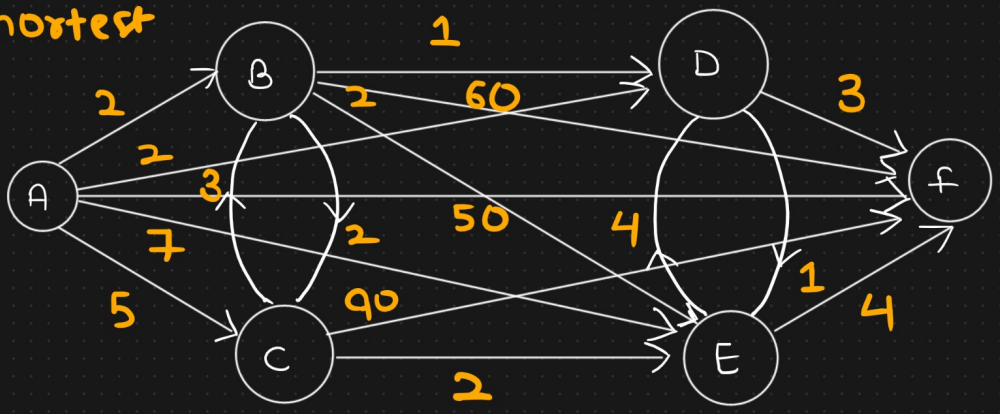
Dijkstra's Algorithm

4 single source shortest path



Source = A

Distances

Degree

A: {B:2, C:5, D:2, E:7, f:50} (5)

B: {C:2, D:1, E:2, f:60} (4)

C: {B:3, E:2, f:90} (3)

D: {E:1, f:3} (2)

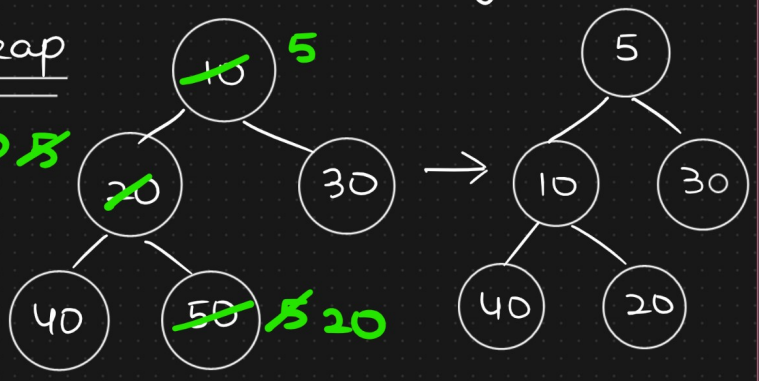
E: {D:4, f:4} (2)

f: {}

+ve edge weights

$\Theta(\log n)$

Minheap



Adjacency List

Dijkstra's Algorithm

Time complexity

Build minheap — Decrease key

$\Rightarrow V + V\log V + E\log V + 2E$

$\Rightarrow \Theta((V+E)\log V)$

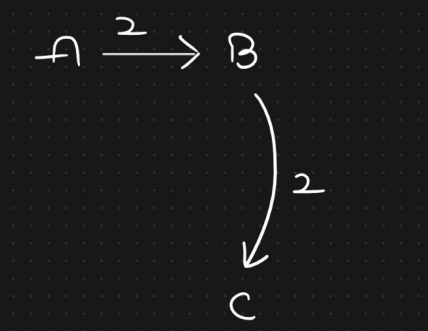
$2E$ — sum of degree

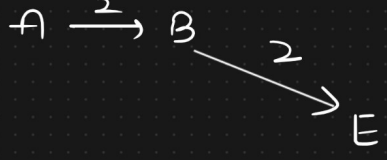
Pop (Minheap) = $\Theta(\log n)$

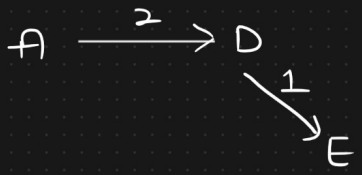
Pop elements from minheap

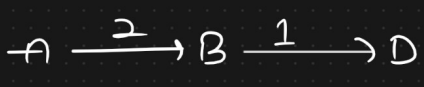
Source = A

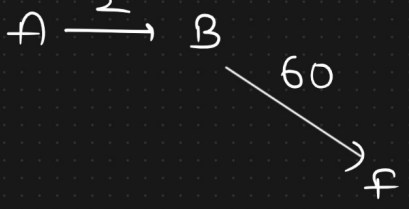
	A	B	C	D	E	F	
	0	∞	∞	∞	∞	∞	
	N	N	N	N	N	N	
A		2	5	2	7	50	$5\log V + 5$
		A	A	A	A	A	
B			4	2	4	50	$2\log V + 4$
			B	A	B	A	
D			4		3	5	$2\log V + 2$
			B		D	D	
E			4			5	$0\log V + 2$
			B			D	
C						5	$0\log V + 3$
						D	
F							

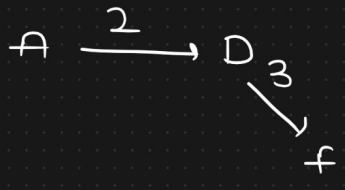


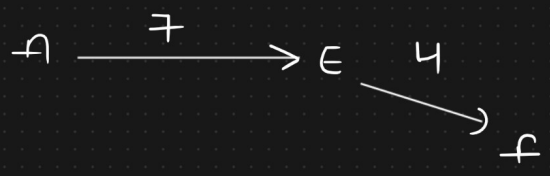


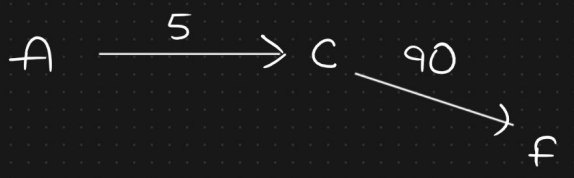












Time complexity Analysis

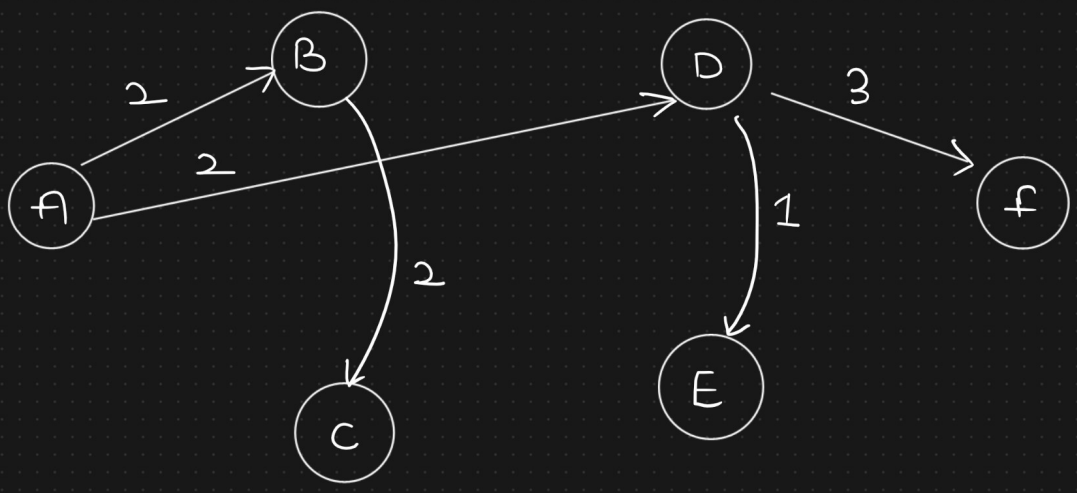
Adjacency Matrix → $V + V\log V + E\log V + V^2$ ← Degree

$\Rightarrow \Theta(V^2)$

Result

A - B = 2	A - D = 2	A - f = 5
A - C = 4	A - E = 3	A - A = 0

final output



Limitation

↳ -ve edge weight

↳ Dijkstra's Algorithm is not working well

⇓

Dynamic Programming ← Explore all the possible path

↳ More time to give result

↳ Bellman ford Algorithm