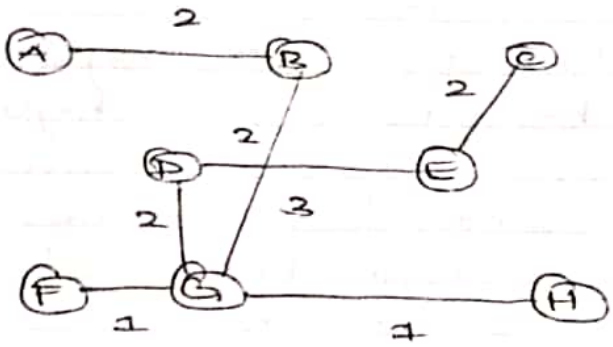
# ASSIGNMENT - 3

MADHAV T
B008149 16

1) i) PRIMS ALGORITHM :

Minimum spaning tree of prim's Algorithm is as follows:




Total weight = 13

order of added nodes are

A B G H F D E C

ii) KRUSKAL'S ALGORITHM :

sorted edges of above graph is

FG, GH = 1

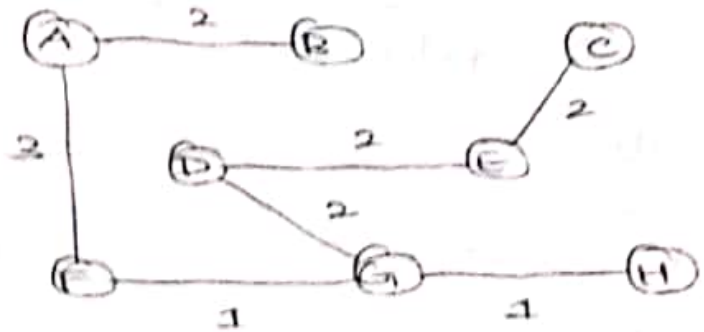
AB, DE, DG, EC } = 2

AF, BG } = 3

CH, BE, DF } = 4

BC = 5

Minimum spanning tree can be written as.




total height = 12

order of edges are

FG, GH, AB, DE, DG, EC, AF

c(MST) = 12 // cost of MST

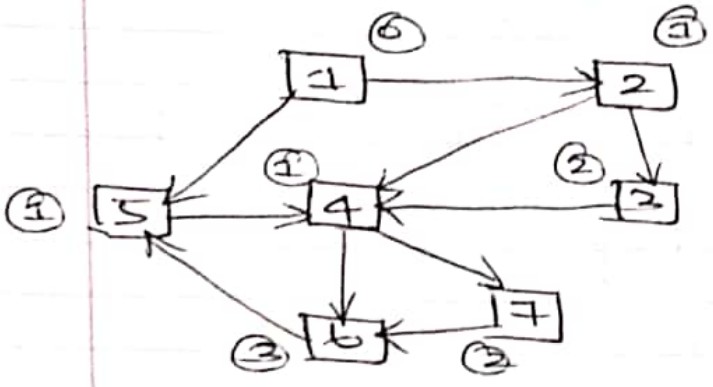
2> a) Algorithm to determine if cyclic is as follows:

```
Algorithm isCyclic () {
  // Mark all vertices as not visited.
  Boolean [] visited = new Bool ;
  for ( int i = 0; i<V ; i++)
      visited [i] = false;
  // call the recursive () to detect
  cycle in DFS.
  for( u=0 ; u<V ; u++) {
  // don't recur if already visited.
  if (! visited [u])
      if ( is cyclic)
          return true ;
  return false ;
}
```

$x_1 \rightarrow y_1 , y_2$
$x_2 \rightarrow y_1 , y_2$
$x_3 \rightarrow y_2 , y_3$

* For the two graphs listed, since the data is non-linear, trees can be used as DS. As it is best suited. (or) doubly linked list. as connected to each node head

b) order of nodes when applying BFS.




Added

1

2 4 5

4 5 3 7 6

Removed

3 7 6

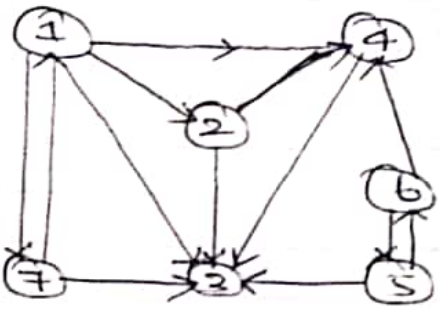
7 6

6

order of nodes visited are

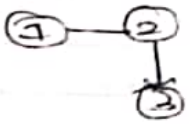
1 2 4 5 , 3 7 6

3> GRAPH



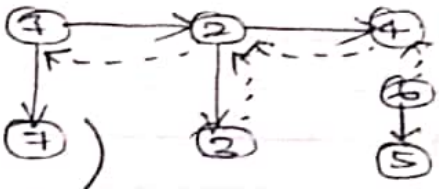

Applying Depth first search,

1 → 2 = New Edge




= New Forwarded edge

So,




New Forwarded edge

Back edge is denoted by dotted lines

Hence DFS sequence is ⇒ 1 2 3 4 7.

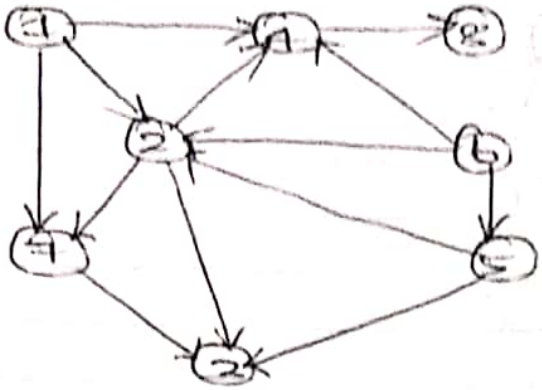
Node 5 and 6 does not be visited / Reached.

Tree edges = { (1,7) (1,2) (2,4), (2,3) (6,5) }

Back edges = (7,1) (5,6)

Fwd edges = (1,3) (1,4)

4) Topology sort :

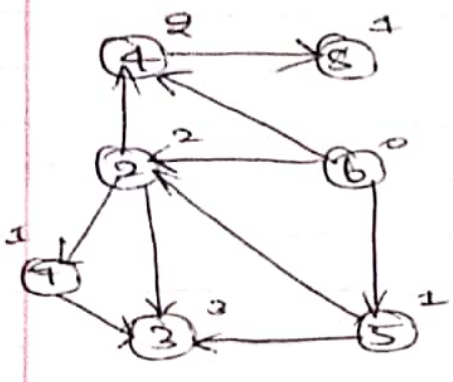



Topology sorting is given as follows.

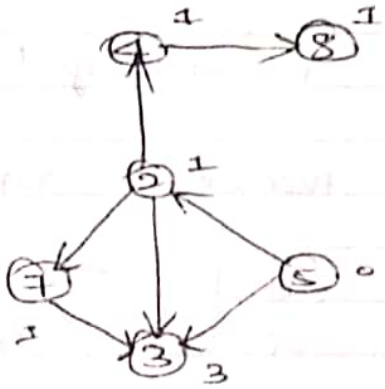
We start from node (1)

Degree's of each node will be updated in every step and mentioned along the node.

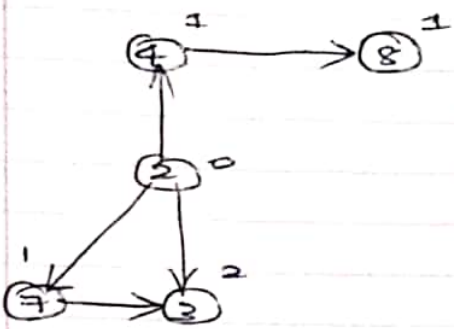
After removing node (1)



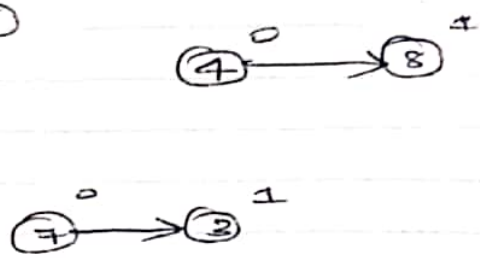

Removing → node (6)

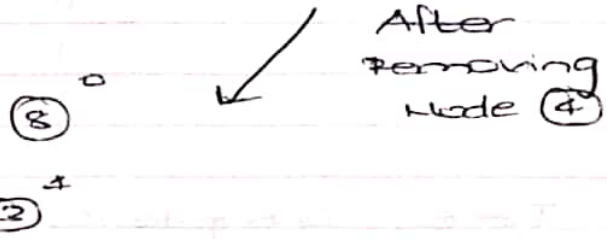


After Removing Node (5)



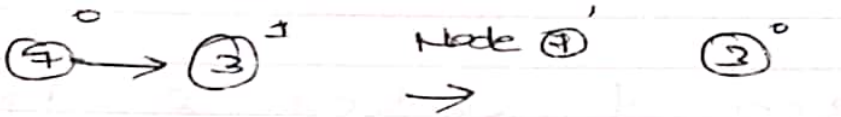
Removing → Node 2





After Removing Node (4)

Removing Node (8)



Node (1) →

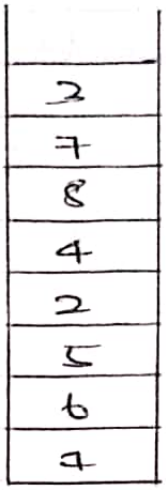
Hence the sequence is stored in a ~~state~~ stack.

order in which it is removed,

⇒ 1 6 5 2 4 8 7 3

stack will be stored as FILO

Stack for the topological sort can be formed as,




→ 1st element.

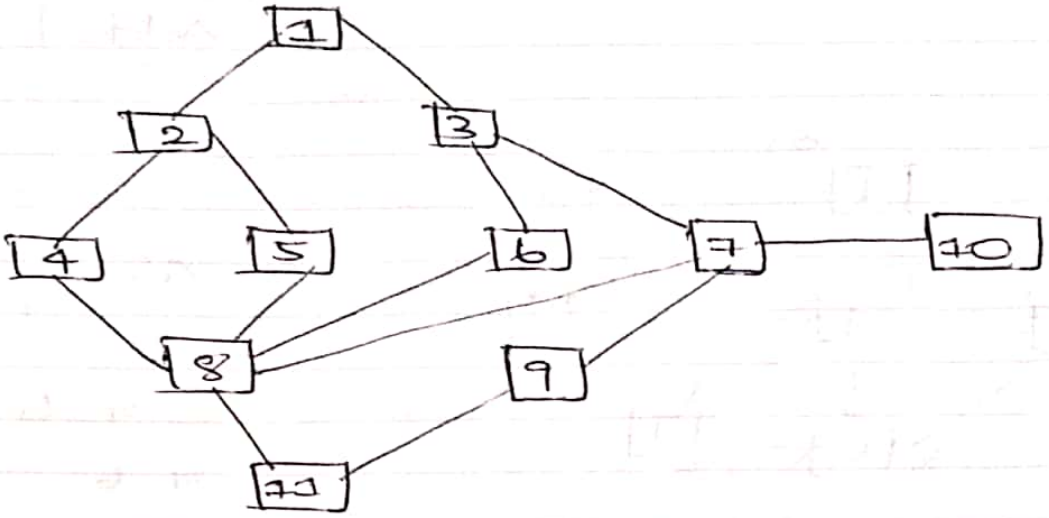
Time complexity for the topological sort is

$$O(V+E)$$

V + E → sum of vertices & edges.

* Sequence of the node found by the application will be

5) BIPARTITE GRAPH

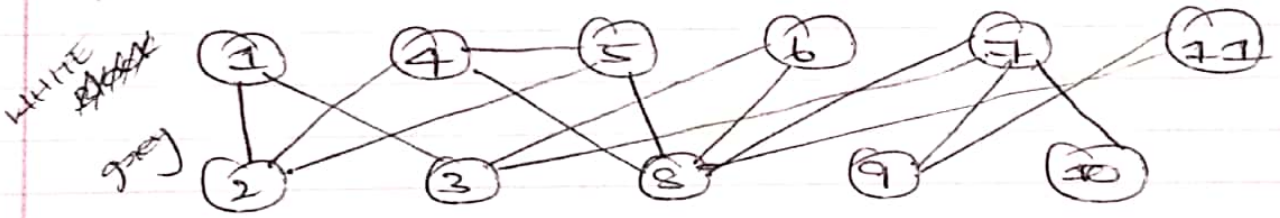



Let $V_1$ = 1, 4, 5, 6, 7, 11

$V_2$ = 2, 3, 8, 9, 10

After applying bipartite.

R/W



EXAMPLE:

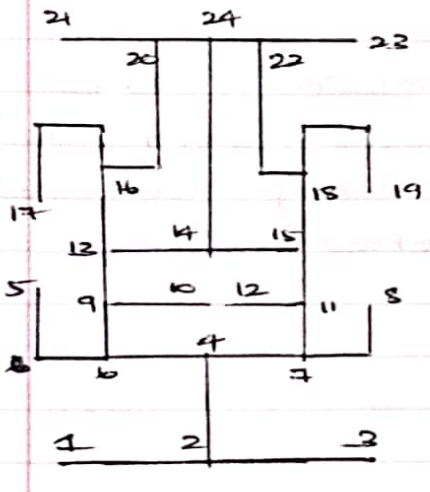
ALGORITHM:

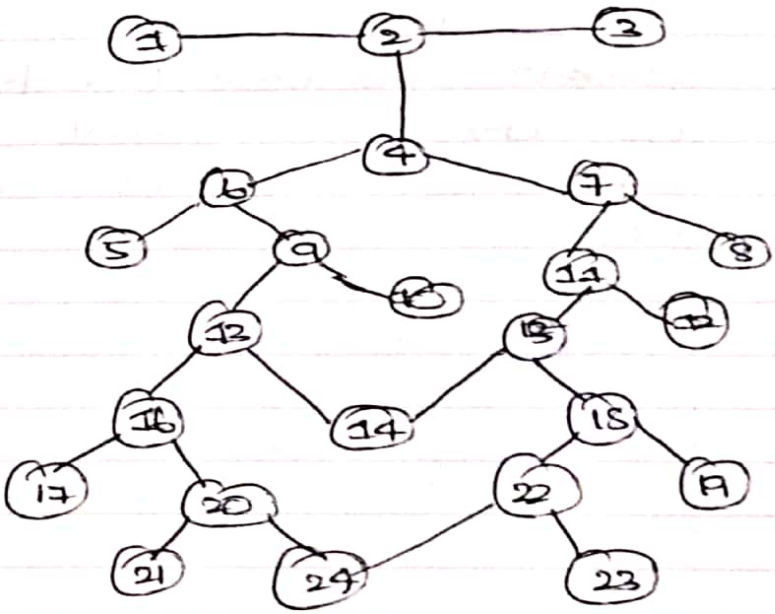
```
For each vertex u in V[G] - {s}
 do colour[u] ← WHITE
    do [u] ← ∞
    colour[s] ← GRAY
    position [s] ← 1
     d[s] ← 0
     Q ← [s]
    while Q is ! empty
    if position [u] ≤ position[v]
        return 0
    else if
          colour [v] ← white
             colour [v] ← grey
              d[u] = d [u] + 1
    Return 1.
```

b) MAZE :

a) Graph for the following Maze :

We are given a Maze, constructing a graph with node numbering system from the entrance point and ending at exiting point of the maze.



b) DFS:

Depth first search is more efficient for passing through the maze because there are more possibilities of finding a path if we explore each node. than visiting each neighbouring node. and then returning back to the previous junction.

When DFS traverses the next vertex (node) it is connected to the current node by an edge, whereas it is not the case in BFS.

7> MERCHANT PROBLEM

Given

profit ($250) = $ 45
profit ($400) = $ 50

Total demand = 250 units
Total amount = $ 70,000

$ 250 = x & $ 400 = y

$$P = 45x + 50y \rightarrow (1)$$

$$x + y \leq 250 \rightarrow (2)$$

$$250x + 400y \leq 70,000 \rightarrow (3)$$

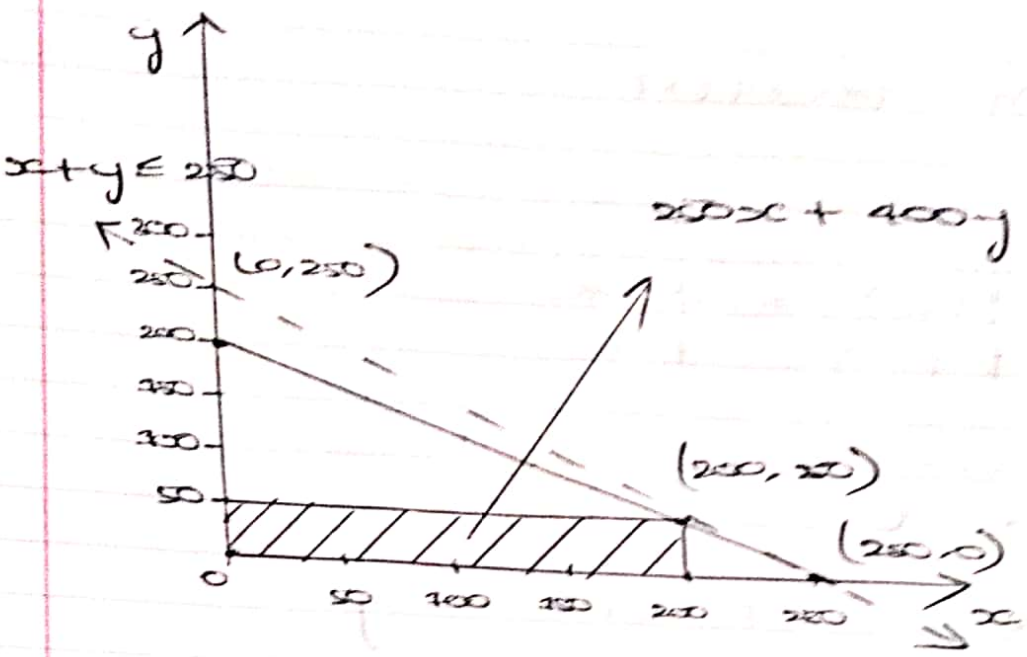
solving eq (2) & (3) we get

$x = 200$ & $y = 50$ to get the Maximum profit.

For graphical representation,

points to consider are

(0, 0) (250, 0) (200, 50)




At $(0, 0)$ $= 45x + 50y \Rightarrow 0$

At $(250, 0)$ $= 45x + 50y \Rightarrow \$11,250$

At $(200, 50)$ $= 45x + 50y \Rightarrow \$11,250$

So the merchant should opt for 200 units of model 1 and 50 units of model 2 to get maximum profit.

8) GREEDY ALGORITHM :

Given:

No of files = n
size of each CD = B bytes
size of each file i = $s_i$ bytes ($s_i < B$)

The files needs to be stored in minimum number of CD's, which can be done using greedy algorithm.

```
Algorithm storage {

    A[] : // Array to store files
    sort A[] in descending order
    A[] = { 1, 2, 3 ... n files }

    CD1 ← A[]
    check ( A[2] to A[n] ) // for loop
{
    If any file can be made available
    with size equal to the left size in
    CD1,
    store A[1] in CD1
     Else
          create CD2
          A[2] ← CD2
}
    Return no of CD's CDi.
}
```

EXAMPLE

If we have files 3, 2, 2, 1 and Max disk size is 4.

1st disk we fill 3.

Since we can't fill ② in 1st disk. We move to 2nd disk and fill.

Then we fill next ② in disk 2 as there is space

Then at last we fill 1 in disk 2.