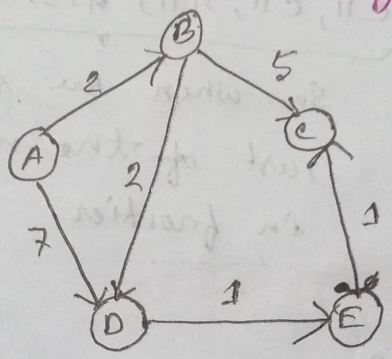
Single source shortest Path

Dijkstra's Algorithm (Greedy Algorithm)



/dike stra/

$A \to B \to C = 7$

$A \to B \to D \to E \to C = 4$

(2 + 2 + 1 + 1)

shortest in terms of weight

current node	Frontier (stores all available nodes, can be any data structure)
	S0 (started to start)
S0	P1, d3, e9
P1	d3, e9, q16
d3	b4, e5, e9, c11, q16
b4	e5, a6, e9, c11, q16
e5	a6, π7, e9, c11, h13, q16
a6	π7, e9, c11, h13, q16
π7	f8, e9, c11, h13, q16
f8	e9, g10, c11, c11, h13, q16
e9	g10, c11, c11, π11, h13, q16, h17

(new c from f to c)

from S0 we can go to d3 e9 P1

when we insert these we insert them in sorted order

P to q is start to q = 1 + 15 = 16

d3 to we can go b4 c11 e5

b4 to a6

e5 to h13, π7

e9 to h17 π11

g 10
goal node

c11, c11, r11, h13, q16, h17

So, when we stop searching rest of the nodes remains in frontier.

s
d
e
r
f
g

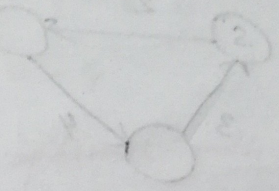
So, s d e r f g } It's the shortest path from start to goal.

Another Problem (Dijkstra's Algorithm)
We can reach any goal

current node	Frontier
	S0
S0	A1, (B1)
A1	~~B1, C4, D3, E5~~ B1, D3, C4, E5
(B1)	H2, D3, (F3) C4, E5
	If two nodes have same cost it will be FIFO
H2	D3, F3, C4, E5, T6
D3	F3, C4, E5, T6, N8
(F3)	C4, (G4) E5, T6, K6, N8
C4	G4, E5, M5, T6, K6, N8

G4 E5, M5, T6, K6, N8

Goal node
⇓
we can go upto K6



So,

S
↓
B
↓
F
↓
G

MID TERM ON NOVEMBER 10

Dijkstras doesn't gurantee to provide the shortest path. Negative Index.