M T W T F S S

COMPASS

Date :

14/2/24

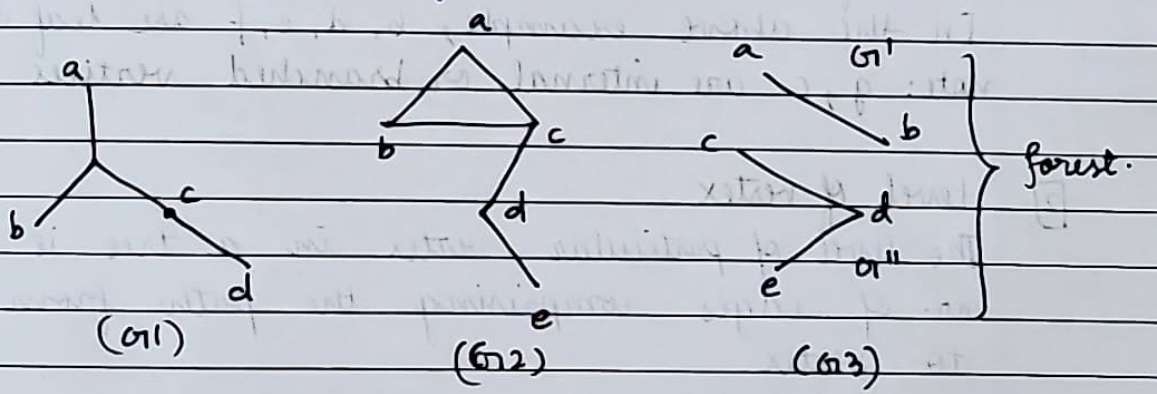
UNIT-5 TREES & ITS APPLICATIONS

Tree

A graph $G = (V, E)$ be a loop free graph, it is called as tree iff

i) G is connected

ii) G has no cycles



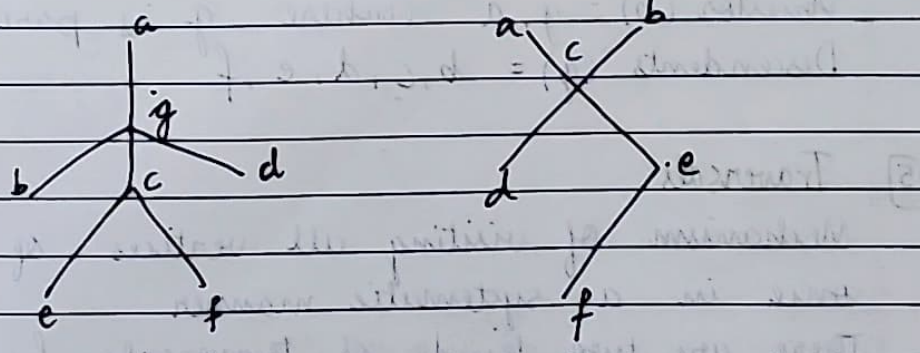
G1 is a ~~tttte~~ tree.

G2 is not a tree because it has a cycle.

G' and G'' are trees so G3 is known as a forest

Rooted tree

$G = (V, E)$ is called as rooted tree iff there is a unique vertex r called root with $id(r) = 0$ and for all other vertices v, $id(v) = 1$.



a rooted tree

tree but not rooted tree

M T W T F S S

COMPASS

Date :

Terminologies

1. Leaf

A vertex whose out degree i.e. $od(v) = 0$ is known as leaf or terminal vertex.
In the above example, b, d, e, f are leaf vertices
note: g, c are internal or branched vertices

2. Level of vertex

The level of particular vertex in a tree is the no. of edges compising the path from root to vertex.
eg: level (b) = 2 , level (e) = 3
The root is always at level 0.

3. Sibling

Two vertices having same parents.
eg: c, d, b are siblings as they have common parent 'g'
e, f are siblings

4. Ancestor and Descendant

Ancestor (b) = g, a (where g is parent)
Descendents (g) = b, c, d, e, f

5. Traversals

Mechanism of visiting all vertices of tree exactly once in a systematic manner.
There are two kinds of traversals for a tree:

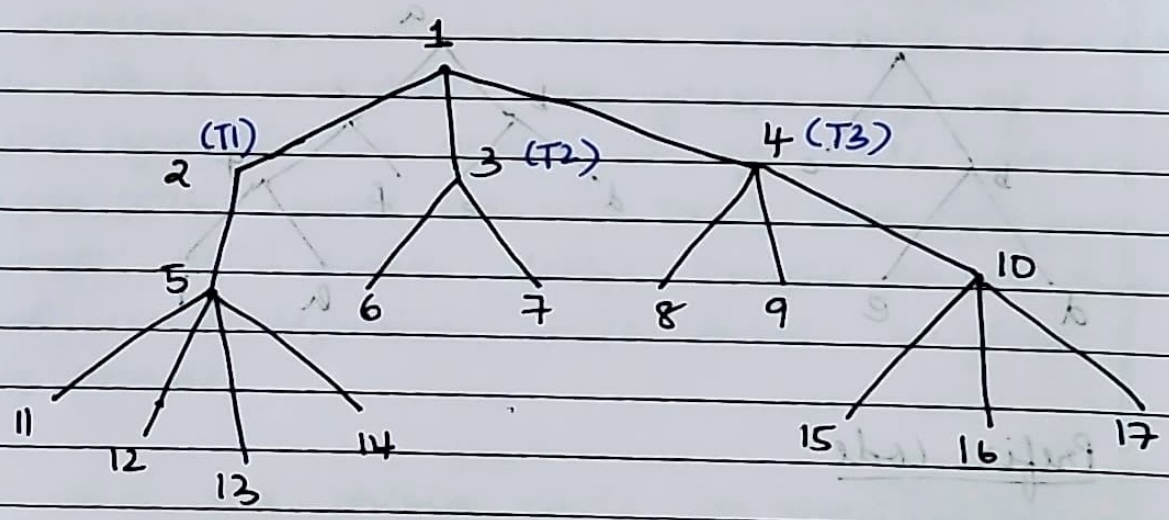
M T W T F S S

COMPASS

Date :

Preorder and Postorder.

For binary tree, we have one more traversal i.e. inorder traversal.



Preorder:

T1: 2, 5, 11, 12, 13, 14

T2: 3, 6, 7

T3: 4, 8, 9, 10, 15, 16, 17

∴ Traversal: 1, 2, 5, 11, 12, 13, 14, 3, 6, 7, 4, 8, 9, 10, 15, 16, 17

Postorder

T1: 11, 12, 13, 14, 5, 2

T2: 6, 7, 3

T3: 8, 9, 15, 16, 17, 10, 4

∴ Traversal: 11, 12, 13, 14, 5, 2, 6, 7, 3, 8, 9, 15, 16, 17, 10, 4, 1

Binary Tree Rooted Tree

A rooted tree is a binary rooted tree if for each vertex v, od(v) = 0, 1, or 2 i.e. v has atmost two children.

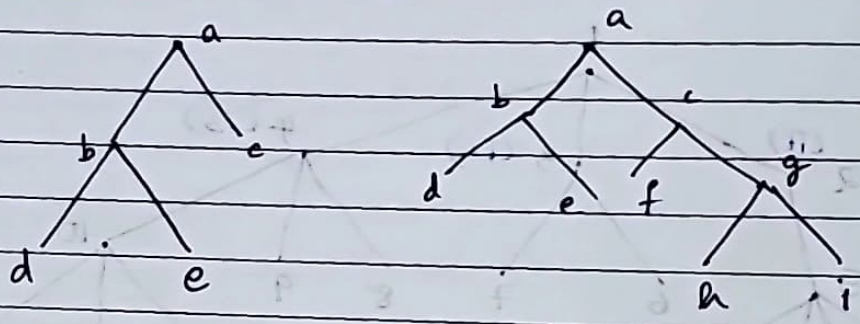
M T W T F S S

COMPASS

Date :

Complete Binary Tree

Rooted Binary Tree is a complete binary tree iff $\forall v \in V$ $od(v) = 0$ or 2.



Prefix code

A set P of binary sequences containing 0 or 1 representing a set of symbols is called prefix code if no sequence in P is the prefix of any other sequence in P. Such a sequence for a given set of symbols is called prefix code.

eg: a: 111, e: 0, n: 1100, r: 1101, t: 10

given a set of symbols {a, e, n, r, t}

P: {111, 0, 1100, 1101, 10}

Encoding

ata : 11110111

ene : 011000

ane : 111111000

Decoding:

1100100 : nte

111010 : aet

M T W T F S S

COMPASS

Date :

Huffman Method

It is used for constructing optimal prefix code by generating a labeled complete binary tree which considers the frequency of occurance of the symbols in order to generate variable length prefix codes wherein most frequent used symbols will have lesser length ~~than~~ and least used symbols will have higher length code words.

(1) (10 marks)

Determine the prefix code for the following symbols. {a, u, o, q, y, z} with frequencies {20, 17, 28, 4, 12, 7}

Apply Huffman method to generate optimal prefix code,

Step 1: Consider the frequencies of the symbols as the weights and arrange them in ascending order.

q : 4	z : 7	y : 12	u : 17	a : 20	o : 28

Step2: Consider the first two nodes which are the least ones and combine them into a tree with root node which is having ~~some~~ sum of frequencies as the weight.

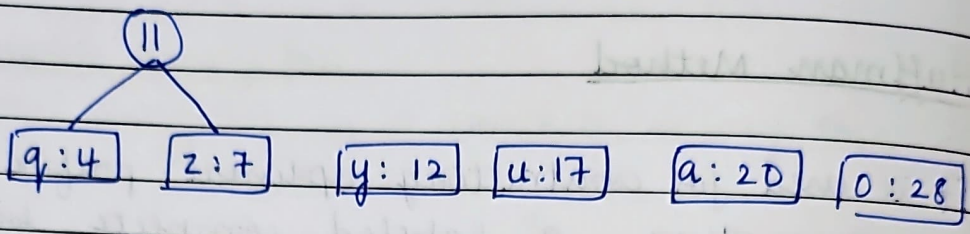
After constructing the tree, again arrange them in ascending order till one complete binary tree is being formed.

M T W T F S S

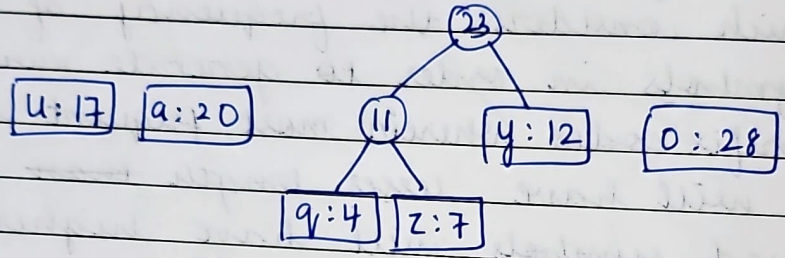
COMPASS

Date :

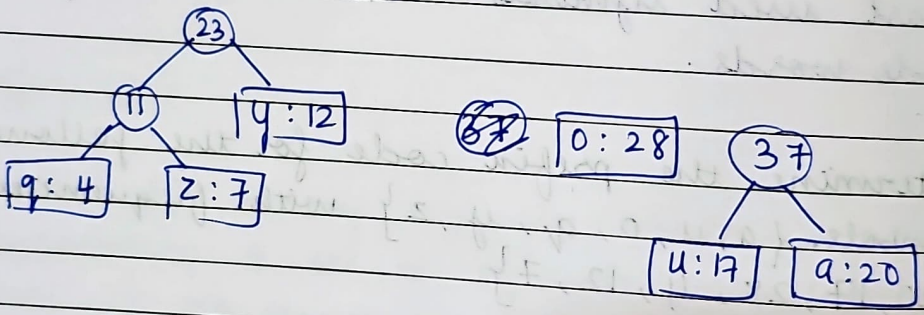
*



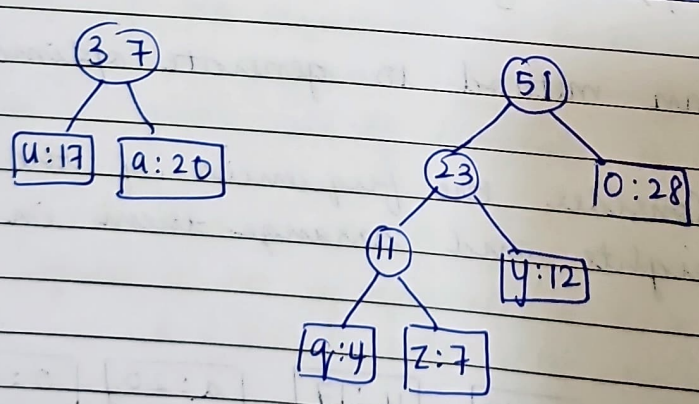
*



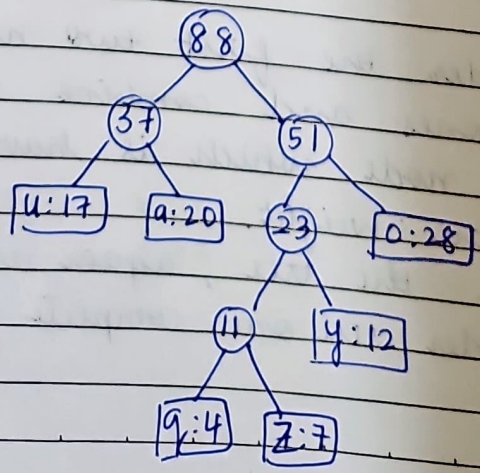
*



*



*

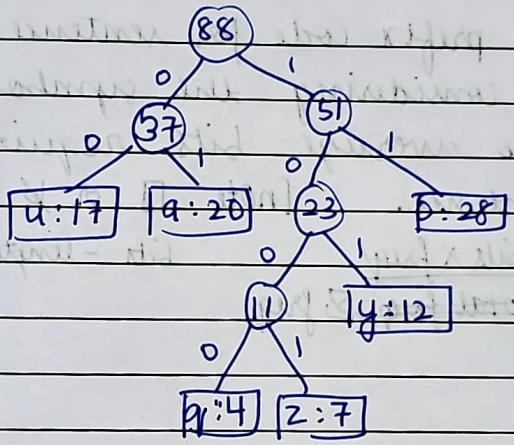


M T W T F S S

COMPASS

Date :

Step 3: Label the complete binary tree obtained in Step 2 with 0 and 1 for left and right edge respectively. This will result in giving Huffman tree.



Step 4: Assign the code word for the symbols is done by accumulating the labeled edges from root node till leaf node that represents the symbols.

Symbol	Code word
a	01
o	11
u	00
q	1000
y	101
z	1001

Decode the below sequence by using the prefix code generated:

01111001100011

aozqo

M T W T F S S

Date :

② Construct an optimal prefix code for the letters of the word ENGINEERING. Hence deduce the code for this word.

(HW)

③ Obtain the prefix code for sentence 'MISSION IS SUCCESSFUL' considering the symbols and obtain the average bits required to encode the ~~word~~ sentence. [note □ or ∅ - space]

Avg: $\frac{\Sigma \text{ bits} \times \text{freq}}{\Sigma \text{ freq}}$ bits - length of code word.

Ans ②

Step 1:

Letters : { E, N, G, I, R }

Frequencies : { 3, 3, 2, 2, 1 }

R:1 | G:2 | I:2 | E:3 | N:3

Step 2:

* I:2 | E:3 | N:3 | 3 (R:1, G:2)

* N:3 | 3 (R:1, G:2) | 5 (I:2, E:3)

* 5 (I:2, E:3) | 6 (N:3, 3 (R:1, G:2))

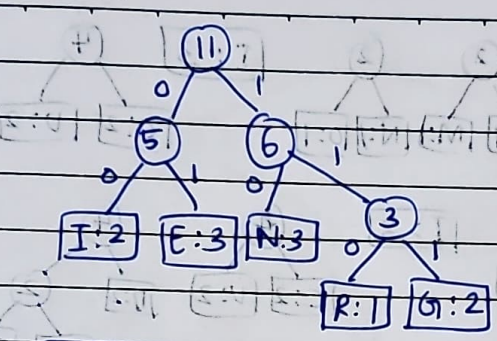
* 11 (5 (I:2, E:3), 6 (N:3, 3 (R:1, G:2)))

M T W T F S S

COMPASS

Date :

Step 3:



Step 4:

Symbol	Code word
E	01
N	10
G	111
I	00
R	110

∴ Code for ENGINEERING is

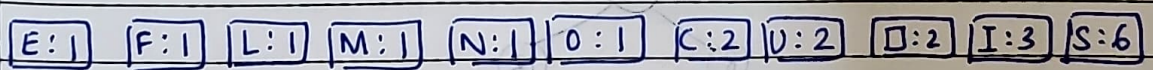
"01101110010010111000101111"

Ans ③

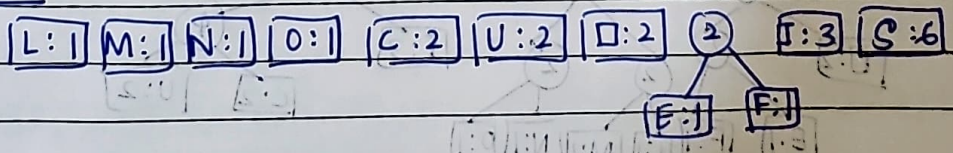
Step 1:

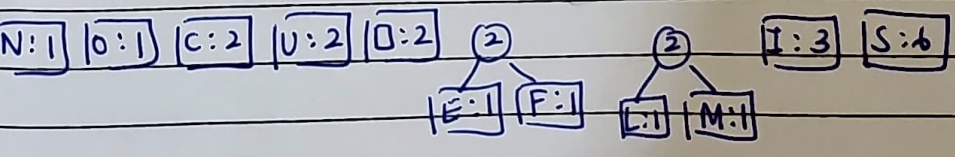
Symbols: {M, I, S, O, N, U, C, E, F, L, □} where □ is space

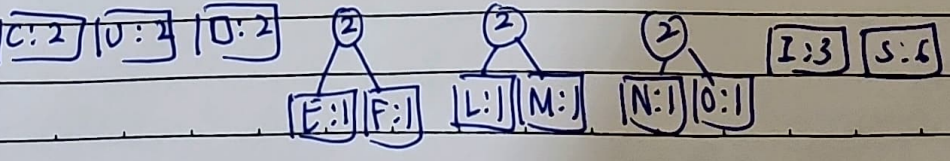
Frequencies: {1, 3, 6, 1, 1, 2, 2, 1, 1, 1, 2}



Step 2:

* 

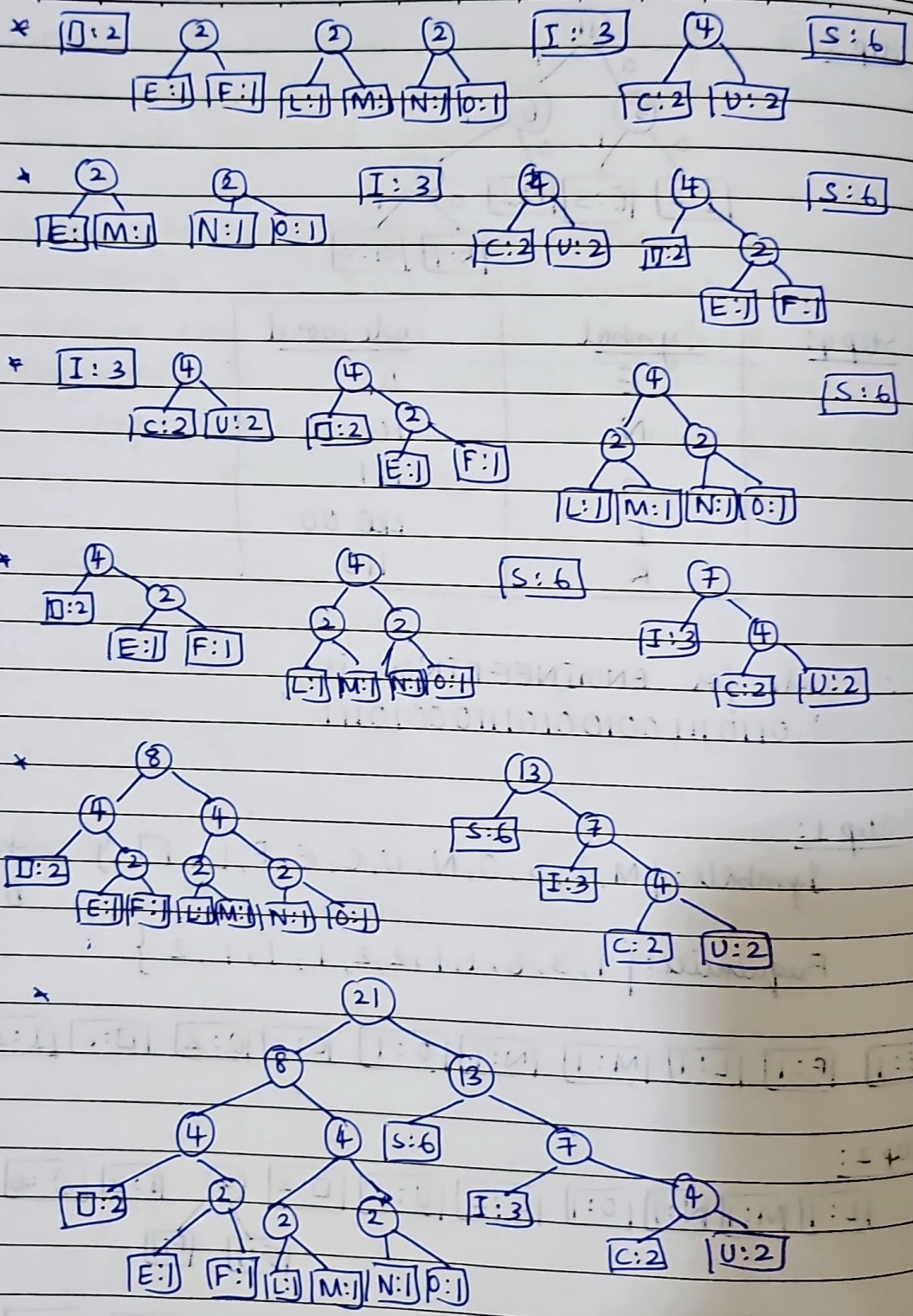
* 

* 

M T W T F S S

COMPASS

Date :

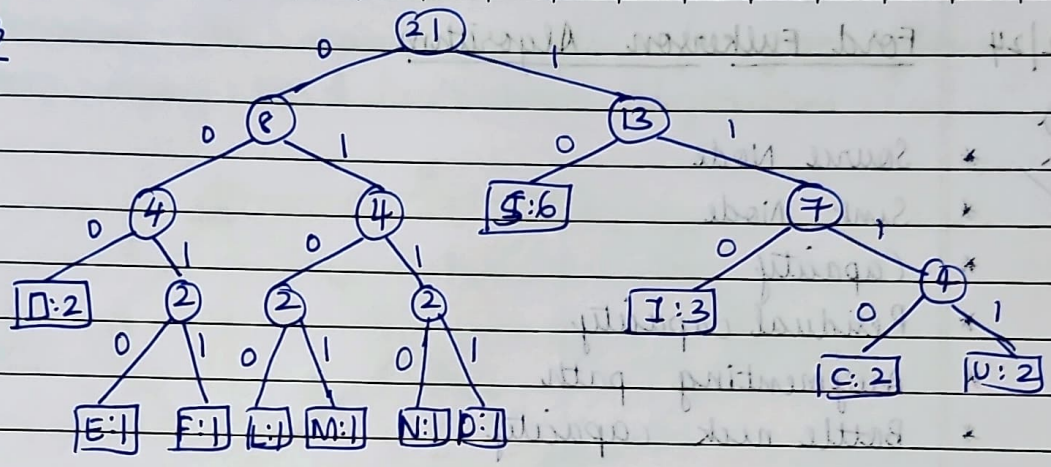


M T W T F S S

COMPASS

Date :

Step3



Step4:

Symbols	Code word
M	0101
I	110
S	10
O	0111
N	0110
□	000
U	1111
C	1110
E	0010
F	0011
L	0100

Prefix code for 'MISSION IS SUCCESSFUL' is

0101110101011001110110 000 11010 000101111110111000101010 00111111110100

Average bits $= \frac{(4\times1)+(3\times3)+(2\times6)+(4\times1)+(4\times1)+(3\times2)+(4\times2)+(4\times2)+(4\times1)+(4\times1)+(4\times1)}{21}$

$= \frac{67}{21} = 3.2 \times 4$

Refer in textbk for binary tree representation of expression

M T W T F S S

COMPASS Date :

16/2/24

Ford Fulkerson Algorithm

Final diff

* Source Node
* Sink Node
* Capacity
* Residual capacity
* Augmenting path
* Bottle neck capacity

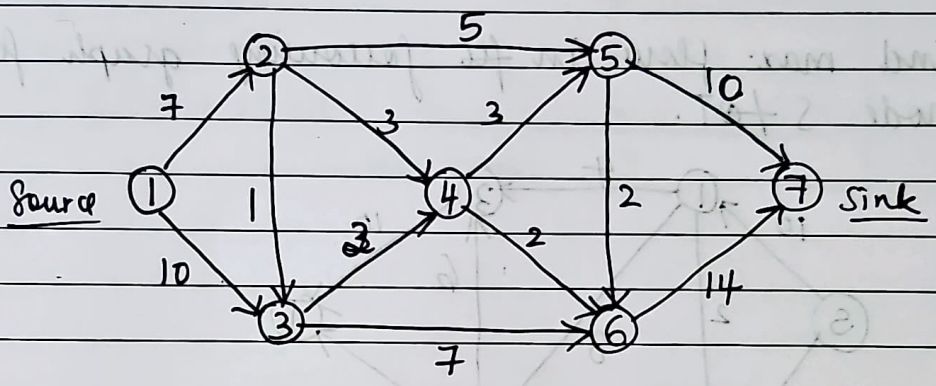
- Ford Fulkerson algorithm is used to solve maximum flow problems in flow network
- Maximum flow problem involves determining the maximum amount of flow that can be sent from source to sink in a directed graph subject to capacity constraints on the edges
- The algorithm works by iteratively finding augmenting path from source to sink in the residual graph i.e. the graph obtained by subtracting the current flow from capacity of edge.
- Given a graph which denotes a flow network, every edge has a capacity and two vertices: source and sink designated as S and T.
- Max. flow from S to T is obtained with the following constraints:
  i) Flow on any edge doesn't exceed the given capacity of the edge
  ii) Incoming flow = outgoing flow for all vertices except source and sink node.

M T W T F S S

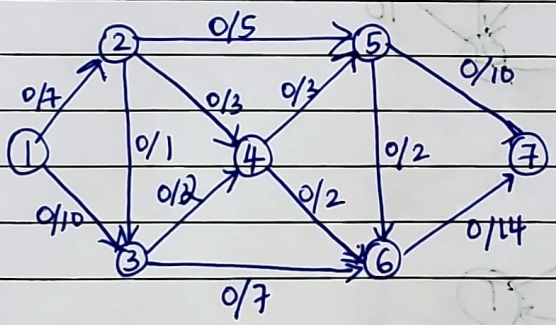
COMPASS

Date :

Q Determine the maximum flow for given flow network using Ford Fulkerson algorithm.

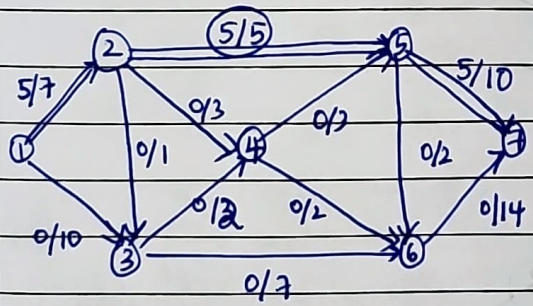


1

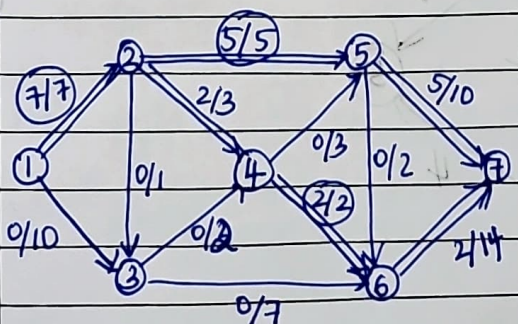


Augmenting path	Bottleneck capacity
1→2→5→7	5
1→2→4→6→7	2
1→3→6→7	7
1→3→4→5→7	2
Max flow:	16

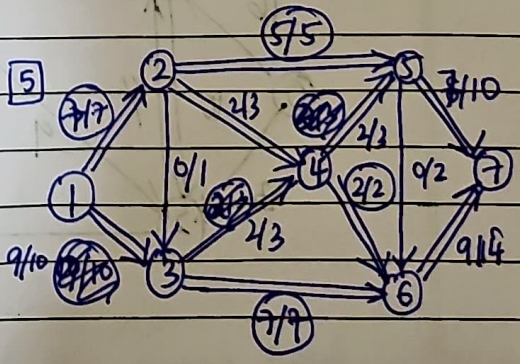
2



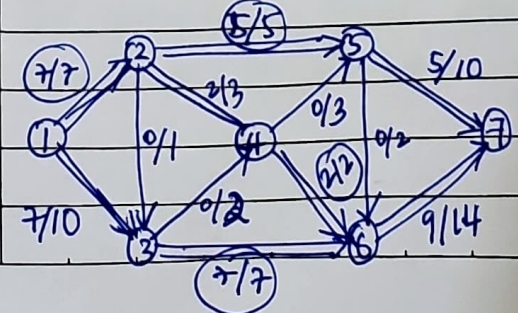
3



5



4



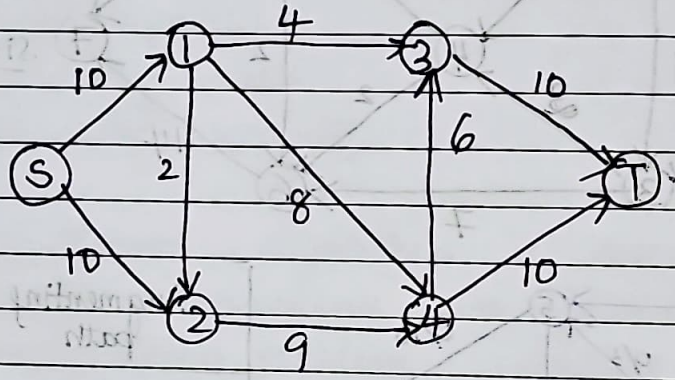
M T W T F S S

Date :

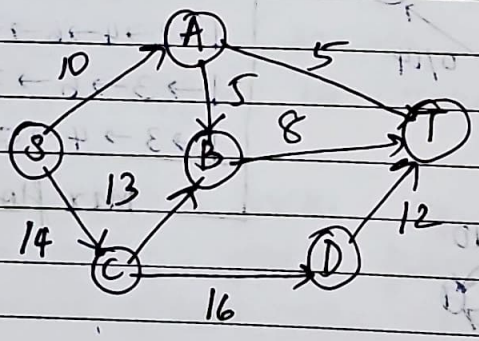
The maximum flow obtained by using Ford & Fulkerson algorithm is $5 + 2 + 2 + 7 = 16$

HW Q) Find max. flow in the following graphs from node S to T.

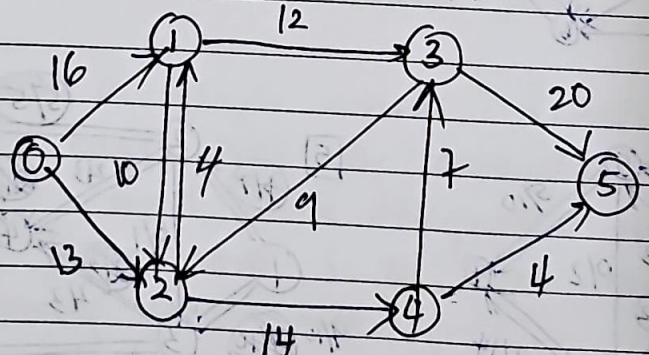
a)



b)



c)



M T W T F S S

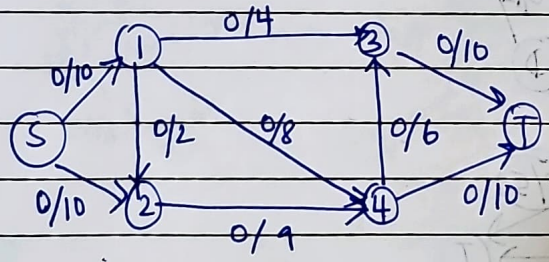
COMPASS

Date :

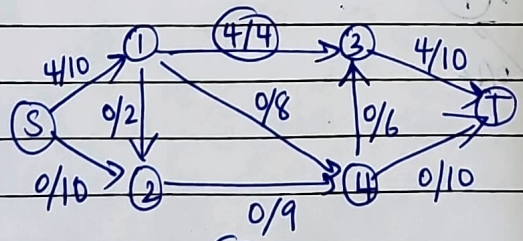
a)

Augmenting path	Bottle neck capacity
S→1→3→T	4
S→2→4→T	9
S→1→4→3→T	6
Max flow:	19

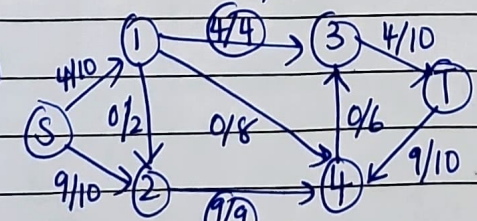
1



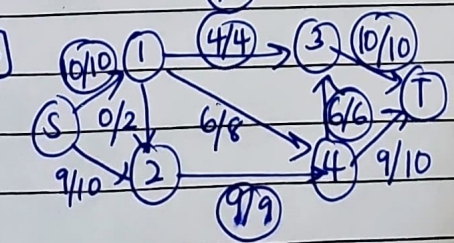
2



3



3



The maximum flow obtained by using Ford Fulkerson algorithm is $4+9+6 = \underline{19}$

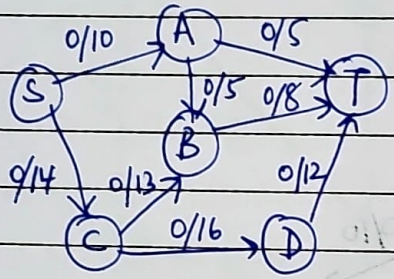
M T W T F S S

Date :

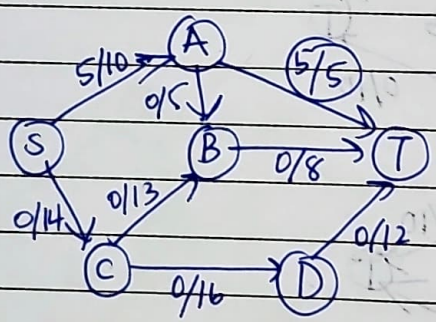
b)

Augmenting path	Bottleneck capacity
S → A → T	5
S → A → B → T	5
S → C → D → T	12
S → C → B → T	2
Max flow	24

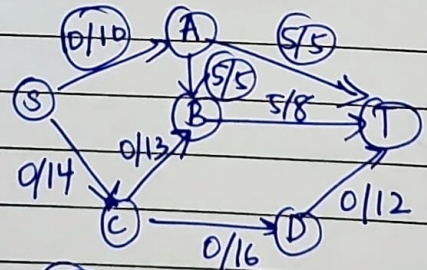
1



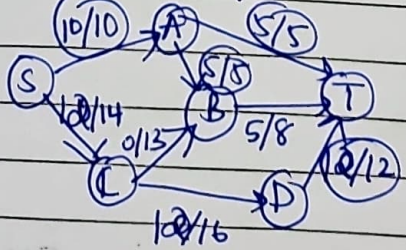
2



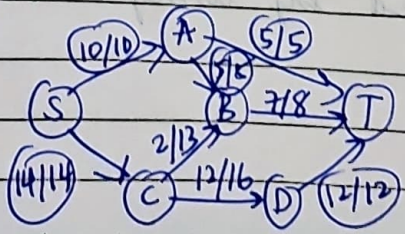
3



4



5



M T W T F S S

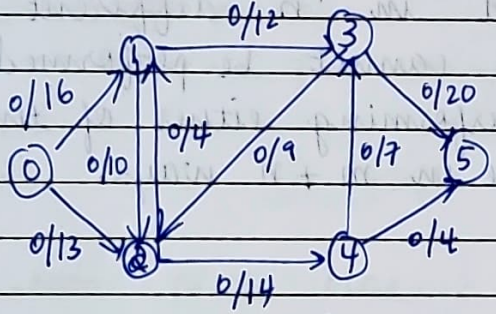
COMPASS

Date :

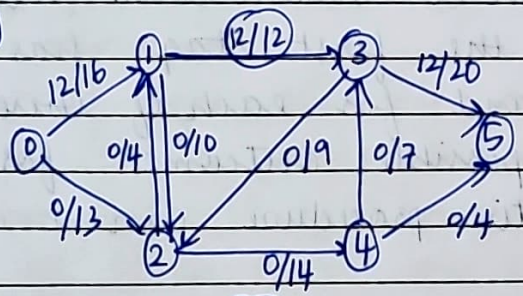
c)

Augmenting path	Bottle neck capacity
$0 \to 1 \to 3 \to 5$	12
$0 \to 2 \to 4 \to 3 \to 5$	7
$0 \to 2 \to 4 \to 5$	4
Max flow	23

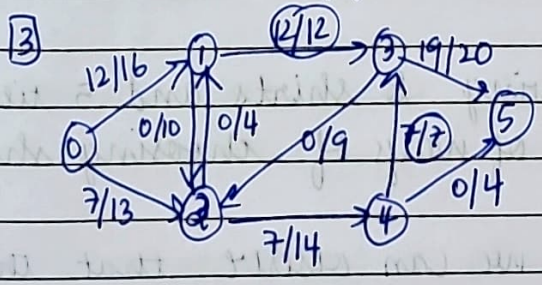
1



2



3



4