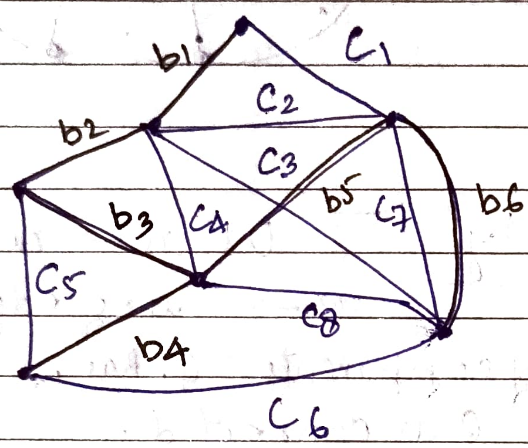
SPANNING TREES :- [skeleton or scaffolding] [maximal tree subgraph or maximal tree of G]

A tree T is said to be spanning tree of a connected graph G is T is subgraph of G and T contains all vertices of G.

→ A disconnected graph with K components having a spanning forest consisting of K spanning trees.



Theorem :- Every connected graph has atleast one spanning tree.

→ An edge in a spanning tree T is called branch of T [b1, b2, b3, b4, b5, b6]

→ An edge of G that is not in given spanning tree T is called a chord [or link or tie]

[C1, C2, C3, C4, C5, C6, C7, C8] Chord Set

$T \cup \bar{T} = G$, T → spanning tree

$\bar{T}$ → Complement of T in G.

Theorem :- A connected graph of n vertices and e edges has n-1 tree branches & e - n+1 chords.

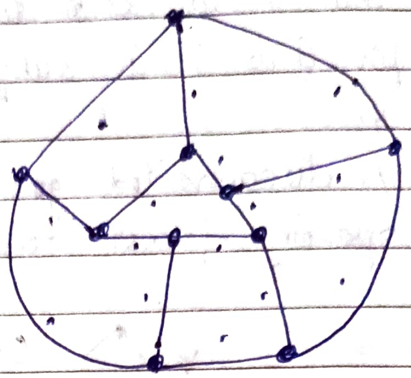
Date :
Page :

$n = 7$, $e = 14$, has 6 tree branches & eight chords

Ex

Here $n = 10$,
$e = 15$,

$e - n + 1$
$= 15 - 10 + 1$
$= 6$ [chords]



& 9 branches.

→ **Rank And Nullity** :→ Let n → no. of vertices
e → no. of edges in G.

Then K, the no of components G has.
If $K = 1$, G is connected.

→ Since every component of a graph must have at least on vertex, $n \geq k$.

~~Moreover~~ Moreover, $e \geq n - k$.

Apart from the constrains $n - k \geq 0$ &
$e - n + k \geq 0$

So these three numbers are independent they are fundamental no. in a graph. From these numbers are derived two other component numbers called rank & nullity

rank $r = n - K$
nullity $\mu = e - n + k$

→ The rank of connected graph is $n - 1$ & the nullity, $e - n + 1$.

Date :
Page :

rank of G = no. of branches in any spanning tree (or forest) of G.

nullity of G = number of chords in G.

rank + nullity = number of edges in G.

The nullity of a graph is also referred to as its cyclomatic number, or first Betti ~~no~~ number.

FUNDAMENTAL CIRCUITS :-

Theorem :- A connected graph G is a tree if and only if adding an edge between any two vertices in G creates exactly one circuit.

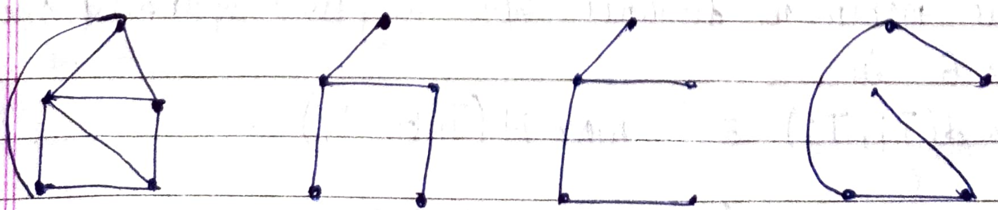
→ Adding any one chord to T will create exactly one circuit. Such circuit, is called a fundamental circuit.

no. of fundamental ~~cir~~ circuit = no. of chords

$= e - n + k$.

FINDING ALL SPANNING TREES :-

The distance b/w two spanning trees T_i & T_j of a graph G is defined as the ~~no~~ number of edges of G present in one tree but not in other.



T_1 T_2 T_3

Date :
Page :

$d(T_2, T_3) = 3$.

$d(T_i, T_j) = \frac{1}{2} N(T_i \oplus T_j)$

$N(g) \rightarrow$ no. of edge in a graph G.

$T_i \oplus T_j \rightarrow$ subgraph of G containing all edges of G that are either in T_i or T_j but not in both.

$T_i \oplus T_j = 3$

$N(g) = 4$

$\therefore d(T_i, T_j) = \frac{1}{2} \times 4 \times 3 =$

SPANNING TREE IN A WEIGHTED GRAPH :-

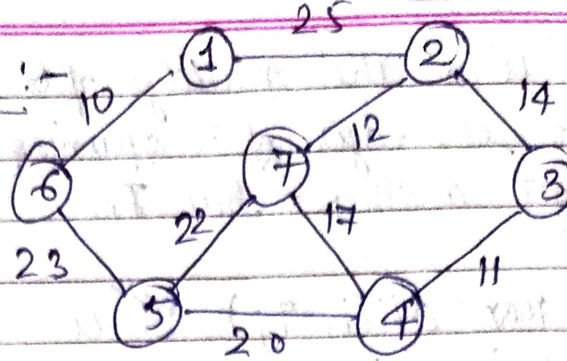
A spanning tree or shortest - distance spanning tree or minimal spanning tree with the smallest weight in a weighted graph.

Algorithm shortest spanning tree

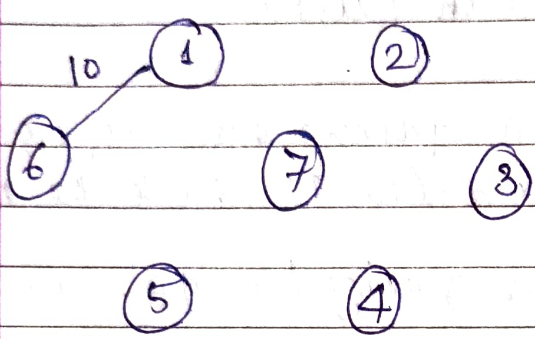
→ Kruskal

→ Prim

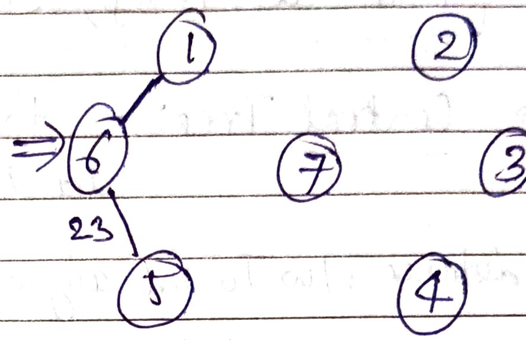
Prim's Algorithm :-



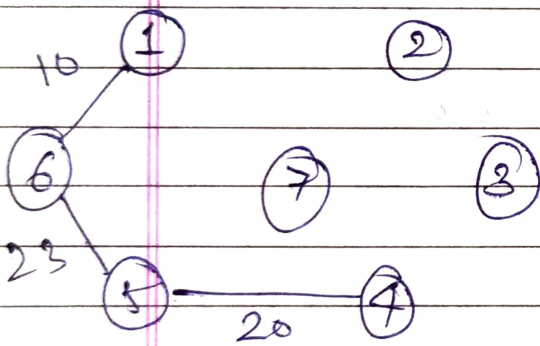
We will consider all the vertices first. Then we select edge with minimum weight. The ~~ad~~ algorithm proceeds by selecting adjacent edges with minimum weight.



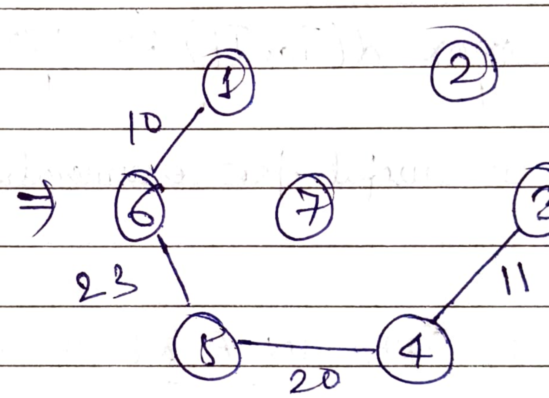
total weight = 10



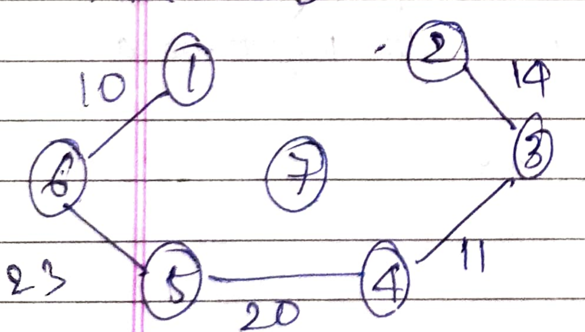
total weight = 33



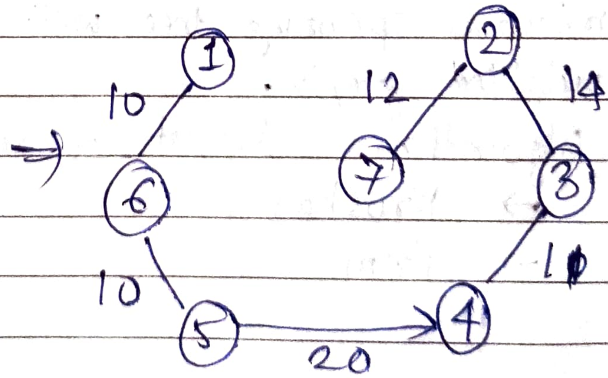
total = 53



total = 84



total = 78



total = 90.

Time taken by for(k=1 to node-1) loop

Analysis $T(n) = \sum_{K=1}^{node-1}\left(\sum_{i=0}^{node-1} 1 + \sum_{i=0}^{node-1} 1\right)$

for (i = 0 to node-1)

for(j = 0 to node-1)

Let node = n

$$T(n) = \sum_{K=1}^{n-1}\left(\sum_{i=0}^{n-1} 1 + \sum_{j=0}^{n-1} 1\right)$$

$$= \sum_{K=1}^{n-1}\left[((n-1)+0+1) + ((n-1)+0+1)\right]$$

$$= \sum_{K=1}^{n-1} 2n = 2n\left[(n-1)-1+1\right]$$

$$= 2n(n-1) = 2n^2 - 2n$$

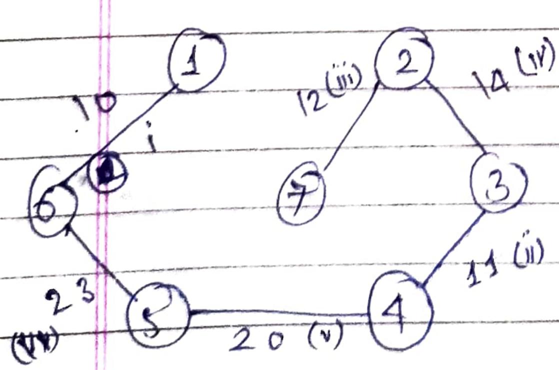
$$T(n) = n^2$$

$T(n) = \Theta(n^2)$ But $n \rightarrow$ total no. of nodes or vertices

Time complexity of Prim's Algo $= \Theta(|V^2|)$

Kruskal's Algorithms :-

First we will select all the vertices. Then an edge with optimum weight is selected from head, even though it is not adjacent to previously selected edge.



total = 10

$\Rightarrow$ total weight = 90

time complexity =

$\Theta(|E| \log |E|)$

$E \rightarrow$ total no. of edges.