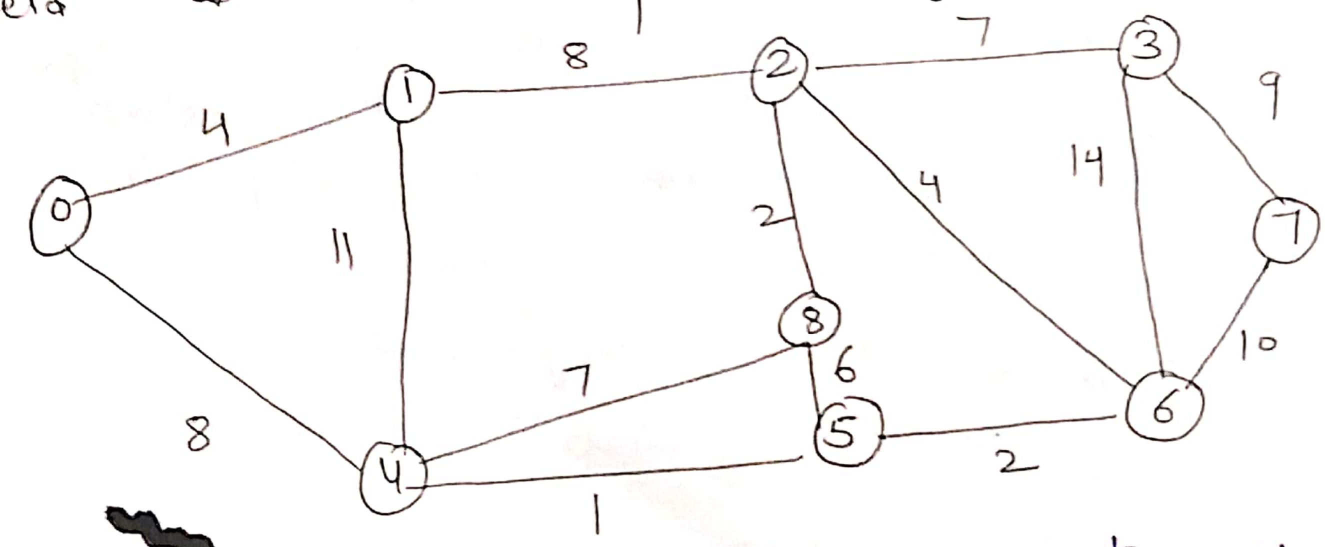
Djikstra Algorithm
(Single source shortest Path)

Shortest path in weighted graphs

Er. Meenakshi Rana

→ Djkstra algo is a single - source shortest path algorithm. Here, single source means that only one source is given and we have to find the shortest path from source to all nodes.

Let's ~~us~~ understand from the given example :→

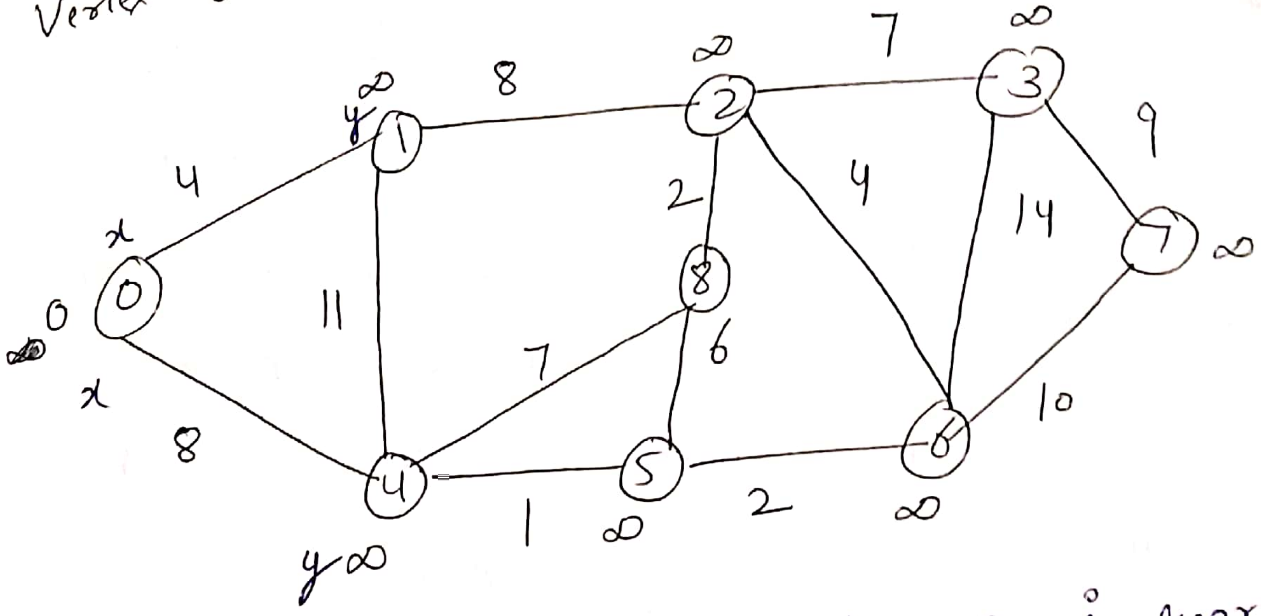


[redacted]

first we have to consider any vertex as a source vertex. Suppose we consider vertex 0 as source Vertex.

① Here we assume that 0 as a source vertex and distance to all other vertices as ∞.

② Two vertices are directly
Vertex 0.



③ Let us assume that the vertex 0 is represented by 'x' and vertex 1 is represented by 'y'. The distance b/w the vertices can be calculated by using the below formula:

$$d(x,y) = d(x) + c(x,y) < d(y)$$
$$= (0+4) < \infty$$
$$= 4 < \infty$$

Since $4 < \infty$, so will update $d(y)$ from ∞ to 4.

→ Now we will consider vertex 0 same as 'x' and vertex 4 as 'y'

$$d(x,y) = d(x) + c(x,y) < d(y)$$
$$= (0+8) < \infty \quad = 8 < \infty$$

ierefore the value of d(y) is 8.

We will replace the infinity values of vertices 1 and 4 with the values 4 and 8 respectively.

→ Now we have to found the shortest path from the vertex 0 to 1 and 0 to 4. Therefore, vertex 0 is selected.

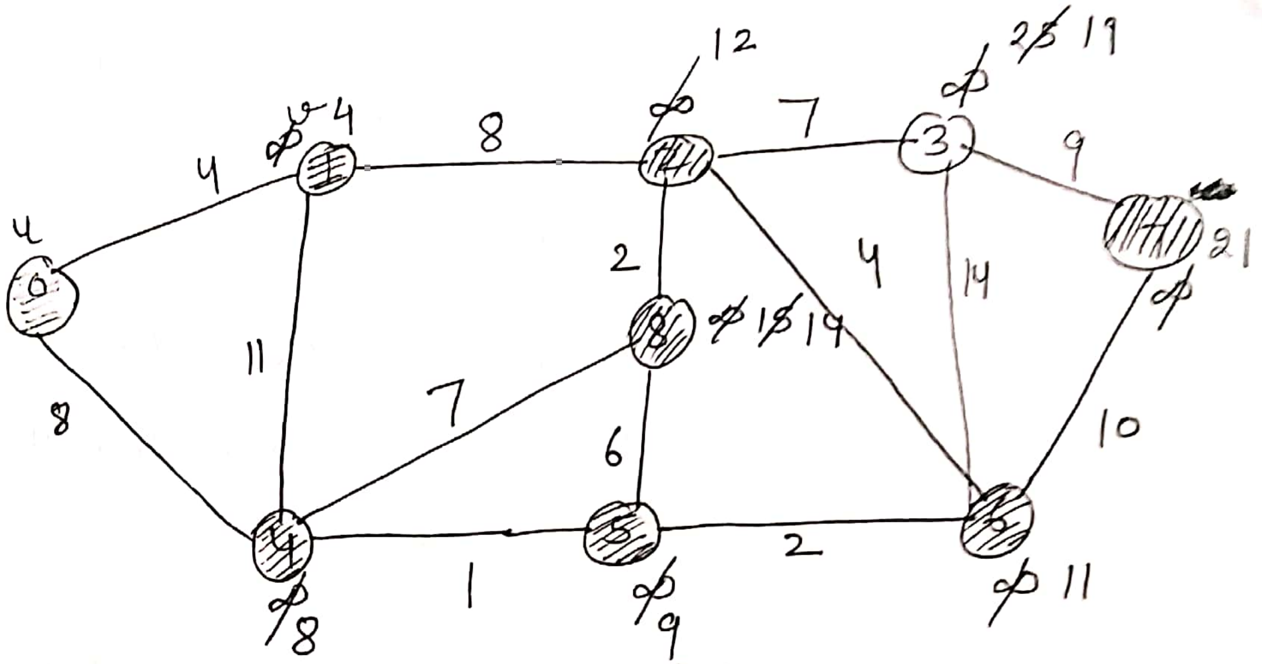
→ Now, we will compare all the vertices except the vertex 0.

→ Since the vertex 1 has the lowest value, i.e. 4 ∴, vertex 1 is selected.

Since vertex 1 is selected, so we consider the path from 1 to 2 and 1 to 4. We will not consider the path from 1 to 0 as vertex 0 is already selected.

So, like this, we will continue for the whole graph.

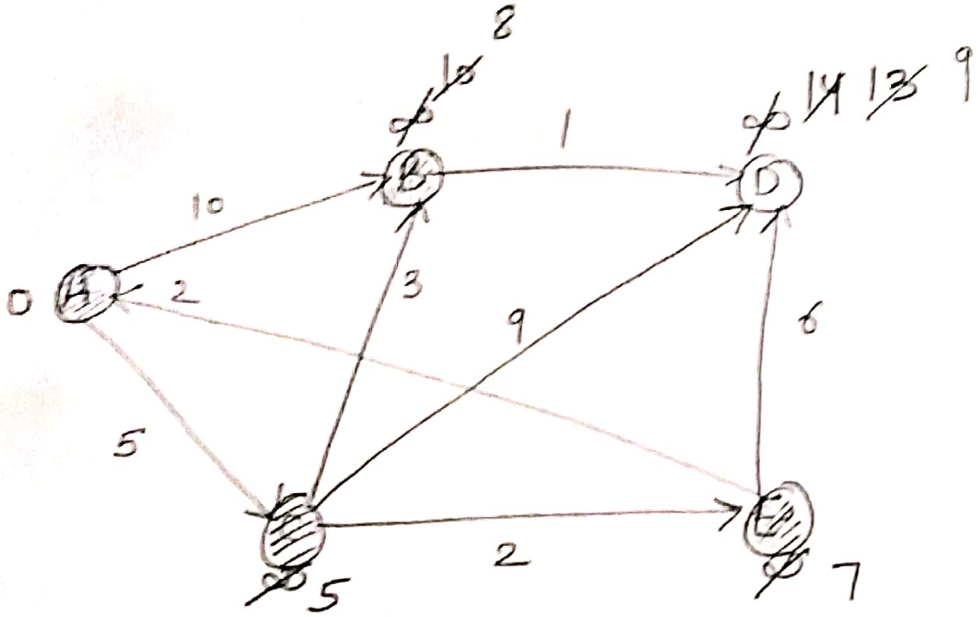
- The graph after applying dijkstra algo is shown as follows:



- Now, if we want to find out the shortest distance from 0 to 5 that is 9.

Similarly, from 0 to 7, it is 21.
Hence we can find out the shortest distance of each node from the above graph.

Now, we will solve the dijkstra algo for a directed graph as shown below.



Take source Vertex = A

	A	B	C	D	E	
A	$\boxed{0}$	∞	∞	∞	∞	
C		10	$\boxed{5}$	∞	∞	
E		8		14	$\boxed{7}$	← minimum distance vertex
B		$\boxed{8}$		13 ←		
D				$\boxed{9}$ ← pointer		

$$d(u) + c(u,v) < d(v)$$

If we want to find out the shortest Path :⇒

Suppose from A to D ⇐ we will set the pointer at 9

selected vertex is D. Move the pointer one step backward. If the value is changed, we will check in row which vertex is selected.

(A to D) Path: DBCA ⇒ ACBD ⇒ 9
5+3+1 ⇒ 9

from A to B → (8)

from the table, we can see the shortest distance of B is 8.

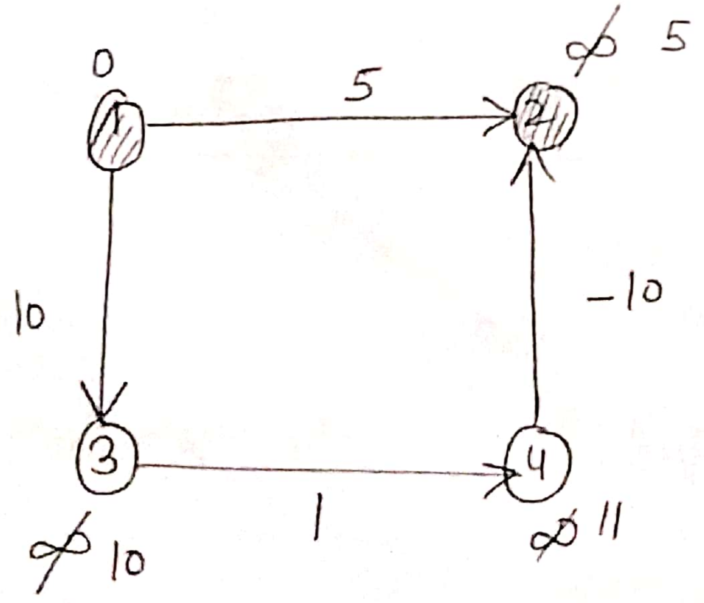
Just set the pointer. Move one step backward

$$BCA \Rightarrow \underline{ACB}\ (\text{Shortest path})$$
$$= 5+3 = 8$$

Draw back of Dijkstra Algorithm :→

It may or may not work when the weight of edges are negative.

Here is an example:



Suppose we want to update the distance from Vertex 4 (4) to Vertex 2 (v).

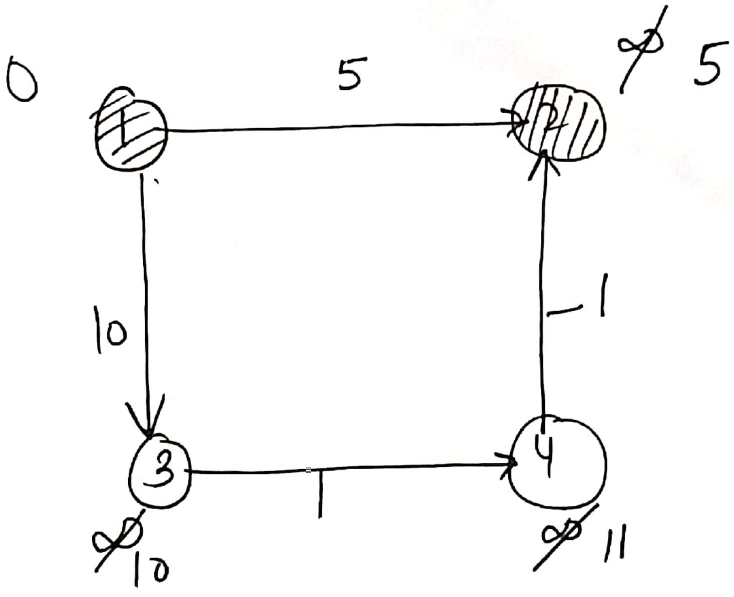
$$11 - 10 = \textcircled{1}$$

→ if we go from 1 to 3 to 4 to 2, then distance would be 1

and $1 < 5$

but we cannot update it.
because the node is already visited.
So, in this case, dijkstra algo has wrong distance.

→ This is not always be true
Suppose if we change the value as −1 from vertex 4 (4) to vertex (2) v.



$11 - 1 = 10$

and $5 < 10$
So 5 is already updated.