Single Source Shortest Path

Dijkstra's Algo – assume that all edge weights in the input graph are non-ve as in the road map example.

Bellman-Ford Algo – allow -ve weight edges in the input graph and produce a correct answer as long as no -ve weight cycles are reachable from the source.

1) **Dijkstra's Algo** – It is a greedy algo that solves the SSSP problem for directed graph $G = (V, E)$ with non-ve edge weights ie $w(u, v) \geq 0$ for each edge $(u, v) \in E$

Eg – Electric Wire fitted to cover up the room with $\min^m$ amount of wire
Google Map

Dijkstra Algo(G, w, s) S = solution

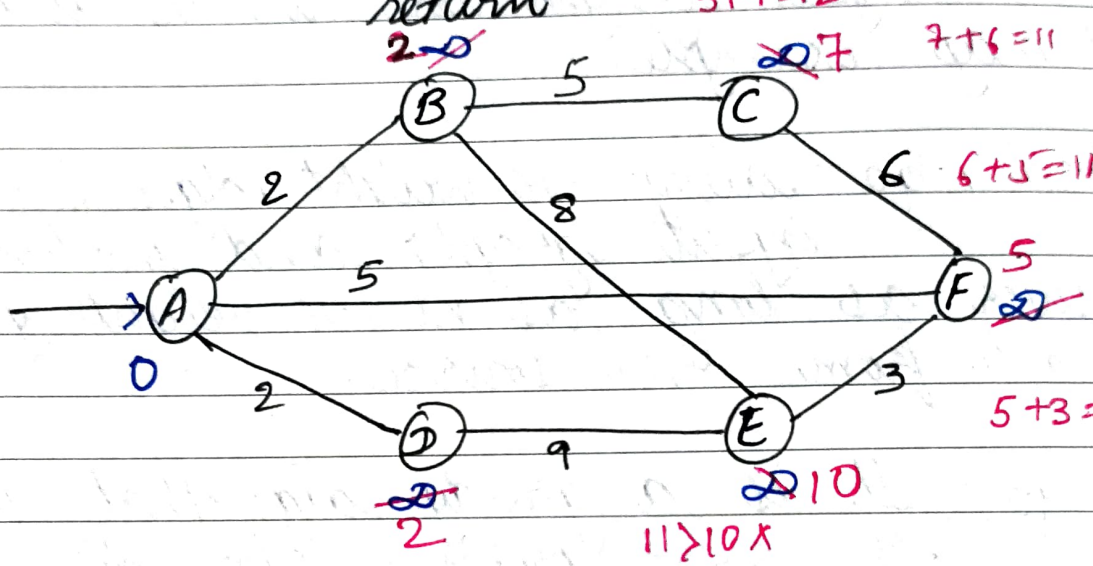
1) Initialize single source (G, s) → (1)
2) $S = \phi$ $S = 0$
3) Q = All the vertices v of G Q - Queue → (1)
4) while $Q \neq 0$ ↳ log V (heap)
   u = extract-min(Q)
   $S = S \cup \{u\}$ or $S + \{u\}$
   for each vertex $v \in G$ Adj$[u]$ → Each Vertex ke Ek baar chal raha. To Jitni Edges hai
   Relax (u, v, w) ↳ E

So $E \log V$

edge/vertex
To select har min → adjacent & wt vertex
Relax(u, v, w) → distance/weight

if (dis[v] > dis[u] + length(u,v))
    dis[v] = dis[u] + length(u,v)
    return

Eg.

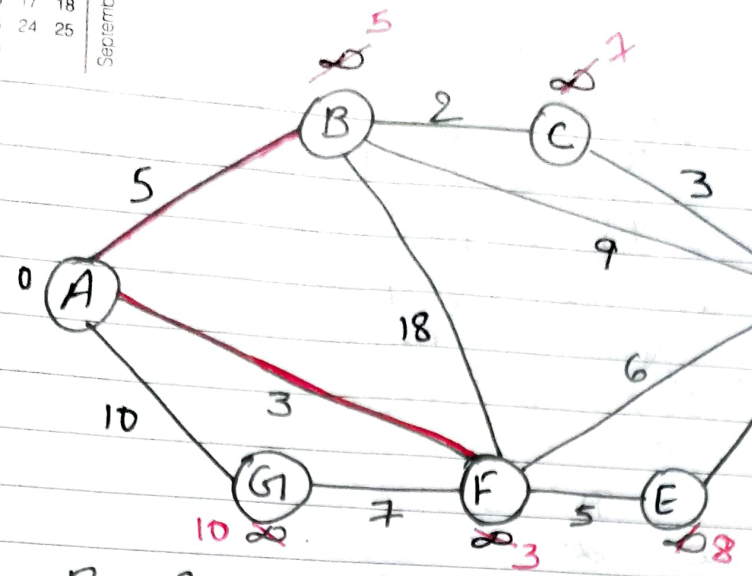


	A	B	C	D	E	F
A	0	∞	∞	∞	∞	∞
B		2	∞	2	∞	5
D			7	2	10	5
F			7		10	5
C			7		8	
E					8	

S = {A, B, D, F, C, E}

14 Sunday 0 2 2 5 7 8

Eg.



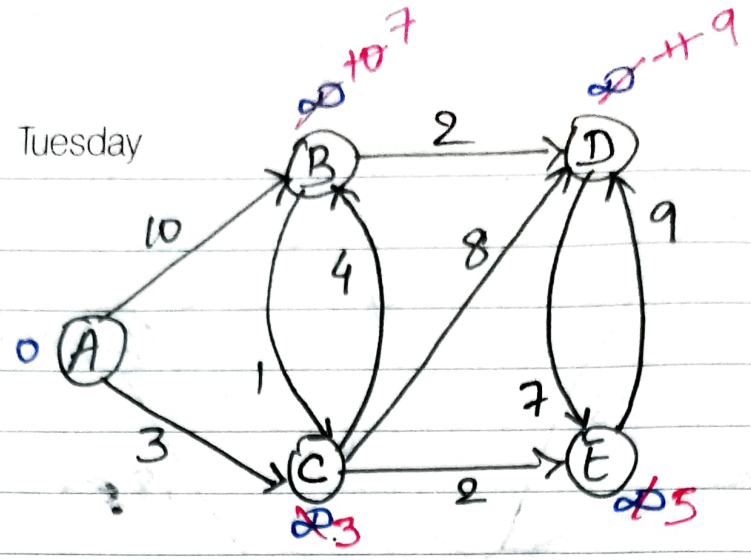
	A	B	C	D	E	F	G1	
A	0	∞	∞	∞	∞	∞	∞	
F		5	∞	∞	∞	3	10	F-G1 = 3+7 = 10 F-D = 3+6 = 8
B		5	∞	9	8		10	F-E = 3+5 = 8 F-B = 3+18 = 21
C			7	9	8		10	B-C = 5+2 = 7 B-F = 5+18 = 23
E				9	8		10	B-D = 5+9 = 14 ✗ C-B = 7+2 = 9 ✗ C-D = 7+3 = 10
D				9			10	E-D = 8+18 = 26 ✗
G1							10	E-F = 8+5 = 13 ✗
							10	D-C = 9+3 = 12 ✗ D-F = 9+6 = 15 ✗
								D-B = 9+9 = 18 ✗ D-E = 9+18 = 27 ✗

$S = \{A, F, B, C, E, D, G1\}$

0 3 5 7 8 9 10

Time Complexity – $O[E(\log V)]$

8)



	A	B	C	D	E
A	[0]	∞	∞	∞	∞
C		10	[3]	∞	∞
E		7		11	[5]
B		[7]		11	
D				[9]	

C-E = 3+2 = 5
C-D = 3+8 = 11
C-B = 3+4 = 7 < 10 update
E-D = 5+9 = 14 ✗
B-D = 7+2 = 9 < 11
B-C = 7+1 = 8 ✗

S = {A, C, E, B, D}
0 3 5 7 9