• Dijskstra's Algorithm :-

→ This is other kind of shortest path problem.

→ Here there exists source vertex from where the path is to be started & calculates its shortest path from all vertices to reach the destination vertex.

→ This algorithm is given by E.W. Dijkstra and is called Dijskstra's Algorithm.

→ The steps to be followed in this dijskstra's Algorithm are,

i) Initially mark all distances from the source vertex/node as '∞' except the source vertex..., ie, mark the distance of source vertex/node as '0'.

2) Then check all the possible outgoing edges of source node and calculate their distances as,

(current length of source vertex to next vertex) + (Prev. (source vertex) distance) < its previous distance

ie, let V_1 be source vertex & V_2, V_3 are its attached nodes or outgoing edges.

$$\left[\text{length}(V_1 \text{ to } V_2) + V_1 \text{ distance (Prev.)}\right] < \text{length}[i]$$

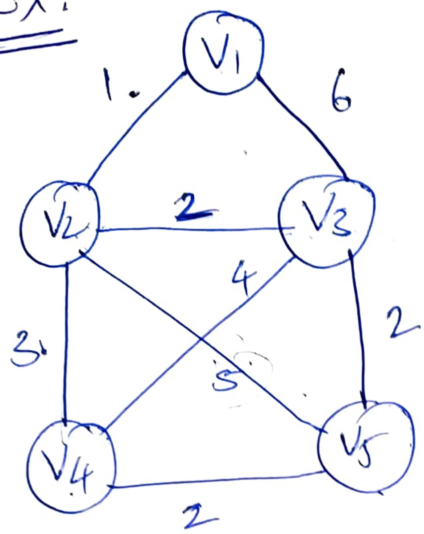
where 'i' is the prev. distance of that node which was calculated already.

3) Then mark the node V_1. Also if above condition is true then LHS part is taken as distance or RHS part is taken.

4) Continue the same method till it reaches all nodes of graph & find shortest distance & path from final table

(14. A)

Ex:-



step 1:- Initially set all the vertices length to '∞' except the first node in graph (take it as 0).

status	node/vertices	length/distance
→	V_1	0
	V_2	∞
	V_3	∞
	V_4	∞
	V_5	∞

step 2:- check all outgoing edges of node V_1 (status indicated node)

status	node	distance
→	V_1	0
	V_2	$1+0<\infty$
	V_3	$6+0<\infty$
	V_4	∞
	V_5	∞

{∵ $V_1 - V_2$ it is 1 and prev. V_1 value is 0 and ~~prev~~ value is ∞).

⇓

status	node	distance
✓	V_1	0
→	V_2	1
	V_3	6
	V_4	∞
	V_5	∞

Step 3 :- Now v_1 node is marked. Then check all outgoing edges of node v_2 (ie, the next least).

14.B

status	node	distance
✓	v_1	0
→	v_2	1
	v_3	$2+1 < 6$ (T)
	v_4	$3+1 < \infty$
	v_5	$5+1 < \infty$

⟹

status	node	distance
✓	v_1	0
✓	v_2	1
→	v_3	3
	v_4	4
	v_5	6

Step 4 :- Now v_1, v_2 are marked. Then check all outgoing edges of node v_3 (ie, next least).

status	node	distance
✓	v_1	0
✓	v_2	1
→	v_3	3
	v_4	$4+3 < 4$ (F) $\quad 7<4$ (F)
	v_5	$2+3 < 6$. (T) $\quad 5<6$ (T)

⟹

status	node	distance
✓	v_1	0
✓	v_2	1
✓	v_3	3
→	v_4	4
	v_5	5 ~~6~~

Step 5 :- Now v_1, v_2, v_3 are marked. Then check all outgoing edges of node v_4 (ie, next least).

status	node	distance
✓	v_1	0
✓	v_2	1
✓	v_3	3
→	v_4	4
	v_5	$2+4 < 5$ (F) $6 < 5$ (F)

⟹

status	node	distance
✓	v_1	0
✓	v_2	1
✓	v_3	3
✓	v_4	4
✓	v_5	5

So finally the shortest path for vertex 5 from source vertex 1 is of length '5' ie, $V_1 - V_2 - V_3 - V_5$ ✓