.

∴ j = LPS[5-1]

= LPS[4] = 3

So, i = 5, j = 3

	0	1	2	3	4	5					
	a	b	a	b	a	b	a	b	a	a	b	b
pattern			a	b	a	b	a	a				

Characters 'b' in text & pattern will be compared.

3) a)

```
boolean RELAX(u, v, w)
    if v.d > u.d + w(u, v)
        v.d = u.d + w(u, v)
        return true
    return false

BELLMAN-FORD(G, w, s)
    INITIALIZE-SINGLE-SOURCE(G, s)
    boolean cycle = false
    for i = 1 to |G.V|
        Count = 0
        for each edge (u, v) ∈ G.E
            if relax(u, v, w)
                if (i == V)
                    cycle = true
                    break
                else
                    count = Count + 1
        if (Count == 0)
            return true
    if (cycle == true)
        return false
    else
        return true
```

b) Edmonds-Karp:

First augmenting path:

Queue:

~~5~~	~~4~~	~~3~~	~~2~~	~~1~~	~~0~~

Node id	0	1	2	3	4	5
visit state	1	1	1	1	1	1
Predecessor	-1	0	0	1	2	3

Aug. path = $5 \leftarrow 3 \leftarrow 1 \leftarrow 0$

$\Rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 5$.

So, 0-1-2-4-5 is not the correct augmenting path.

4) Push-Relabel:

Node Id	0	1	2	3	4
Height	5	~~1~~	~~0~~ 1	~~1~~ 2	0
Excess Flow	-	~~7~~ 0	~~20~~ 0	~~15~~ 2 ~~0~~	~~0~~ 7 ~~42~~ ~~40~~ ~~24~~

Capacity

	0	1	2	3	4
0	0	7	20	15	0
1	0	0	0	0	15
2	0	8	0	0	25
3	0	0	7	0	13
4	0	0	0	0	0

Flow

	0	1	2	3	4
0	0	~~7~~	~~20~~	~~15~~	0
1	~~7~~	0	0	0	~~7~~
2	~~20~~	0	0	~~2~~	~~22~~ ~~20~~
3	~~15~~	0	~~2~~	0	~~13~~
4	0	~~7~~	~~20~~ ~~22~~	~~0~~	~~13~~

After preflow:

i) Excess flow node = 1;

Relabel node 1 $\Rightarrow$ Ht(1) = 0+1 = 1

Flow from 1 to 4 = min(7, 15) = 7 ✓

ii) Excess flow node = 2

Relabel node 2 $\Rightarrow$ Ht(2) = 0+1 = 1

Flow from 2 to 4 = min(20, 25) = 20 ✓

iii) Excess flow node = 3

Relabel node 3 $\Rightarrow$ Ht(3) = 0+1 = 1

Flow from 3 to 4 = min(15, 13) = 13 ✓

iv) Excess flow node = 3

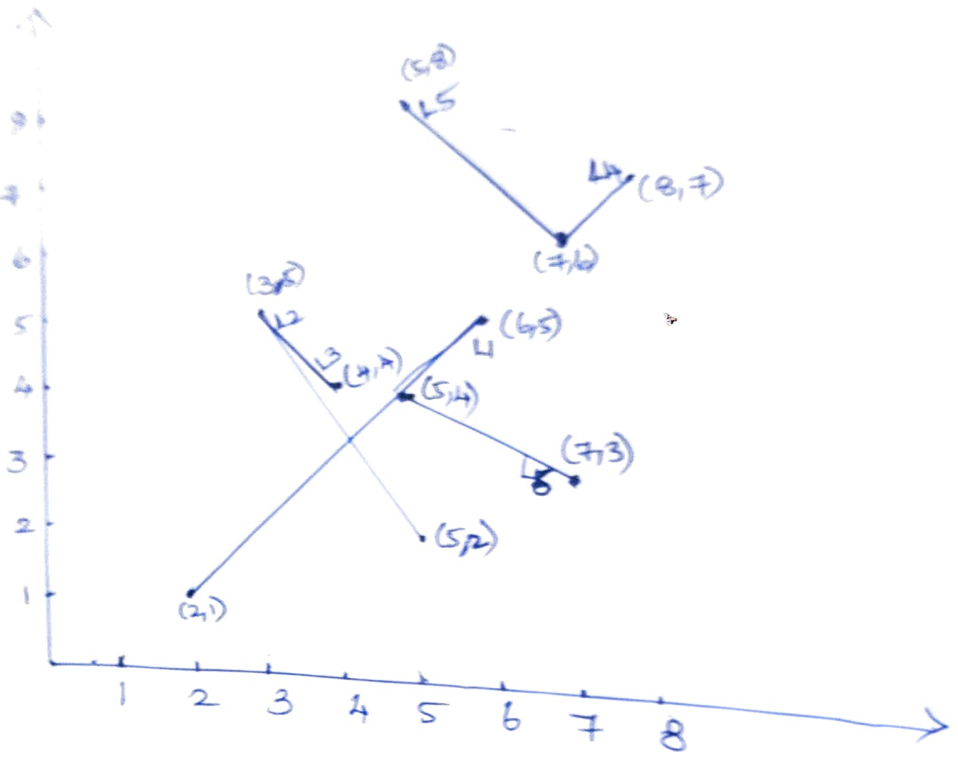
Relabel node 3 $\Rightarrow$ Ht(3) = 1+1 = 2.

Flow from 3 to 2 = min(7, 2) = 2

v) Excess flow node = 2

Flow from 2 to 4 = min(2, 25-20) = 2 ✓

No more excess flow nodes. Maxflow = 7+20+13+2 = 42



Line segment	$d_1 = \text{dir}(P_3, P_4, P_1)$	$d_2 = \text{dir}(P_3, P_4, P_2)$	$d_3 = \text{dir}(P_1, P_2, P_3)$	$d_4 = \text{dir}(P_1, P_2, P_4)$
L2 (3,5), (5,2)	$\begin{vmatrix} -1 & -2 \\ -4 & -3 \end{vmatrix} = 11$	$\begin{vmatrix} 3 & 2 \\ 0 & -3 \end{vmatrix} = -9$	$d_3 = \begin{vmatrix} 1 & 4 \\ 4 & 4 \end{vmatrix} = -12$	$\begin{vmatrix} 3 & 4 \\ 1 & 4 \end{vmatrix} = 8$
L3 (3,5), (4,4)	$\begin{vmatrix} -1 & 1 \\ -4 & -1 \end{vmatrix} = 5$	$\begin{vmatrix} 3 & 1 \\ 0 & -1 \end{vmatrix} = -3$	$d_3 = \begin{vmatrix} 1 & 4 \\ 4 & 4 \end{vmatrix} = -12$	$\begin{vmatrix} 2 & 4 \\ 3 & 4 \end{vmatrix} = 4$
L4 (7,6), (8,7)	$\begin{vmatrix} -5 & 1 \\ -5 & 1 \end{vmatrix} = 0$	$\begin{vmatrix} -1 & 1 \\ -1 & 1 \end{vmatrix} = 0$	$\begin{vmatrix} 5 & 4 \\ 5 & 4 \end{vmatrix} = 0$	$\begin{vmatrix} 6 & 4 \\ 6 & 4 \end{vmatrix} = 0$
L5 (7,6) (5,8)	$\begin{vmatrix} -5 & -2 \\ -5 & 2 \end{vmatrix} = -20$	$\begin{vmatrix} -1 & -2 \\ -1 & 2 \end{vmatrix} = 4$	$\begin{vmatrix} 5 & 4 \\ 5 & 4 \end{vmatrix} = 0$	$\begin{vmatrix} 3 & 4 \\ 7 & 4 \end{vmatrix} = -16$
L6 (5,4), (7,3)	$\begin{vmatrix} -3 & 2 \\ -3 & -1 \end{vmatrix} = 9$	$\begin{vmatrix} 1 & 2 \\ 1 & -1 \end{vmatrix} = -3$	$\begin{vmatrix} 3 & 4 \\ 3 & 4 \end{vmatrix} = 0$	$\begin{vmatrix} 5 & 4 \\ 2 & 4 \end{vmatrix} = 12$

Statement A:

If "and" is replaced by "or": according to the cross product, for L3, $d_1 > 0$ and $d_2 < 0$ but d_3 and d_4 are both negative; But still because of "or" L3 will be reported as intersecting with L1. From figure, it can be seen that this result contradicts.

Statement B:

For L5, $d_3 = 0$. Lets check onsegment (P_1, P_2, P_3).

onsegment(P_1, P_2, P_3): $P_3 = (7,6)$, $P_1 = (2,1)$, $P_2 = (6,5)$

~~$2 < 7$ and~~ onsegment $\Rightarrow$ false $\quad 2 < 7 \not< 6$ and $1 < 6 \not< 5$

For L6, $d_3 = 0$. Lets check onsegment (P_1, P_2, P_3)

$P_3 = (5,4)$, $P_1 = (2,1)$, $P_2 = (6,5)$

$2 < 5 < 6$ ✓ and

$1 < 4 < 5$ ✓

$\therefore$ Onsegment (P_1, P_2, P_3) = true.

From figure, it can be seen that L5 does not intersect with L1 but L6 intersects, although in both $d_3 = 0$. This proves that "onsegment" condition is a must.

Statement C:

For L5, $d_3 = 0$ but still does not intersect with L1 as onsegment(P_1, P_2, P_3) is false.

For L4, all direction values are zeroes.

onsegment(P_3, P_4, P_1): $7 \not< 2 < 8 \Rightarrow$ false

onsegment(P_3, P_4, P_2): $7 \not< 6 < 8 \Rightarrow$ false

onsegment(P_1, P_2, P_3): $2 < 7 \not< 6 \Rightarrow$ false

onsegment(P_1, P_2, P_4): $2 < 8 \not< 6 \Rightarrow$ false

$\therefore$ L4 also does not intersect with L1

—o—