COLLEGE OF MANAGEMENT & INFORMATION TECHNOLOGY

BACHELOR IN INFORMATION TECHNOLOGY

Submitted by:

ISHWOR KHATIWADA

Year/ Semester: 1st Semester

LCID: LC00017000772

Submitted to:

Lincoln University College

Date: 03-07-2020

Q.N.1

Define contradiction and tautology. Show that the compound proposition is tautology $[p \wedge (p \rightarrow q)] \rightarrow q$

Solution: A compound proposition is said to be a contradiction if its value of truth is always false regardless of the value of truth of its constituent proposition.

P	~P	P^~P
T	F	F
F	T	F

A compound proposition is said to be tautology if its value of truth is always true regardless of the value of truth of its constituent proposition.

P	~P	Pv~P
T	F	T
F	T	T

$[p \wedge (p \rightarrow q)] \rightarrow q =$

P	q	P→q	P^(p→q)	[p^(p→q)]→q
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Hence, The compound proposition $[p \wedge (p \rightarrow q)] \rightarrow q$ is tautology as its truth value is always true at last.

Q.N. 2 Without expanding show that

$$\begin{vmatrix} 1 & 1 & 1 \\ a & b & c \\ bc & ca & ab \end{vmatrix} = (a-b)(b-c)(c-a)$$

SOLLUTION:

Taking L.H.S

$$\begin{vmatrix} 1 & 1 & 1 \\ a & b & c \\ bc & ca & ab \end{vmatrix}$$

step 1: $R_1 \rightarrow R_1-R_2$

$$\begin{vmatrix} 1-1 & 1 & 1 \\ a-b & b & c \\ bc-ca & ca & ab \end{vmatrix}$$

Now, (a-b) will be common in first row R_1

Step2: $R_2 \longrightarrow R_2-R_1$

$$= (a-b) \begin{vmatrix} \mathbf{0} & \mathbf{1-1} & \mathbf{1} \\ 1 & b-c & c \\ -c & ca-ab & ab \end{vmatrix}$$

Now (b-c)will be common in row R_2

$$= (a-b)(b-c)\begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & c \\ -c & -a & ab \end{vmatrix}$$

Then,

$$\begin{vmatrix} 1 & 1 \\ -c & -a \end{vmatrix}$$

= 1x-a-(-c)x1

= (c-a)

Hence, (a-b)(b-c)(c-a) is what we got therefore,

L.H.S=R.H.S proved

Q.N.3. Define homogeneous recurrence relation and Solve the recurrence relation?

If k1, . . . , kr are constants, a recurrence relation of the form an = k1an−1 + k2an−2 + · · · + kran−r + f (n) is called a linear recurrence relation with constant coefficients of order r.

The recurrence relation is called homogeneous when f (n) = 0. If g(n) is a function such that an = g(n) for n = 0, 1, 2, . . .

$$a_n=6a_{n-1}+8a_{n-2} \text{ for } n\geq 2, a_0=4, a_1=10$$

SOLLUTION:

Given recurrence relation is $a_n=6a_{n-1}+8a_{n-2}$……….(i)

This is the recurrence relation of second order. Its characteristic equation is given by ;

Or, $r^2=6r^1-8r^0$

Or, $r^2-6r+8=0$

Or, $r(r-4)-2(r-4)=0$

Or, $(r-4)(r-2)=0$

Either, $r=2$, Or $r=4$ i.e. $R_1=2$; $R_2=4$

The solution of the eqn(i) is ,

$a_n= \alpha_1 r_1^n+\alpha_2 r_2^n$

$a_n= \alpha_1 2^n+\alpha_2 4^n$………………….(ii)

Put n=0 in eqn(ii), we get

$a_0=\alpha_1 2^0+\alpha_2 4^0$

Given $a_0=4$, then

Or, $4=\alpha_1+ \alpha_2$

Or, $\alpha_2=4-\alpha_1$

NOW, put n=1

Or, $a_1=\alpha_1 2^1+\alpha_2 4^1$

Given $a_1=10$

Or, $10= \alpha_1 2+(4-\alpha_1)4$ Or, $5=\alpha_1+8-2\alpha_1$

Or, $\alpha_1=3$, $\alpha_2=1$

Hence, the equation is $a_n=3*2^n+1*4^n$

Q.N. 4

Define the term tree, minimum spanning tree, simple, multiple and pseudo graph. Write prim's and kruskal's algorithm to construct minimum spanning tree. Explain the process of generating spanning tree using kruskal's algorithm? Assume example of your own.

SOLLUTION:

A Tree is an undirected acyclic graph with Each pair of vertices in G having a unique direction. A tree with N number of vertices contains (N−1) number of edges. The vertex of 0 degree is called tree base. The vertex of 1 degree is referred the tree's leaf.

A spanning tree with a weight given less than or equal to the weight of every spanning tree of a weighted, connected and undirected graph G is named a minimal spanning tree.

A graph which has neither loops nor multiple edges is called a simple graph, i.e. where each edge connects two distinct vertices and no two edges link the same pair of vertices or make circular path.

Any graph containing certain multiple edges is called a multiple graph. No loops are enabled in a Multi-graph.

A graph where it allows loops and multiple edges is called a pseudo graph.

The Prims algorithm is considered a Greedy algorithm. It begins with an empty tree spanning out. The idea is to keep the vertices in two sets. The first set contains the vertices already included in the Minimum Spanning Tree, while the other set contains the not yet included vertices. At each step, it considers all the edges connecting the two sets, and from these edges it picks the minimum weight edge.

ALGORITHM:

Step 1Detach all the parallel loops and edges

Step 2Pick the root node for every random node

Step 3Check outgoing edges and choose the one with lower cost

Kruskal's Algorithm

Kruskal 's algorithm is a greedy algorithm, which finds a minimum spanning tree for a weighted graph connected. It requires a tree of that graph that contains every vertex, and the cumulative weight of all the edges of the tree is less than or equal to every spanning tree conceivable.

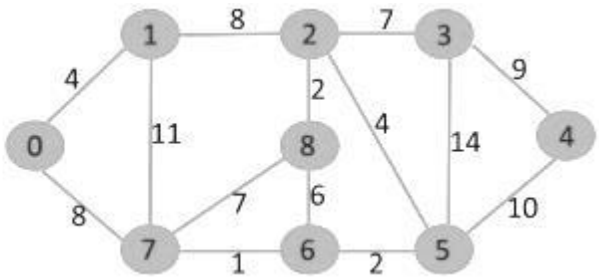
Algorithm

Step 1 – at first Arrange all the edges of the given graph G(V,E) in ascending order as per edge weight of them.

Step 2 − Choose the smallest weighted edge from the graph and check if it forms a cycle with the spanning tree formed so far or not .If there is no cycle, include this edge to the spanning tree else discard it.

Step 4 − Repeat Step 2 and Step 3 until (V−1) number of edges are left in the spanning tree which will be final answer.

Example:



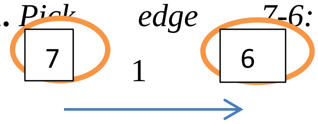
The graph contains 9 vertices. So, the minimum spanning tree formed will have (9 – 1) = 8 edges.

Weight	from	to
1	7	6
2	8	2
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7

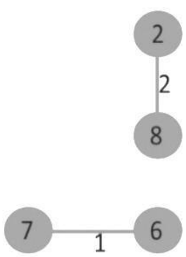
8	1	2
9	3	4
10	5	4
11	1	7
14	3	5

Now pick all edges one by one from sorted list of edges

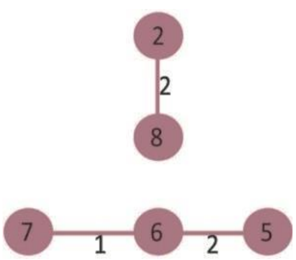
1. *Pick edge 7-6:* cycle is Not formed so include it.



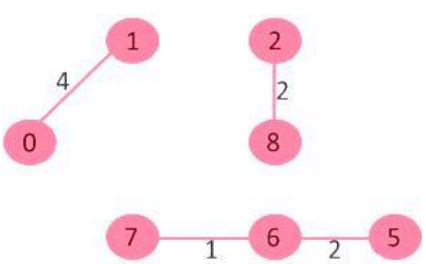
. *Pick edge 8-2:* No cycle is formed, include it.



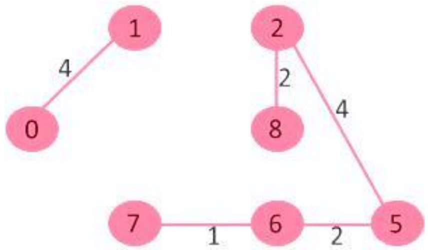
3. *Pick edge 6-5:* No cycle is formed, include it.



4. *Pick edge 0-1:* No cycle is formed, include it.

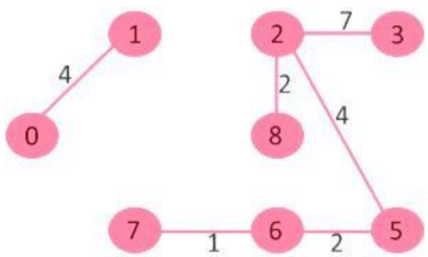


5. *Pick edge 2-5:* No cycle is formed, include it.



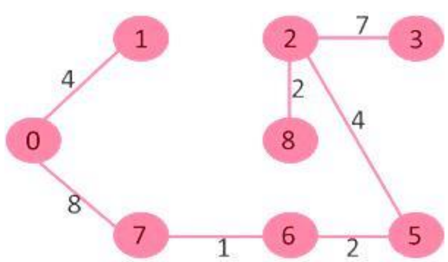
6. *Pick edge 8-6:* Since including this edge results in cycle, discard it.

7. *Pick edge 2-3:* No cycle is formed, include it.



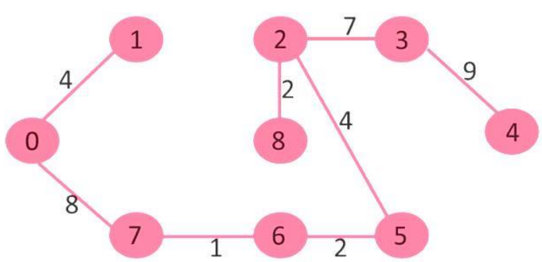
8. *Pick edge 7-8:* Since including this edge results in cycle, discard it.

9. *Pick edge 0-7:* No cycle is formed, include it.



10. *Pick edge 1-2:* Since including this edge results in cycle, discard it.

11. *Pick edge 3-4:* No cycle is formed, include it.



the number of edges =(V – 1), the algorithm stops here.