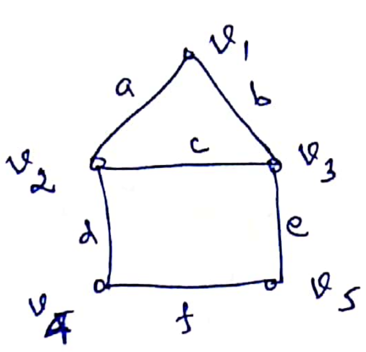
Operations in Graphs

1) Union of Graphs
2) Intersection
3) Ring sum
4) Decomposition
5) Fusion
6) Deletion of a vertex
7) Deletion of an edge

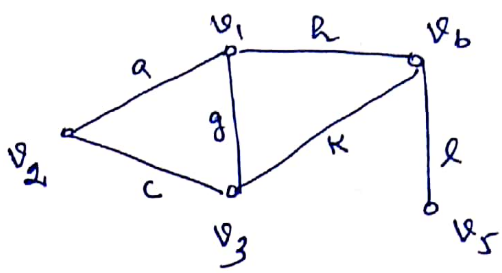
Union:- Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, the union of the two graph is $G_3 = G_1 \cup G_2$ where,
$V(G_3) = V(G_1) \cup V(G_2)$ & $E(G_3) = E(G_1) \cup E(G_2)$.

We can do union operation atleast a point common in G_1 and G_2.

G_1



G_2



$V(G_2) = \{v_1, v_2, v_3, v_5, v_6\}$

$E(G_2) = \{a, c, g, h, K, l\}$

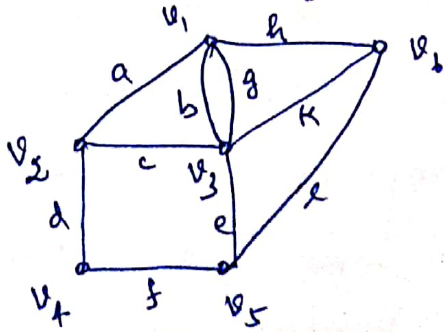
$V(G_1) = \{v_1, v_2, v_3, v_4, v_5\}$

$E(G_1) = \{a, b, c, d, e, f\}$

$G_3 = G_1 \cup G_2$

$V(G_3) = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

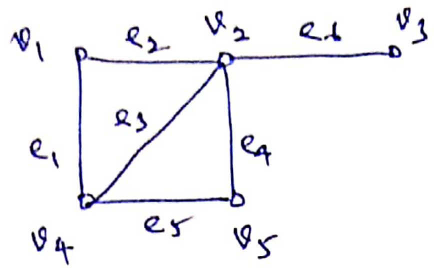
$E(G_3) = \{a, b, c, d, e, f, g, h, K, l\}$



Intersection

let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, the intersection of the two graph is $G_3 = G_1 \cap G_2$, where $V(G_3) = V(G_1) \cap V(G_2)$ and $E(G_3) = E(G_1) \cap E(G_2)$.

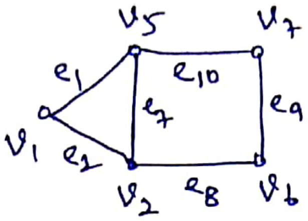
G_1



$V(G_1) = \{v_1, v_2, v_3, v_4, v_5\}$

$E(G_1) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

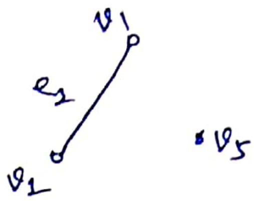
G_2



$V(G_2) = \{v_1, v_2, v_5, v_6, v_7\}$

$E(G_2) = \{e_1, e_2, e_8, e_9, e_{10}, e_7\}$

$G_3 = G_1 \cap G_2$



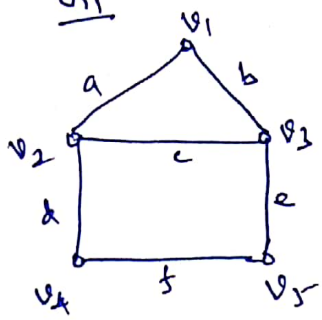
$V(G_3) = \{v_1, v_2, v_5\}$

$E(G_3) = \{e_2\}$

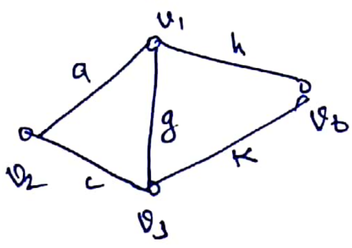
Ring Sum

A ring sum of two graphs G_1 and G_2 written as $G_1 \oplus G_2$ is a graph consisting of the vertex set $V_1 \cup V_2$ and the edge set that are either in G_1 (or) G_2 but not in both.

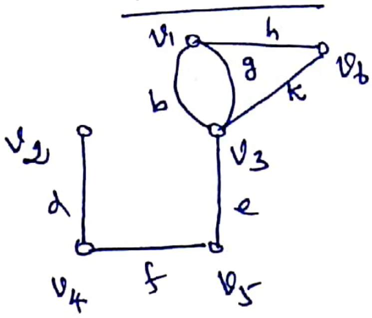
G_1



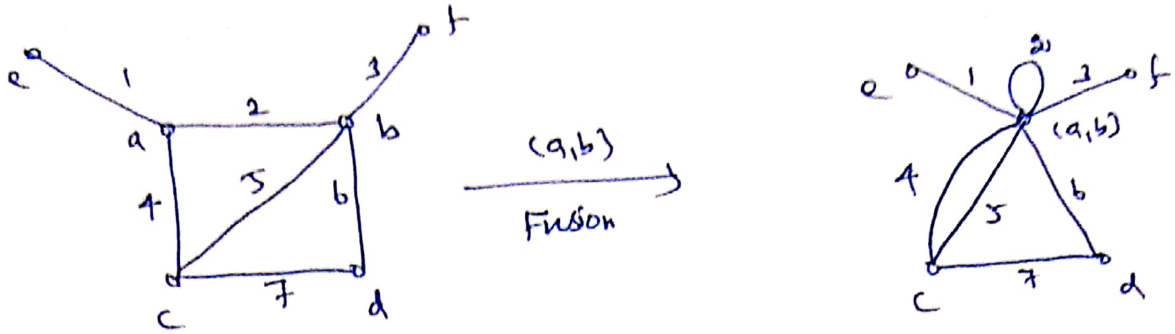
G_2



$G = G_1 \oplus G_2$



Fusion A pair of vertices (a, b) in a graph is said to be fusion (or merged (or) identified) if the two vertices are replaced by a single new vertex such that every edge that was incident on either 'a' (or) 'b' (or) on both is incident on the new vertex.



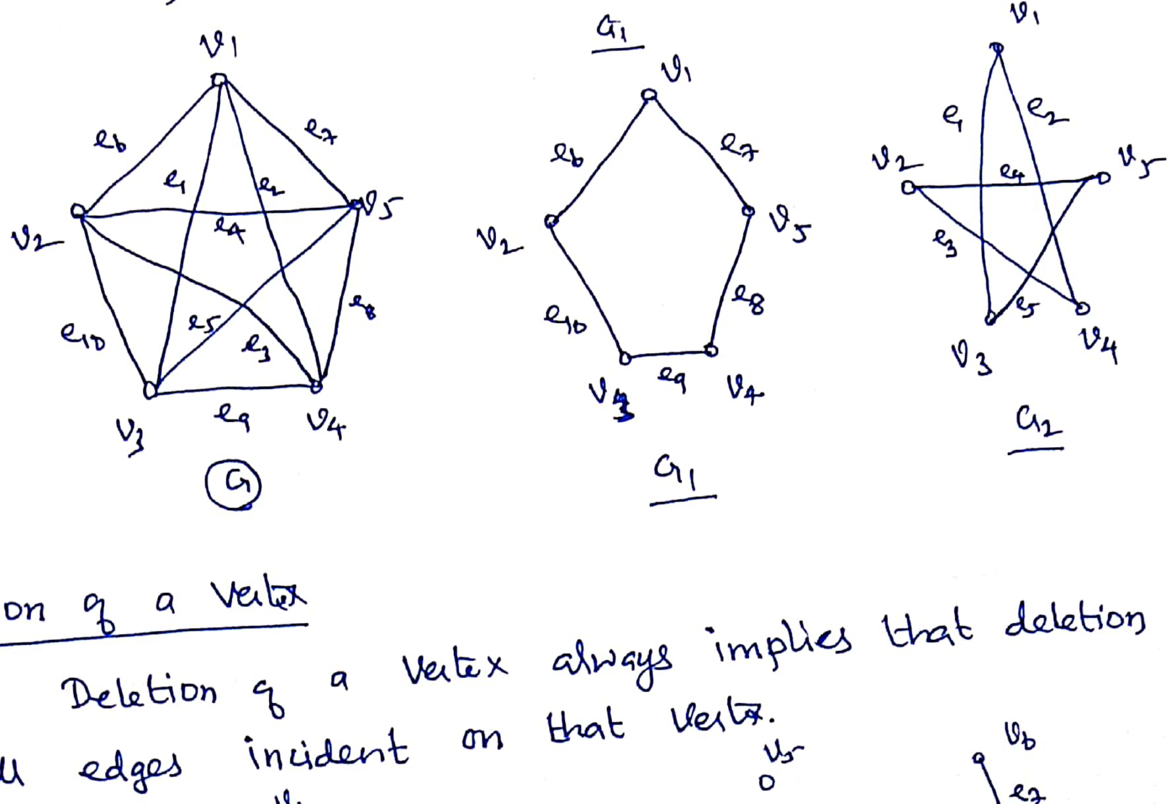
Decomposition:-

A graph G is said to be decomposition into two subgraphs G_1 and G_2, which satisfy the conditions

(i) $G_1 \cup G_2 = G$ (ii) $G_1 \cap G_2$ = Null graph

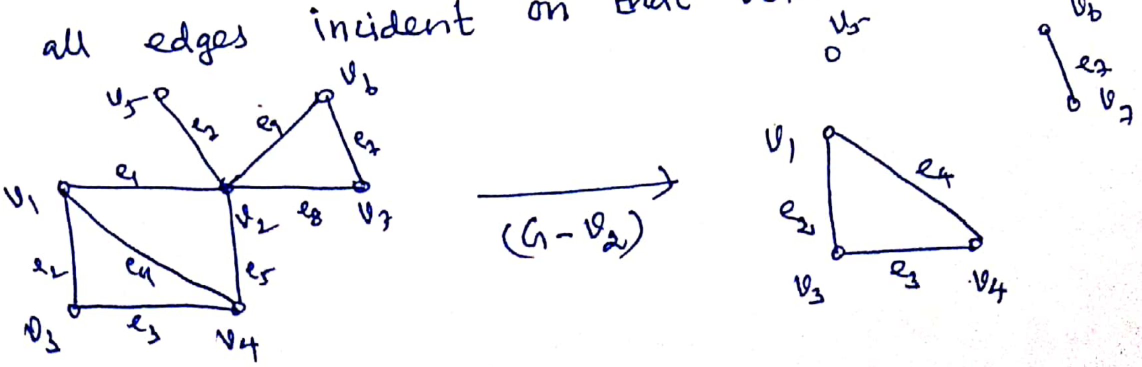
(iii) $V(G) = V(G_1) \cap V(G_2)$ (iv) $E(G) = E(G_1) \cup E(G_2)$

(v) $E(G_1) \cap E(G_2) = \phi$ (empty).



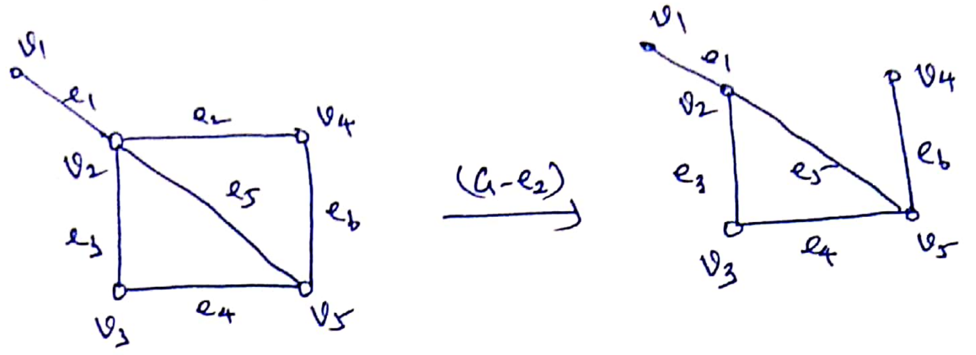
Deletion of a Vertex

Deletion of a vertex always implies that deletion of all edges incident on that vertex.



Deletion of an edge

Let 'e' be an edge in G, the deletion of 'e' which is denoted by $(G-e)$. Deletion of an edge does not imply, deletion of its end vertices.



Shortest Path Algorithms

A graph in which each edge 'e' is assigned a non-negative real number $w(e)$ is called a weighted graph $w(e)$ called the weight of the edge 'e' may represent distance, time, cost etc, in some units.

A shortest path between two vertices in a weighted graph is a path of least weight. In an unweighted graph, a shortest path means one with the least number of edges.

In this section, we shall deal with the problem of finding the shortest path between any two vertices in a weighted graph.

Dijkstra's Algorithm

To find the length (or weight) of the shortest path between two vertices, say a and z, in a weighted graph, the algorithm assigns numerical labels to the vertices of the graph by an iterative procedure. At any stage of iteration, some vertices will have temporary labels (that are not bracketed) and the others will have permanent labels (that are bracketed). Let us denote the label of the vertex v by $L(v)$.

Initial Iteration 0

Let V_0 denote the set of all the vertices v_0 of the graph. The starting vertex is assigned the permanent label (0) and all other v_0's the temporary label ∞ each. Let $V_1 = V_0 - \{v_0^*\}$, where v_0^* is the starting vertex which has been assigned a permanent label.

Iteration 1

Let the elts of V_1 be now denoted by v_1. (The elts v_1 are the same as the elts v_0 excluding v_0^*.) For the elts of V_1 that are adjacent to v_0^*, the temporary labels are revised by using $L(v_1) = L(v_0^*) + w(v_0^* v_1)$, where $L(v_0^*) = 0$, $w(v_0^* v_1)$ is the weight of the edge $v_0^* v_1$ and for the other elts of V_1, the previous temporary labels are not altered. Let v_1^* be the vertex among the v_i's for which $L(v_i)$ is minimum. If there is a tie for the choice of v_1^*, it is broken arbitrarily. Now $L(v_1^*)$ is given a permanent label. let $V_2 = V_1 - \{v_1^*\}$.

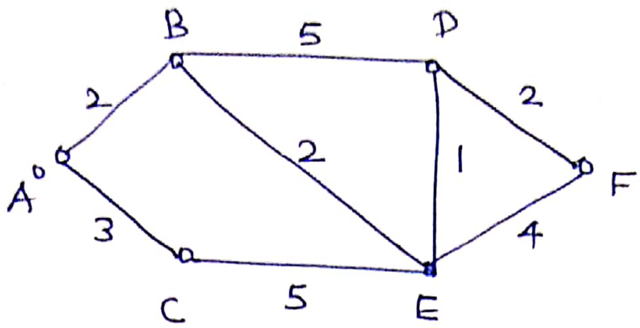
Iteration i

For the elts of V_i that are adjacent to v_{i-1}^*, the temporary labels are revised by using

$$L(v_i) = L(v_{i-1}^*) + w(v_{i-1}^*, v_i)$$ and for the other elts of V_i, the previous temporary labels are not altered. If the temporary label to be assigned to any vertex in the i^{th} iteration is greater than or equal to that assigned to it in the $(i-1)^{th}$ iteration, the previous label is not changed.

The iteration is stopped when the final vertex z is assigned a permanent label eventhough some vertices might not have been assigned permanent labels. The permanent label of z is the length of the shortest path from a to z. The shortest path itself is identified by working backward from z and including those permanently labeled vertices from which the subsequent permanent labels arose.

Example: 1



Iteration Number	Iteration Details	Remarks
0	V_0: A B C D E F $L(V_0)$: (0) ∞ ∞ ∞ ∞ ∞	* A is permanent label and $L(A^*)=0$ is bracketed.
1	V_1: A^* B C D E F $L(V_1)$: — (2) 3 ∞ ∞ ∞	* A is adjacent to B & C $L(B) = L(A^*) + w(A^*B)$ $= 0+2=2$ $L(C) = L(A^*) + w(A^*C)$ $= 0+3 = 3$. B is min so B gets Bracketed.
2.	V_2: A^* B^* C D E F $L(V_2)$: — — (3) 7 4 ∞	* $L(D) = L(B^*) + w(B^*D)$ $= 2+5=7$ $L(E) = 4$ C is not adjacent to B^*, L(C) is brought forward from the previous iteration as 3. Since L(C) is min among L(C), L(D) & L(E), C gets bracketed & $L(C^*)=3$
3.	V_3: A^* B^* C^* D E F $L(V_3)$: — — — 7 (4) ∞	
4.	V_4: A^* B^* C^* D E^* F $L(V_4)$: — — — (5) — 8	$\therefore$ Shortest path $A \to B \to E \to D \to F$ cost 7.
5.	V_5: A^* B^* C^* D^* E^* F^* $L(V_5)$: — — — — — (7)	

Scanned by CamScanner

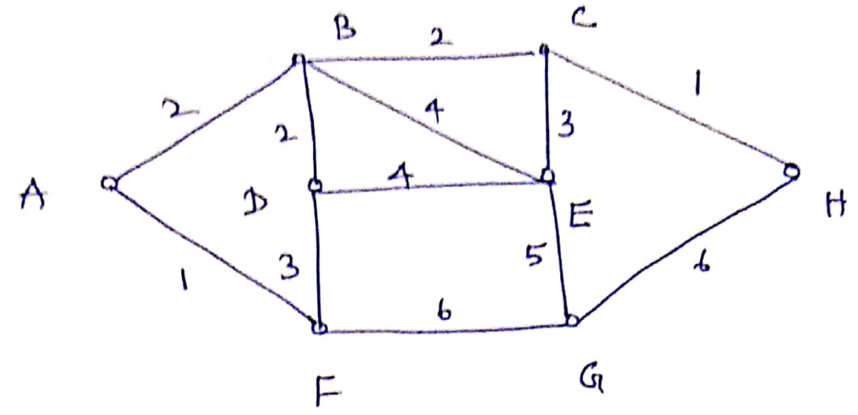
Iteration Number	Iteration Details	Remarks
0.	V_0: A B C D E F $L(v_0)$: (0) ∞ ∞ ∞ ∞ ∞	Initial labels for all the vertices are assumed. A gets the permanent label and $L(A^*) = 0$ is bracketed.
1.	V_1: A^* B C D E F $L(v_i)$: — (2) 3 ∞ ∞ ∞	B and C are adjacent vertices for A^*. $L(B) = L(A^*) + w(A^*B) = 0 + 2 = 2$ $L(C) = L(A^*) + w(A^*C) = 0 + 3 = 3$ Since $L(B) < L(C)$, B gets the permanent label and $L(B^*) = 2$ is bracketed.
2.	V_2: A^* B^* C D E F $L(v_2)$: — — (3) 7 4 ∞	D and E are adjacent vertices to B^*. $L(D) = L(B^*) + w(B^*D) = 2 + 5 = 7$ $L(E) = L(B^*) = w(B^*E) = 2 + 2 = 4$ Since C is not adjacent to B^*, $L(C)$ is brought forward from the previous iteration as 3. Since $L(C)$ is minimum among $L(C)$, $L(D)$ and $L(E)$, C gets the permanent label and $L(C^*) = 3$ is bracketed.
3.	V_3: A^* B^* C^* D E F $L(v_3)$: — — — 7 (4) ∞	D and F are not adjacent to C^*. So $L(D)$ and $L(F)$ are brought forward from iteration (2). $L(E) = L(C^*) + w(C^*E) = 3 + 5 = 8$ Since the revised $L(E) >$ the previous $L(E) >$ the previous value of $L(E) = 4$ is retained. Now E gets the permanent label and $L(E^*) = 4$ is bracketed.
4.	V_4: A^* B^* C^* D E^* F $L(v_4)$: — — — (5) — 8	D and F are adjacent to E^* $L(D) = L(E^*) + w(E^*D) = 4 + 1 = 5$ $L(F) = L(E^*)\ w(E^*F) = 4 + 4 = 8$ Since $L(D) < L(F)$, D gets the permanent label and $L(D^*) = 5$ is bracketed.
5.	V_5: A^* B^* C^* D^* E^* F $L(v_5)$: — — — — — (7)	Since F is the only vertex adjacent to D^* and since $L(F) = L(D^*) + w(D^*F) = 5 + 2 = 7$, the final vertex F gets the permanent label and $L(F^*) = 7$ is bracketed.

Since $L(F^*) = 7$, the length of the shortest path from A to $F = 7$.

To find the shortest path, we work backward from F explained as follows: F became F^* from D^* in iteration (5); D became D^* from E^* in iteration (4); E became E^* from B (but not from C), as $L(E) = L(E^*)$ assumed the label 4 in iteration (2) itself; B became B^* from A^* in iteration (1).

Hence, the shortest path is $A - B - E - D - F$.

(2)



		A	B	C	D	E	F	G	H	Adjacent vertices of latest v^*.
0	V_0 :	A	B	C	D	E	F	G	H	
	$L(V_0)$:	(0)	∞	∞	∞	∞	∞	∞	∞	B & F
1	V_1 :	A^*	B	C	D	E	F	G	H	
	$L(V_1)$:	—	2	∞	∞	∞	(1)	∞	∞	D & G
2	V_2 :	A^*	B	C	D	E	F^*	G	H	
	$L(V_2)$:	—	(2)	∞	4	∞	—	7	∞	C, D & E
3.	V_3 :	A^*	B^*	C	D	E	F^*	G	H	
	$L(V_3)$:	—	—	(4)	4	6	—	∞	∞	E & H
4.	V_4 :	A^*	B^*	C^*	D	E	F^*	G	H	
	$L(V_4)$	—	—	—	∞	7	—	∞	(5)	—

~~H → C → H~~

H is reached from C, C is reached from B & B from A. ∴ A → B → C → H is shortest path.

Length = $w(AB) + w(BC) + w(CH)$

$= 2 + 2 + 1 = 5$ //.