মেট্রোপলিটন ইউনিভার্সিটি

Metropolitan University

Assignment

Course Code: CSE 133 .
Course Title: Data Structure.

Submitted To

Bushra Azmat Hussain
Senior Lecturer
Department of Computer Science and Engineering
Metropolitan University,Sylhet.

Submitted By

Nishat Yeasmin Nisha
ID: 251-115-045
Batch: CSE 62nd (A).

Submission Date

14-12-2025

ID: 251-115-045

-Ans: to the Q. no. 1(a)

Given,

Infix = P-(Q*R*S+(G/H^I^J)*K)

Reverse =)K*)J^I^H/G(+S*R*Q(-P

Input	Stack	Prefix Expression.
)	)	
K	)	K
*	)*	K
)	)*)	K
J	)*)	KJ
^	)*)^	KJ
I	)*)^	KJI
^	)*)^	KJI^
H	)*)^	KJI^H
/	)*)/	KJI^H^
G	)*)/	KJI^H^G
(	)*	KJI^H^G/
+	)+	KJI^H^G/*
S	)+	KJI^H^G/*S
*	)+*	KJI^H^G/*S
R	)+*	KJI^H^G/*SR
*	)+**	KJI^H^G/*SR
Q	)+**	KJI^H^G/*SRQ
(		KJI^H^G/*SRQ**+
-	-	KJI^H^G/*SRQ**+
P	-	KJI^H^G/*SRQ**+P
		KJI^H^G/*SRQ**+P-

Final Infix to Prefix Expression:-

$-P+**QRS*/G\wedge H\wedge IJK$.

Ans. to the Q. No. 1 (b)

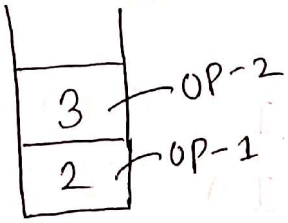
Given,

Postfix Expression: $AB\wedge CAA\wedge *+DE/-$

where, $A=2, B=3, C=5, D=12, E=6$

Postfix Expression: $23\wedge 522\wedge *+12\,6/-$

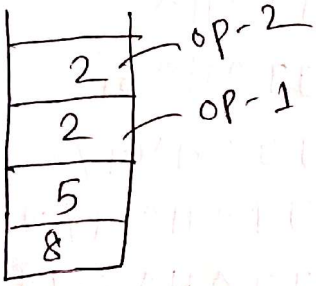
Step 1:-



OP-1 - operator - OP-2

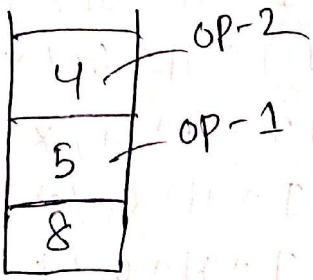
$\therefore 2 \wedge 3 = 8$

Step 2:-



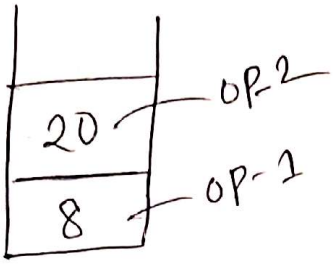
$\therefore 2\wedge 2 = 4$

Step 3:-



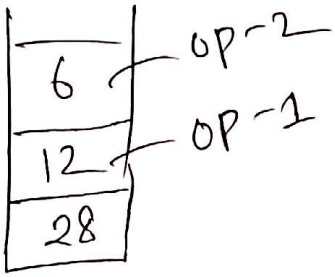
$5 * 4 = 20$

Step 4:-



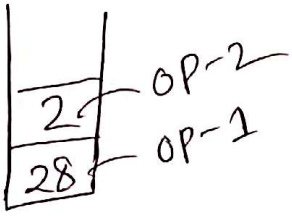
∴ 8 + 20 = 28

Step 5:



∴ 12 / 6 = 2

Step 6:-



28 - 2 = 26.

∴ Ans = 26.

ID: 251-115-045

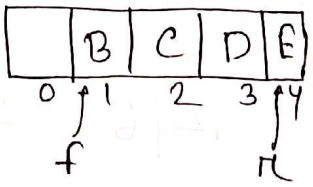
Ans. to the Q. No. 1 (c)

Given,

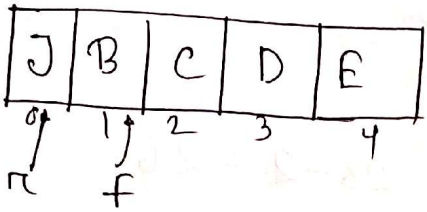
Front = 1

Rear = 4

Queue:



1. Enqueue(J)



Front = 1

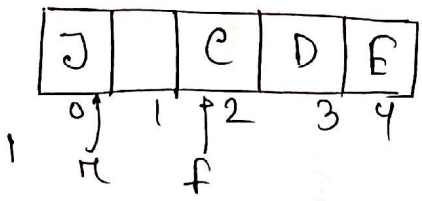
Rear = (4+1) % 5

= 0

2. Enqueue(K)

Queue is overflow.

3. Dequeue( )



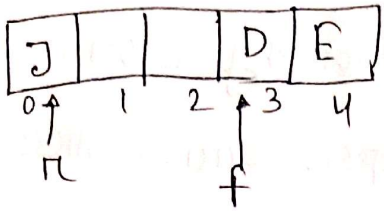
B removed from the queue.

Front = 2

Rear = 0.

ID: 251-115-045

4. Dequeue( )

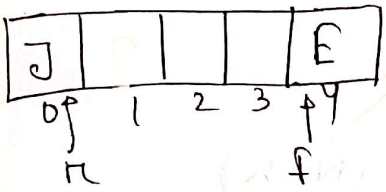


c removed from the Queue.

Front = 3

Rear = 0

5. Dequeue( )

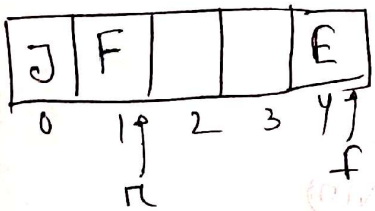


D removed from the Queue.

Front = 4

Rear = 0

6. Enqueue(F)



F insert into the Queue.

Front = 4

Rear = 1

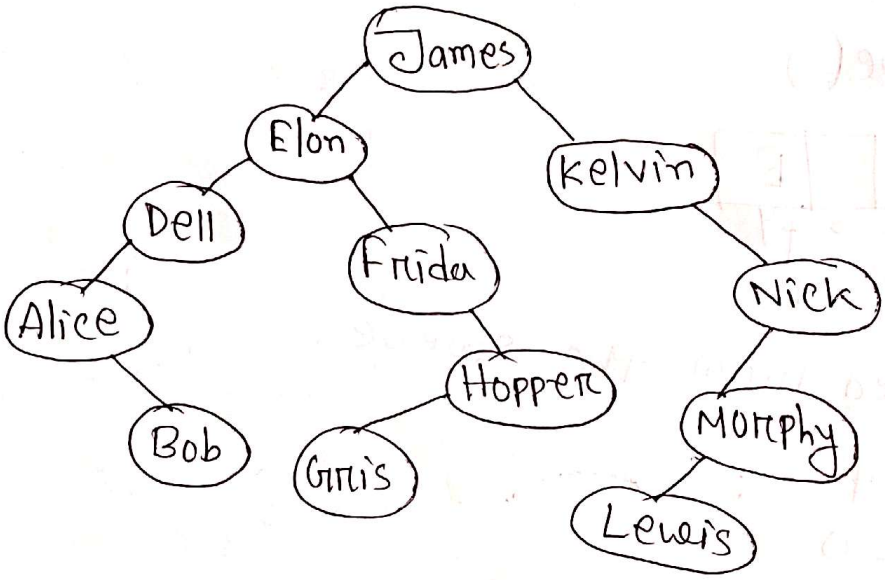
Ans

ID:- 251-115-045

Ans. to the Q. NO. 2(a)

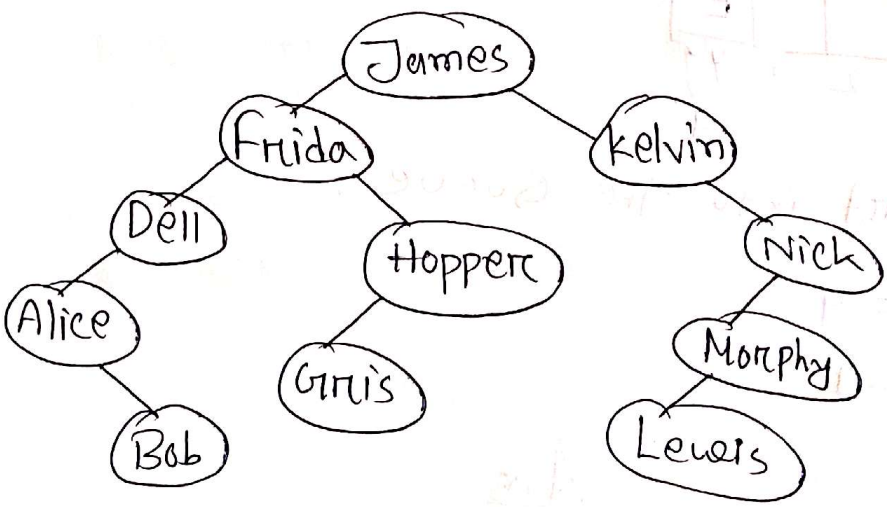
Given: {James, Kelvin, Nick, Morphy, Elon, Frida, Lewis, Dell, Hopper, Gitis, Alice, Bob}.

B.S.T :-



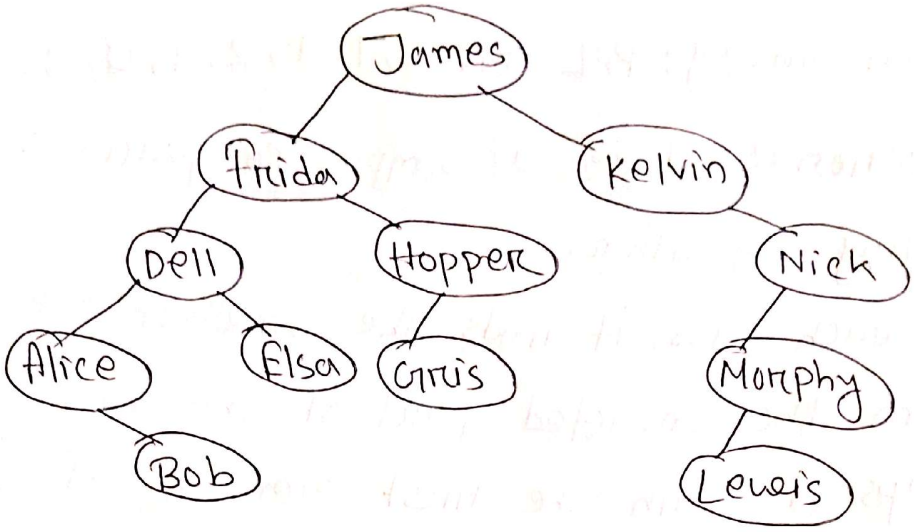
Ans. to the Q - No 2(b)

After delete "Elon".



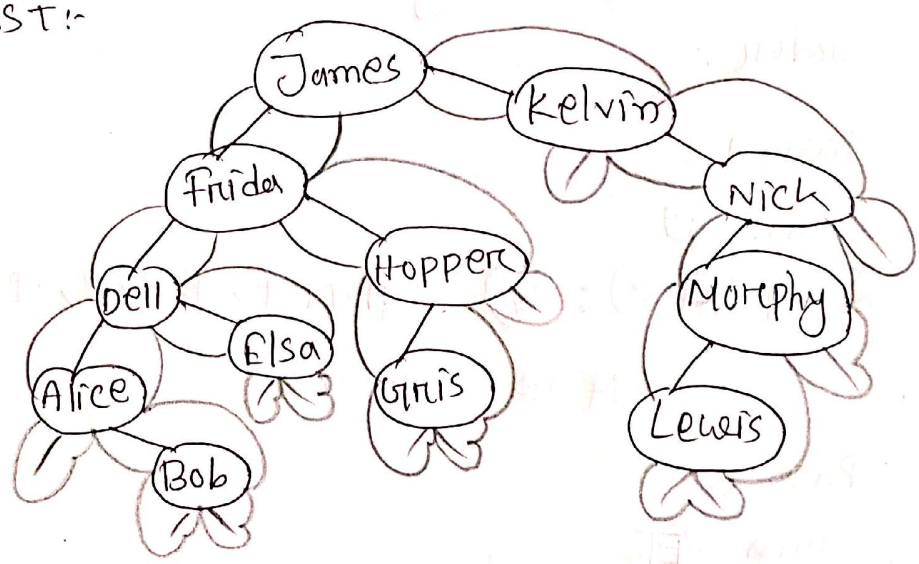
ID: 251-115-045

After Insert "Elsa".



Ans. to the Q.No. 2(c)

from 2(b), BST:-



Pre-order: James, Frida, Dell, Alice, Bob, Elsa, Hopper, Gris, Kelvin, Nick, Morphy, Lewis.

Post-order:- Bob, Alice, Elsa, Dell, Gris, Hopper, Frida, Lewis, Morphy, Nick, kelvin, James.

ID: 251-115-045

Ans. to the Q. No. 3(a)

Given Array: P, L, A, N, E, T, A, R, I, U, M.

Selection Sort is a simple comparison-based sorting algorithm.

In each pass, it finds the smallest element from the unsorted part of the array and swaps it with the first element of that unsorted part. This process repeats until the entire array is sorted in ascending order.

Pass 1:

min = A

Swap (A, P): (A), L, P, N, E, T, A, R, I, U, M.

A is sorted.

Pass 2:

min = A

Swap (A, L): (A), (A), P, N, E, T, L, R, I, U, M

A and A are sorted.

Pass 3:

min = E

Swap (E, P): (A), (A), (E), N, P, T, L, R, I, U, M

Pass 4:-
min = I
Swap (I, N): (A), (A), (E), (I), P, T, L, R, N, U, M.

Pass 5:
min = L
Swap (L, P):- (A), (A), (E), (I), (L), T, P, R, N, U, M.

Pass 6:-
min = M
Swap (M, T): (A), (A), (E), (I), (L), (M), P, R, N, U, T.

Pass 7:
min = N
Swap (N, P): (A), (A), (E), (I), (L), (M), (N), R, P, U, T.

Pass 8:
min = P
Swap (P, R): (A), (A), (E), (I), (L), (M), (N), (P), R, U, T.

Pass 9:
min = R
=> (A), (A), (E), (I), (L), (M), (N), (P), (R), U, T.

Pass 10:-
min = T
Swap (T, U): (A), (A), (E), (I), (L), (M), (N), (P), (R), (T), U.

Pass 11:-
min = U.
Sorted Array: (A), (A), (E), (I), (L), (M), (N), (P), (R), (T), (U).

(Ans)

ID: 251-115-045

Ans. to the Q. no. 3(b)

From 3(a), Sorted Array:

A	A	E	I	L	M	N	P	R	T	U
0	1	2	3	4	5	6	7	8	9	10

key = T; n = 11

Initially l = 0, r = n - 1

Step 1: Check the array is sorted in ascending/ descending order.

Step 2: Find the middle element of the array.

Step 3: If arr[mid] = key. the search stop and value is found.

Step 4: If arr[mid] < key. left element is updated. l = mid + 1.

Step 5: If arr[mid] > key. right element is update r = mid - 1.

Now,

cases:

A	A	E	I	L	M	N	P	R	T	U
0	1	2	3	4	5	6	7	8	9	10

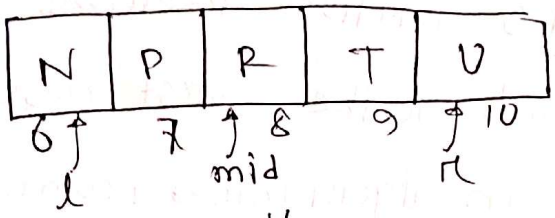
l = 0 (mid = 5) r = 10

$$mid = \frac{l + r}{2} = \frac{0 + 10}{2} = 5$$

arr[mid] < T

$l = mid + 1$
$= 5 + 1$
$= 6 ; r = 10$

case 2:-

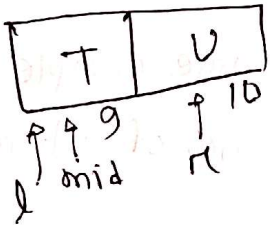


$$mid = \frac{6+10}{2} = \frac{16}{2} = 8$$

$arr[mid] < T$

$l = mid + 1$
$= 8 + 1$
$= 9, r = 10$

case 3:-



$$mid = \frac{9+10}{2} = \frac{19}{2} = 9.5 \approx 9$$

$arr[mid] = T$

$\therefore$ T found at index 9.

ID: 251-115-045

Ans. to the Q. No. 3(c)

The three most common Asymptotic notations are Big O, Big Omega (Ω), and Big Theta (Θ).

(i) Big O Notation:- Big O notation describe the upper bound or the worst case scenario for an algorithm's running time. If focuses on the maximum time an algorithm can take to complete.

Example: In a linear search algorithm, the worst case is when the target element is at the very end of the list or not present at all. The algorithm has to check every element. Thus, its time complexity is $O(n)$, where n is the no. of elements.

(ii) Big Omega (Ω) notation:- Big Omega notation describes the lower bound or the best-case scenario for an algorithm's running time. It is the opposite of Big O notation.

Example:- For a linear search algorithm, the best case is when the target element is the first element in the list. In this case, the

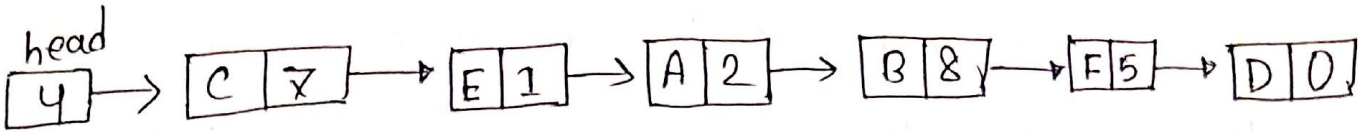
algorithm takes a constant amount of time, regardless of the list's size. It's time complexity is $\Omega(1)$.

(iii) Big theta(Θ) notation:- Big theta notation describes the average-case performance or a tight bound for an algorithm's running time. It is used when the best-case and worst-case complexities are the same.

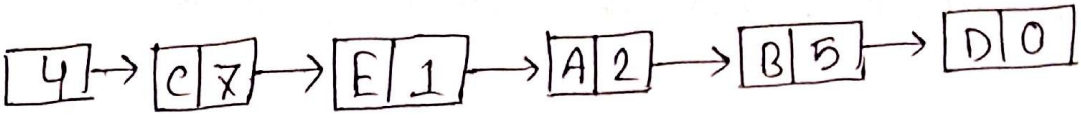
Example:- The Merge sort Algorithm performs consistently in all cases (best, worst and average) due to its divide-and-conquer approach. It's time complexity is $\Theta(n \log n)$, meaning its running time is always in the order of $n \log n$.

(4)(a)

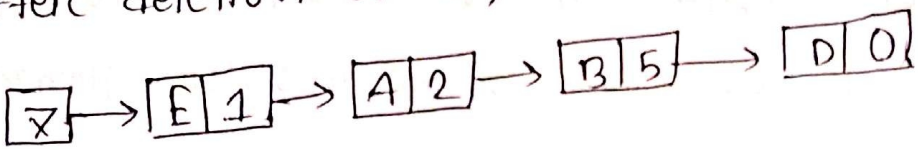
START = 4



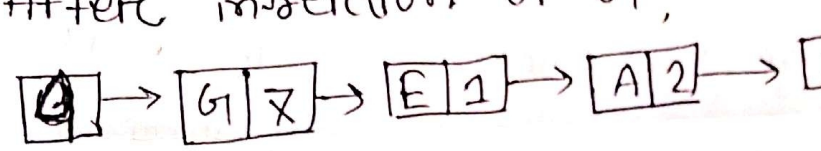
(4)(b) After deletion of F,



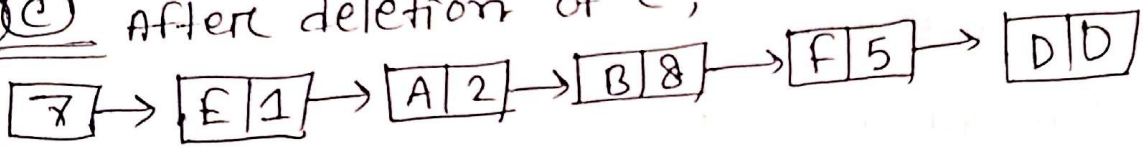
After deletion of C,



After insertion of G1,

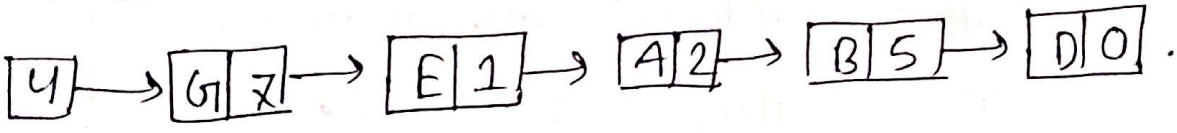


(4)(c) After deletion of C,

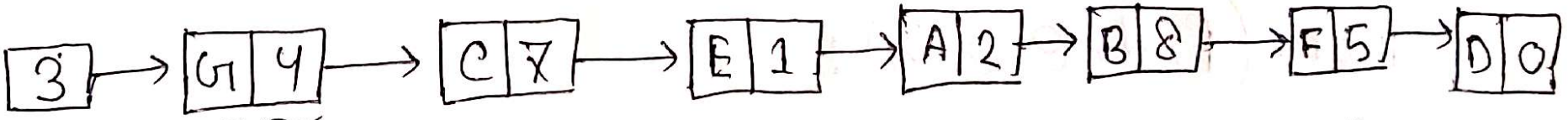


After deletion of F,

After insertion of G1,

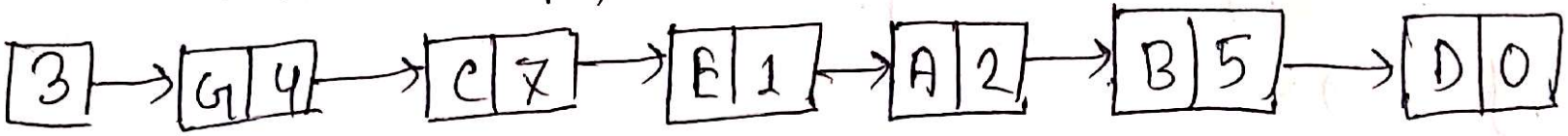


(9d) After insertion of G,

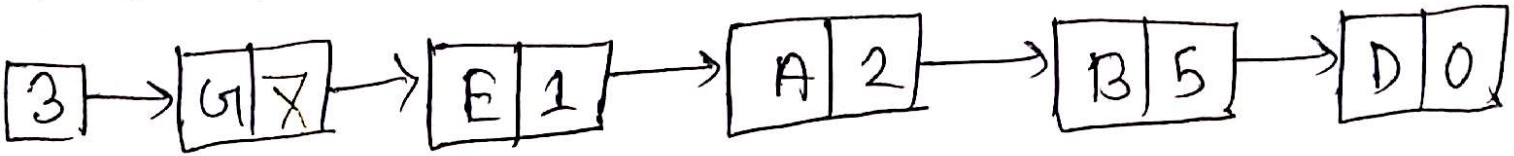


replaced 6 with 4 as the correct link for G.

After deletion of F,



After deletion of C.



ID: 251-115-045

Ans. to the Q. No. 5(a)

Starting from vertex F.

from (F),

f – d = 6 ✓

f – e = 8

f – g = 11

from (F, d):

d – a = 9

d – b = 9

d – e = 15

f – e = 8 ✓

f – g = 11

from (f, d, e):

e – g = 9

e – c = 5 ✓

e – b = 7

from (f, d, e, c):

c – b = 8

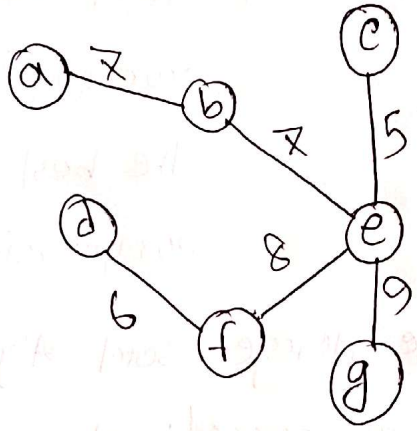
e – b = ✓

from (f, d, e, c, b):

b – a = 7 ✓

~~fro~~ remaining vertex g:-

e – g = 9 ✓



Total minimum cost = 6 + 8 + 5 + 7 + 7 + 9 = 42

vertex = 7

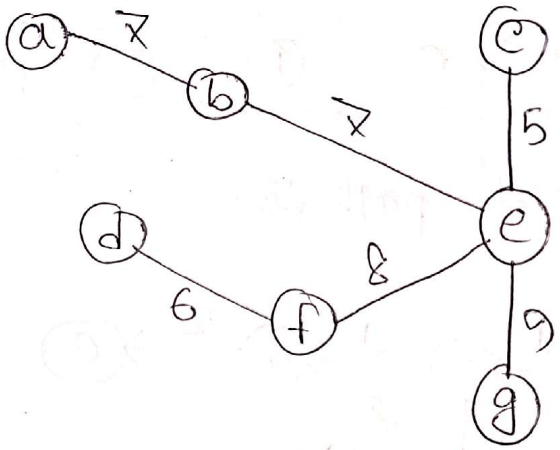
Edge = 7 – 1

= 6

ID:- 251-115-045

Ans. to the Q. No. 5 (b).

no.	source	destination	weight.
1	c	e	5
2	d	f	6
3	a	b	7
4	b	e	7
5	e	f	8
X6	a	d	9
X7	b	d	9
8	e	g	9
X9	f	g	11
X10	D	e	15

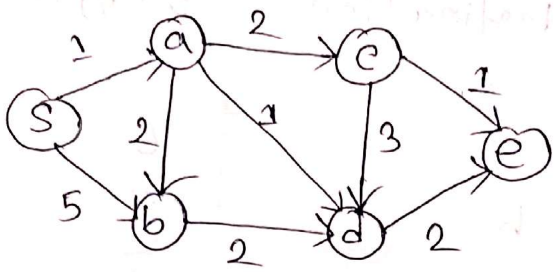


vertex = 7

Edge = 7 - 1
= 6

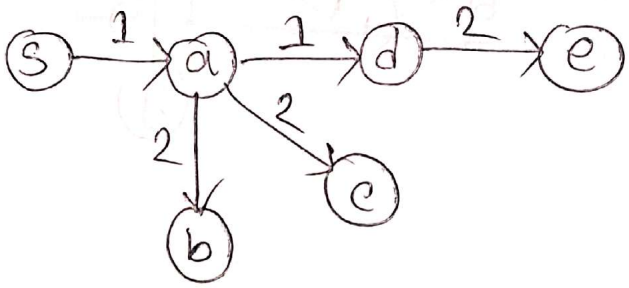
Total cost = 5 + 6 + 7 + 7 + 8 + 9
= 42.

Ans. to the Q. No. 6-(a)



Visited Vertex	Destination				
	a	b	c	d	e
	∞	∞	∞	∞	∞
s	(1)	5	∞	∞	∞
s, a	(1)	3	3	(2)	∞
s, a, d	(1)	(3)	3	(2)	4
s, a, d, b	(1)	(3)	(3)	(2)	4
s, a, d, b, e	(1)	(3)	(3)	(2)	(4)
s, a, d, b, c, e	(1)	(3)	(3)	(2)	(4)

The shortest path is:-

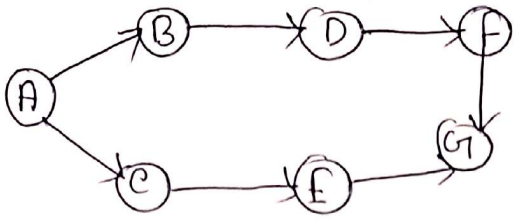


Ans. to the Q. No. 6(b)

Given nodes = {A, B, C, D, E, F, G}

edges = {(A,B), (A,C), (B,D), (C,E), (D,F), (E,G), (F,G)}.

The graph is:-



Unvisited Array:

F	F	F	F	F	F	F
0	1	2	3	4	5	6

visited Array:

A	B	C	D	E	F	G
T	T	T	T	T	T	T
0	1	2	3	4	5	6

Queue :-

~~A~~	~~B~~	~~C~~	~~A~~	~~D~~	~~A~~	~~E~~	~~B~~	~~F~~	~~C~~	~~G~~	~~D~~	~~G~~	~~E~~	~~F~~

B.F.S :-

A	B	C	D	E	F	G

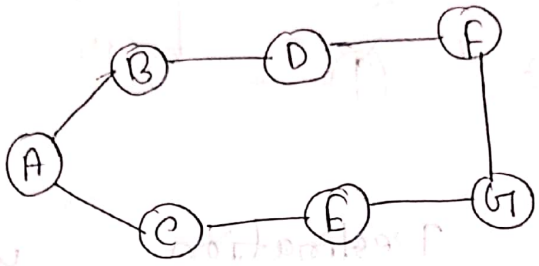
Ans

DFS

Given nodes = {A, B, C, D, E, F, G}

edges = {(A,B), (A,C), (B,D), (C,E), (D,F), (E,G), (F,G)}.

The graph is:



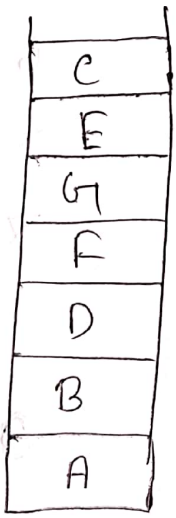
Unvisited Array:

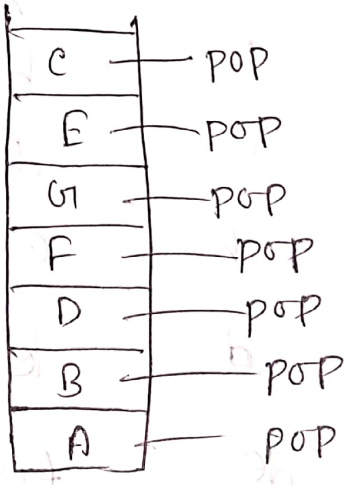
F	F	F	F	f	f	f
0	1	2	3	4	5	6

visited Array:

T	T	T	T	T	T	T
0	1	2	3	4	5	6

stack:



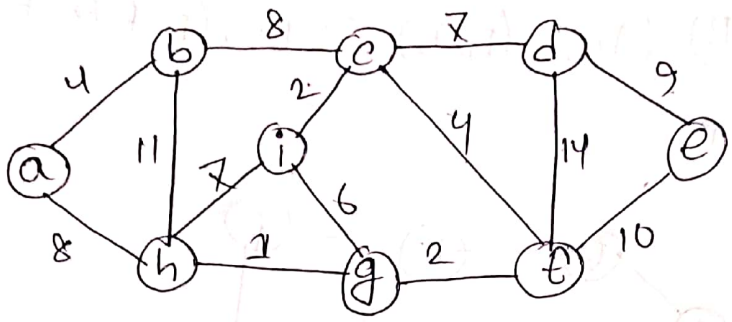


D.F.S :-

A	B	D	F	G	E	C
0						

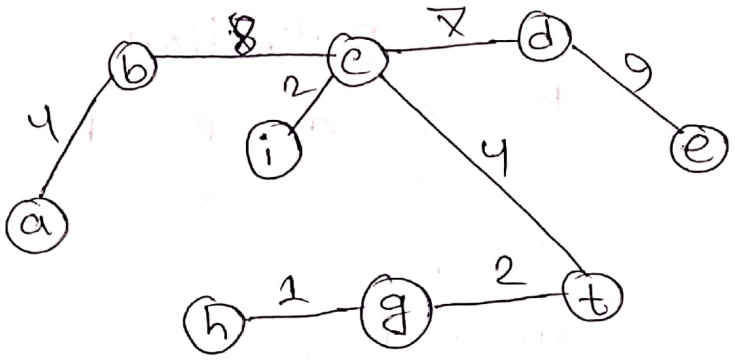
Ans

Ans. to the Q.no. 7(a)



no.	Source	Destination	weight
1	h	g	1
2	g	f	2
3	c	i	2
4	a	b	4
5	c	f	4
6	i	g	6
7	i	h	7
8	c	d	7
9	b	c	8
10	a	h	8
11	d	e	9
12	e	f	10
13	b	h	11
14	d	f	14

MST :-



Vertex = 9

Edge = 9 - 1

= 8

Total weight = 4 + 8 + 7 + 9 + 2 + 4 + 2 + 1

= 37

Ans. to the Q. No. 7(b)

Source	Destination							
a	b	c	d	e	f	g	h	i
	∞	∞	∞	∞	∞	∞	∞	∞
a	(4)	∞	∞	∞	∞	∞	8	∞
a, b	(4)	12	∞	∞	∞	∞	(8)	∞
a, b, h	(4)	12	∞	∞	∞	(9)	(8)	15
a, b, h, g	(4)	12	∞	∞	(11)	(9)	(8)	15
a, b, h, g, f	(4)	(12)	25	21	(11)	(9)	(8)	15
a, b, h, g, f, c	(4)	(12)	19	21	(11)	(9)	(8)	(14)
a, b, h, g, f, c, i	(4)	(12)	(19)	21	(11)	(9)	(8)	(14)
a, b, h, g, f, c, i, d	(4)	(12)	(19)	(21)	(11)	(9)	(8)	(14)
a, b, h, g, f, c, i, d, e	(4)	(12)	(19)	(21)	(11)	(9)	(8)	(14)

The sortest path:-