Q1

```
void find_longest(int A[], int n){
    int start_t = -1, end_t = -1, length_t = 0;
    int start_f = -1, end_f = -1, length_f = 0;

    int start_t = 0, end_t = 0, length_t = 1;

    for(int i=1; i<n; i++){
        if(A[i] >= A[end_t]){
            end_t++;
            length_t++;
        } else {
            if(length_t > length_f){
                start_f = start_t;
                end_f = end_t;
                length_f = length_t;
            }
            start_t = i;
            end_t = i;
            length_t = 1;
        }
    }
    if(length_t > length_f){
        start_f = start_t;
        end_f = end_t;
        length_f = length_t;
    }
    cout << start_f << "#" << end_f << "#" << length_f << endl;
}
```

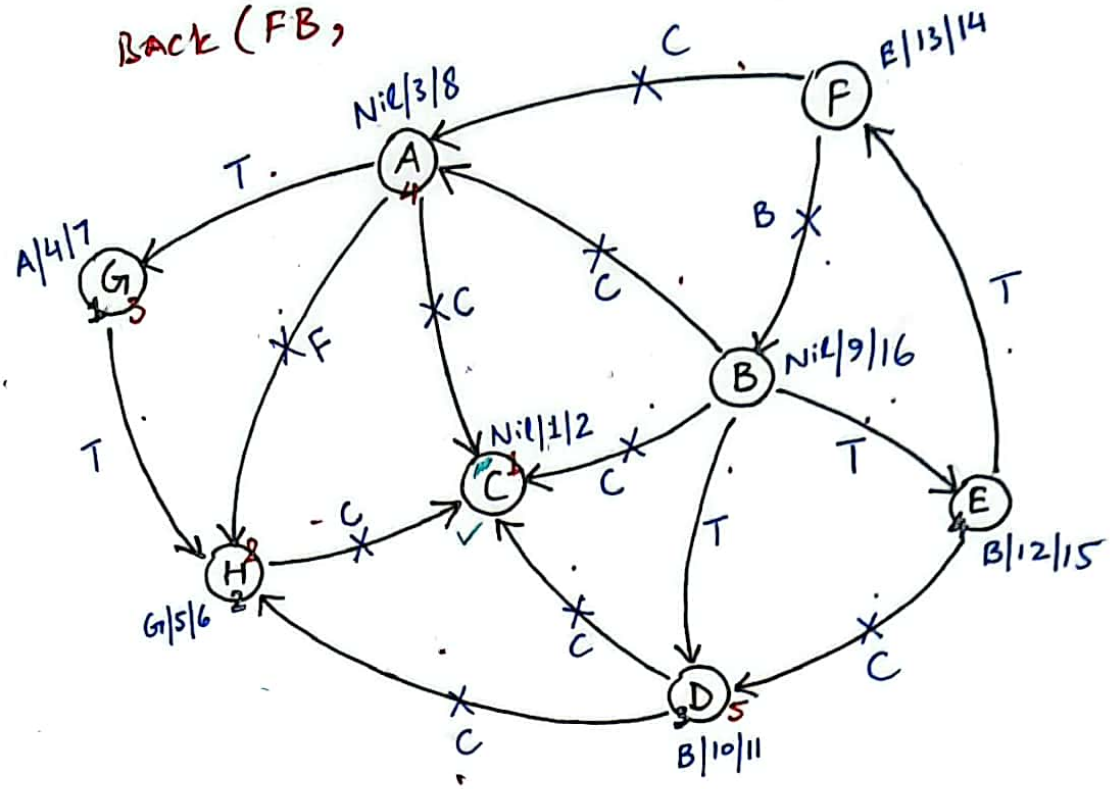
Q-2

Tree (AG, GH, BD, BE, EF,

Forward (AH,

Cross (AC, BA, BC, DC, DH, ED, FA,

Back (FB,



* When DFS-visit called → vertex parent is changed, but color/time not

b =	A	B	C	D	E	F	G	H
	Nil	Nil	Nil	B	B	Nil	A	G
	B	G	B	B	W	W	B	B
	3	9	1	10	-	-	4	5
	8	-	2	11	-	-	7	6

* When line 8 → color B, but t time not assigned.

C =	A	B	C	D	E	F	G	H
	Nil	Nil	Nil	B	Nil	Nil	A	G
	B	G	B	B	W	W	B	B
	3	9	1	10	-	-	4	5
	8	-	2	-	-	-	7	6

Design & Analysis of Algorithm
Final Examination (Fall 2022)

Reg. No: _____ Time: 150 mins

Q1: (20 Marks)

Write a proper function that will find **longest sub sequence** in an **array A of size N,** where array has no particular values sequence. [To be called as sub sequence → every next value can be greater or equal than the last value, otherwise it stops]. This sub sequence length can vary between [1 to N]. This function should work properly, syntax, semantics and logic.
You have to show two Indexes [starting and ending] and the length of the subsequence at the end of the function. [Array index will start from 0]. Time complexity of the algorithm shall be **O(N)**. You have no other extra array available for this process. **Example:**

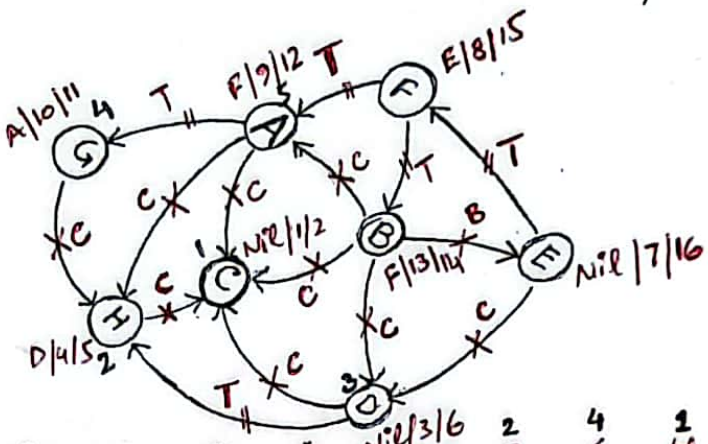
Input: 11, 6, 21, 21, 15, 16, 16, 17, 12, 88
Output: 1 # 7 # 7
Input: 11, 6, 21, 11, 15, 16, 12, 11, 12, 15
Output: 3 # 5 # 3

Q2: (3+6+6 Marks)

```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)

DFS-VISIT(G, u)
1  time = time + 1
2  u.d = time
3  u.color = GRAY
4  for each v ∈ G.Adj[u]
5      if v.color == WHITE
6          v.π = u
7          DFS-VISIT(G, v)
8  u.color = BLACK
9  time = time + 1
10 u.f = time
```

A)



	A	B	C	D	E	F	G	H
	F	F	Nil	Nil	Nil	E	A	D
	9	13	1	3	7	8	10	4
	12	14	2	6	16	15	11	5

Apply DFS algorithm on the given graph and answer the following questions. The source node is 'C'. (**Note:** The adjacent nodes of a vertex are to be traversed in alphabetical order, and DFS algorithm will also work in alphabetical order.)

a) Write the type of each edge of the above graph.
b) Write updated details for all the vertices when DFS-VISIT (G, u) [7th line inside the DFS-VISIT] called for the 4th time (Just Called) write the values for all vertices ('Parent', 'color', 'd-time', 'f-time').
c) Write down details for all the vertices when Line 8 of DFS-VISIT (G, u) completed its work for 5th time - write the values of their ('Parent', 'color', 'd-time', 'f-time').

b)

	A	B	C	D	E	F	G	H
	F F	Nil ✓	Nil ✓	Nil ✓	Nil ✓	E ✓	A A	D ✓
	G B	W ✓	B ✓	B ✓	G ✓	G ✓	W B	B ✓
	9 9	- ✓	1 ✓	3 ✓	7 ✓	8 ✓	- 10	4 ✓
	- -	- ✓	2 ✓	6 ✓	- ✓	- ✓	- 11	5 ✓

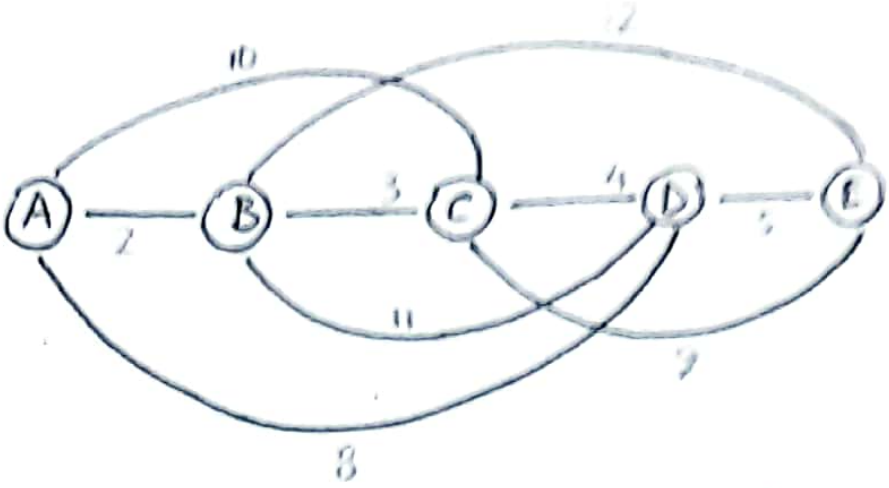
4) c) LHR(5|MUL - 4|SAH) - QUE(10|MUL - 6|LHR).

	MUL	LHR	SAH	QUE	SAW	PES	ISL	MUR	NAR
	0|Nil	∝|Nil	∝|Nil	∝|Nil	∝|Nil	∝|Nil	∝|Nil	∝|Nil	∝|Nil
MUL(0|Nil) I, L, Q, S	X	5|MUL	2|MUL	10|MUL	∝|Nil	∝|Nil	3|MUL	∝|Nil	∝|Nil
SAH(2|MUL) I, L	X	4|SAH	X	10|MUL	∝|Nil	∝|Nil	3|MUL	∝|Nil	∝|Nil
ISL(3|MUL) M	X	4|SAH	X	10|MUL	∝|Nil	∝|Nil	X	5|ISL	∝|Nil
LHR(4|SAH) I, M, P, Q, S*	X	X	X	6|LHR	6|LHR*	7|LHR	X	5|ISL	∝|Nil
MUR(5|ISL) N	X	X	X	6|LHR	6|LHR	7|LHR	X	X	7|MUR
QUE(6|LHR) S	X	X	X	X	6|LHR	7|LHR	X	X	7|MUR
SAW(6|LHR) N, P	X	X	X	X	X	7|LHR[4]	X	X	7|MUR[5]
PES(7|LHR) M, N	X	X	X	X	X	X	X	X	7|MUR
NAR(7|LHR) -	X	X	X	X	X	X	X	X	X

5) a) ISL(3|MUL), LHR(5|MUL), QUE(10|MUL), SAH(2|MUL), LHR(4|SAH), MUR(5|ISL), PES(7|LHR), QUE(6|LHR) SAW(6|LHR)

5) b) line 8 has been executed → check done → parent NAR changed for last execution.

0|Nil 4|SAH 2|MUL 6|LHR ∝|Nil 7|LHR 3|MUL 5|ISL ∝|Nil



Q-4

	A	B 1	C 2	D 2	E 3
	0/Nil	∞/Nil	∞/Nil	∞/Nil	∞/Nil
A(0/Nil) B, C, D	X	2/A	10/A	8/A	∞/Nil
B(2/A) C, D, E	X	X	3/B	8/A	12/B
C(3/B) D, E	X	X	X	4/C	9/C
D(4/C) E	X	X	X	X	5/D
E(5/D)	X	X	X	X	X

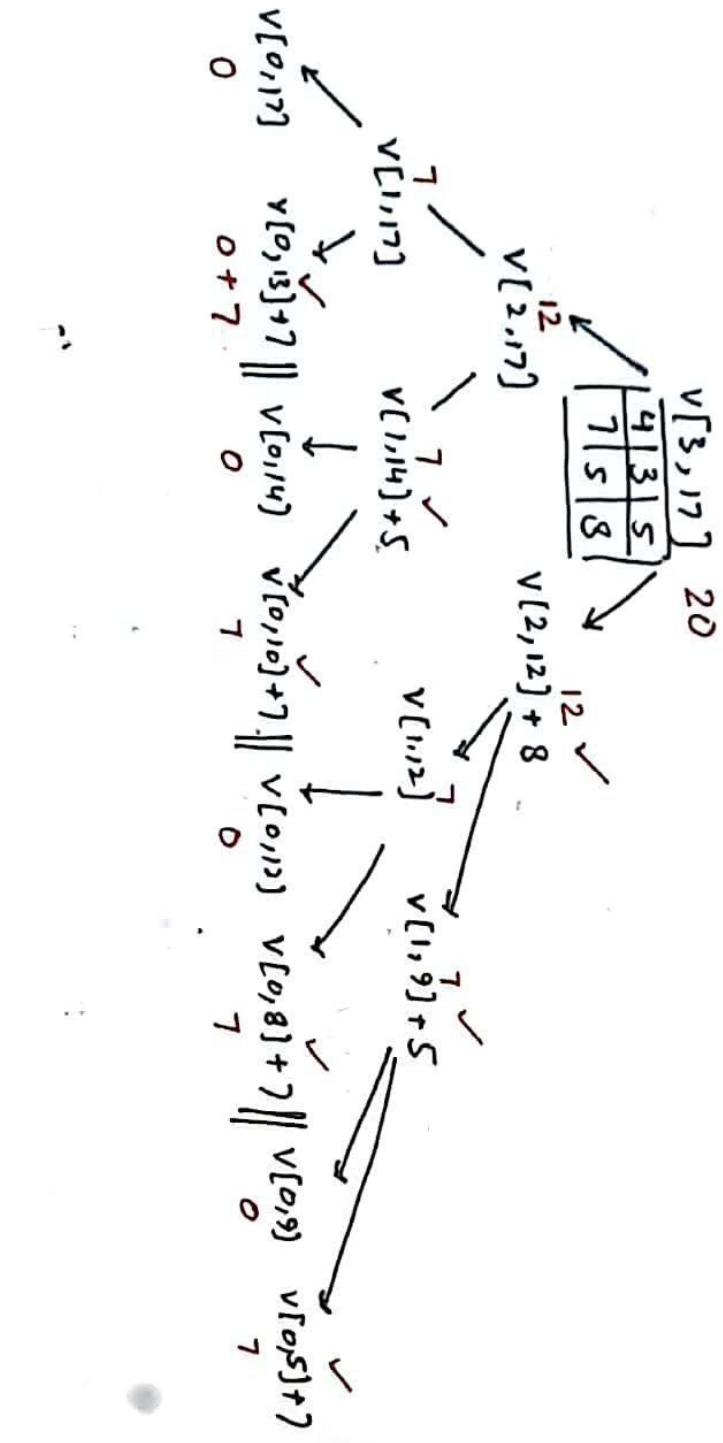
9 have shown one edge in one direction.
- If already marked, then not counted & worked again

(7-8)

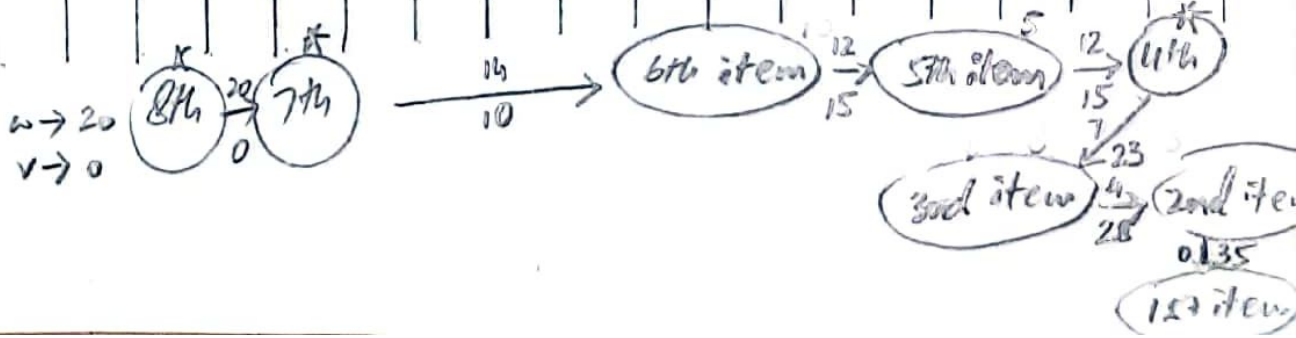
Q-5 (a)

$$V[i,j] = \begin{cases} V[i-1, j] & \text{if } j < w_i \\ \max(V[i-1, j], V[i-1, j-w_i] + v_i) & \text{if } j \geq w_i \\ 0 & \text{if } i = 0 \text{ or } j = 0 \end{cases}$$

(b)



		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
w_0		0																				
$w_1=4$	$V_1=7$	0	0	0	0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
$w_2=3$	$V_2=5$	0	0	0	5	7	7	7	12	12	12	12	12	12	12	12	12	12	12	12	12	12
$w_3=5$	$V_3=8$	0	0	0	5	7	8	8	12	13	15	15	15	20	20	20	20	20	20	20	20	20
$w_4=7$	$V_4=4$	* 0	* 0	* 0	* 5	* 7	* 8	* 8	0th 12	1st 13	2nd 15	3rd 15	4th 15	5th 20	6th 20	7th 20	8th 20	9th 20	10th 20	11th 20	12th 24	13th 24
$w_5=2$	$V_5=5$	* 0	* 0	0th 5	1st 5	2nd 7	3rd 10	4th 12	5th 13	6th 13	7th 17	8th 18	9th 20	10th 20	11th 20	12th 25	13th 25	14th 25	15th 25	16th 25	17th 25	18th 25
$w_6=6$	$V_6=10$	* 0	* 0	* 5	* 5	* 7	4th 10	0th 12	2nd 13	2nd 15	3rd 17	4th 18	5th 20	6th 22	7th 23	8th 25	9th 27	10th 28	11th 30	12th 30	13th 30	14th 35
$w_7=5$	$V_7=6$	* 0	* 0	* 5	* 5	6th 7	0th 10	2nd 12	3rd 13	3rd 15	4th 17	5th 18	6th 20	7th 22	8th 23	9th 25	10th 27	11th 28	12th 30	13th 30	14th 31	15th 35
$w_8=4$	$V_8=5$	* 0	* 0	* 5	* 5	0th 7	1st 10	2nd 12	3rd 13	4th 15	5th 17	6th 18	7th 20	8th 22	9th 23	10th 25	11th 27	12th 28	13th 30	14th 30	15th 32	16th 35



(c)

Selection is preferred