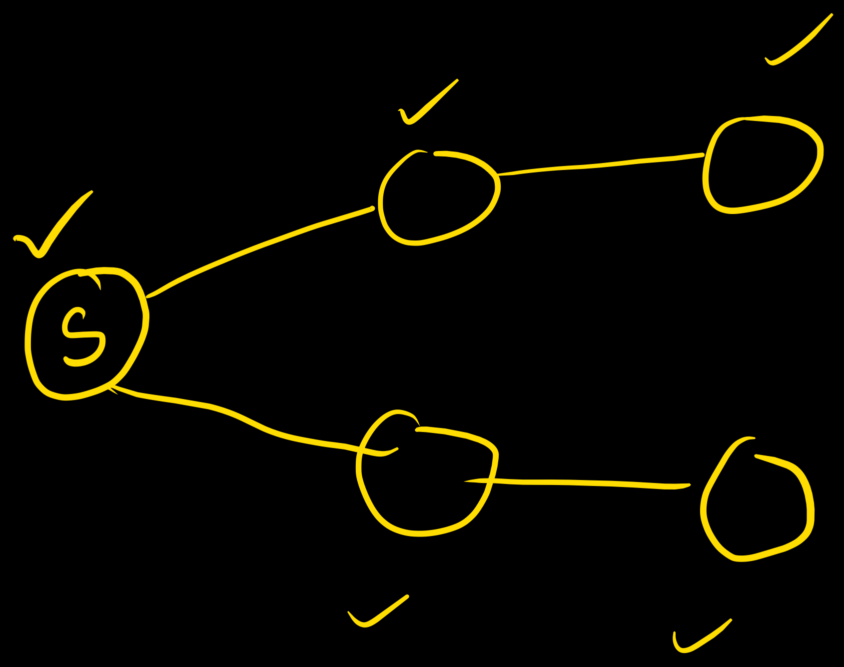
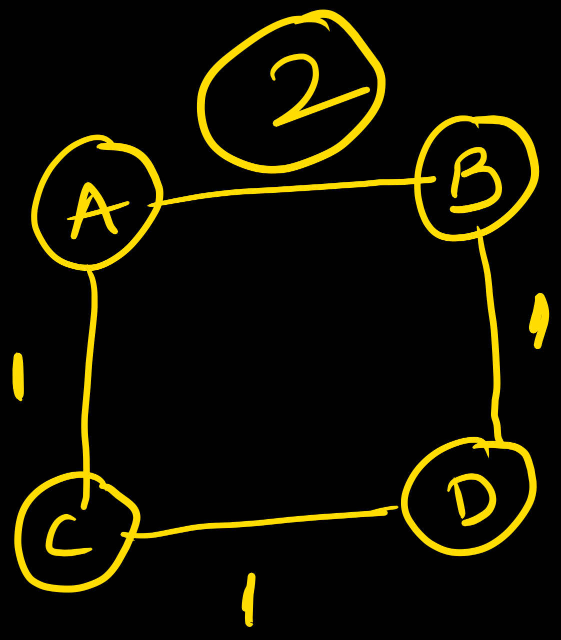
MST

Single · source shortest path

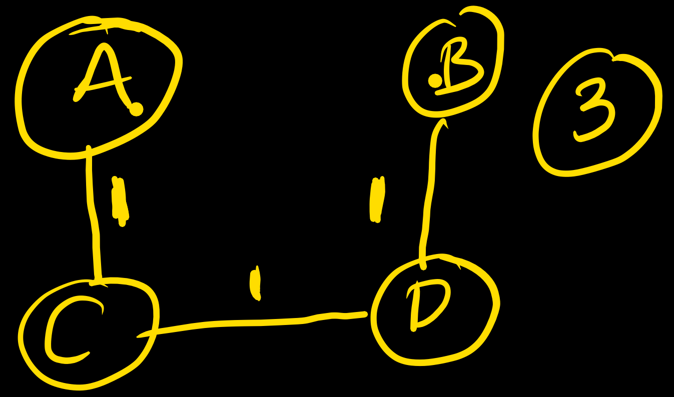




MST

$\Downarrow$

Shortest path



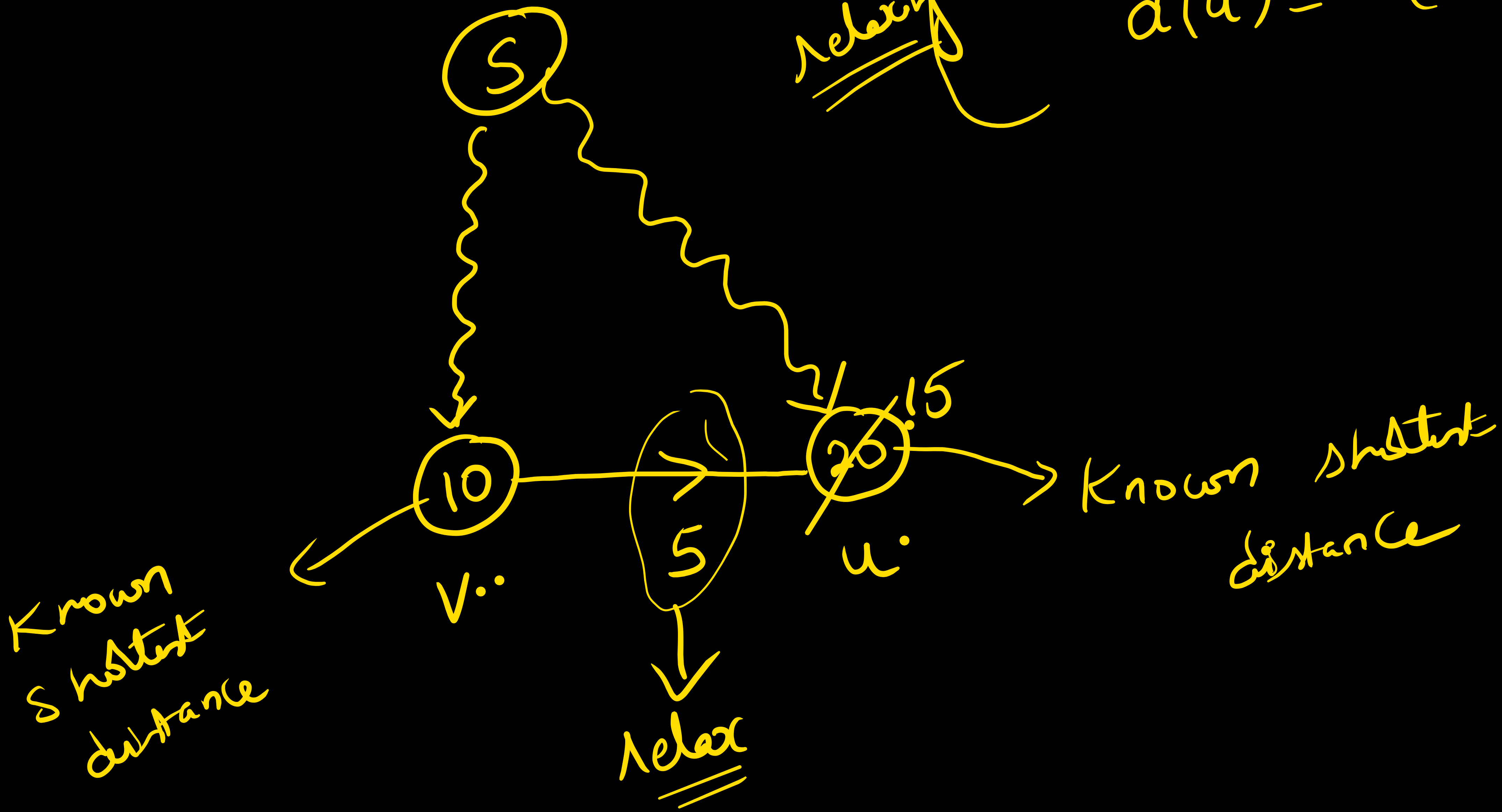
Dijkstra algo:

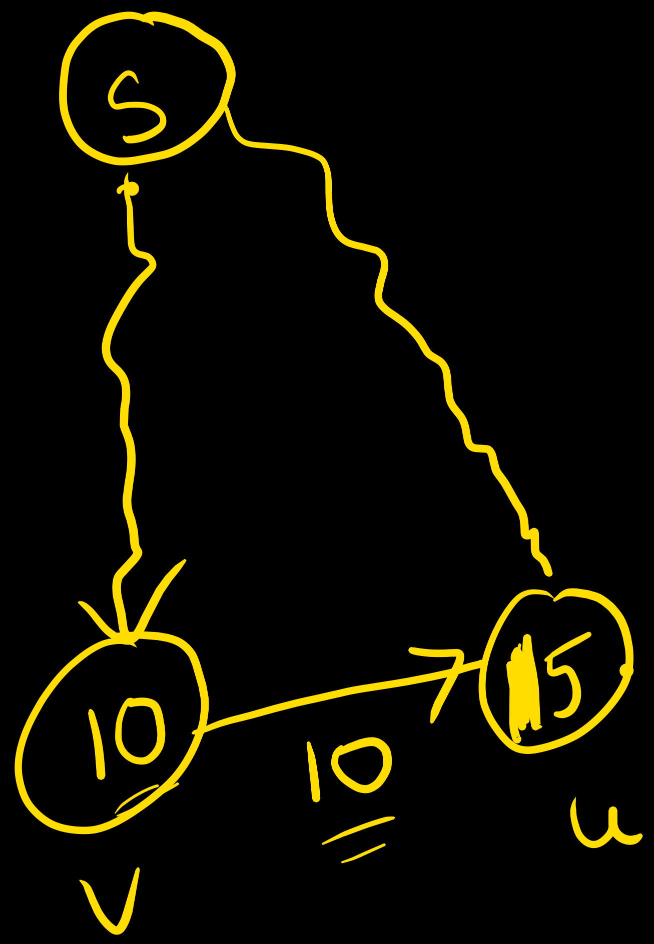
relaxing

relaxing:

$$\text{if } \big(\underbrace{d(u)}_{20} > \underbrace{d(v)}_{10} + \underbrace{c(u,v)}_{5}\big)$$

$$d(u) = d(v) + c(u,v) \checkmark$$





relaxing

useful ✓

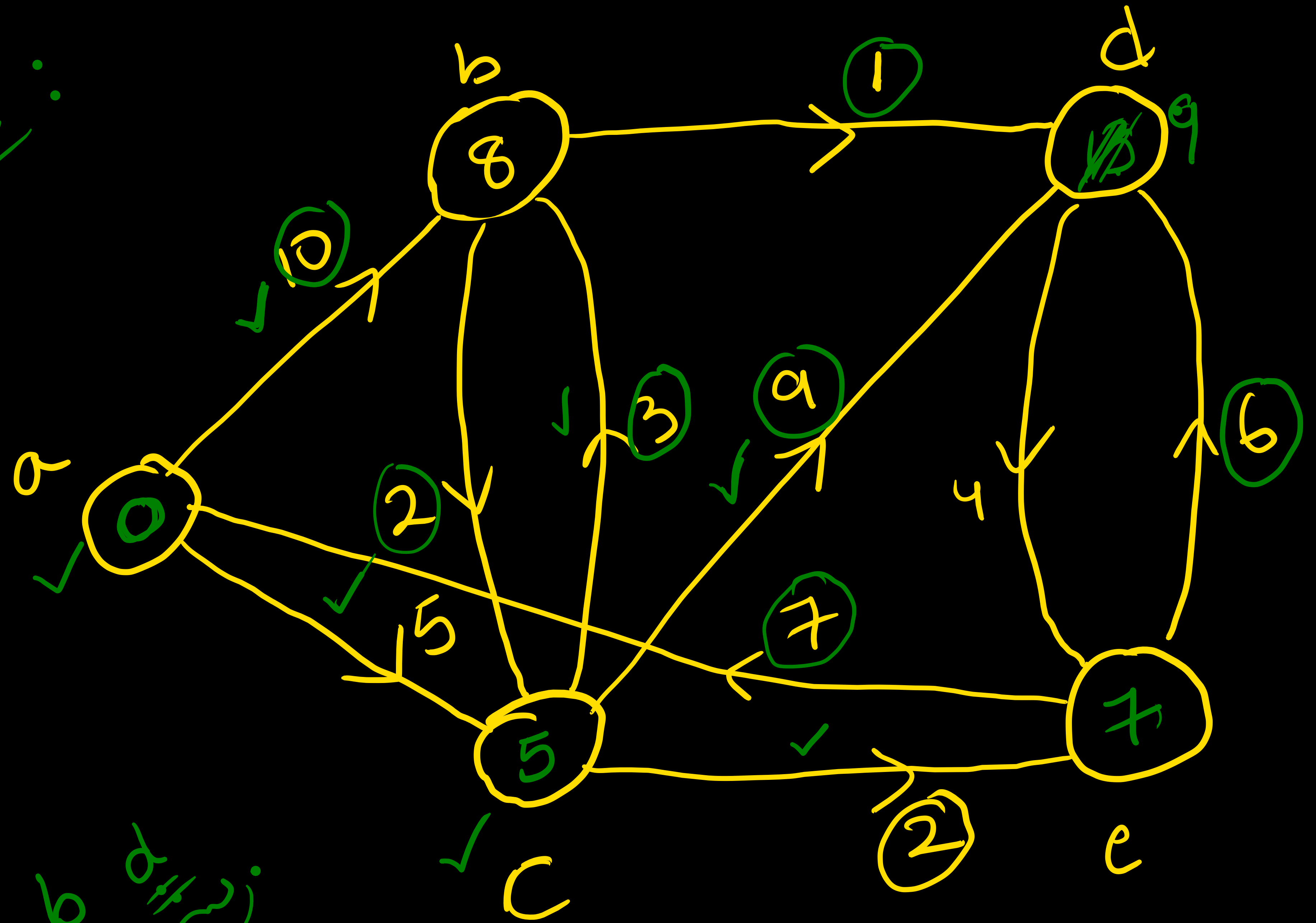
useless. ✓

if I relax (v, u) ⟹ better?

no.
we cannot get better
distance by relaxing.

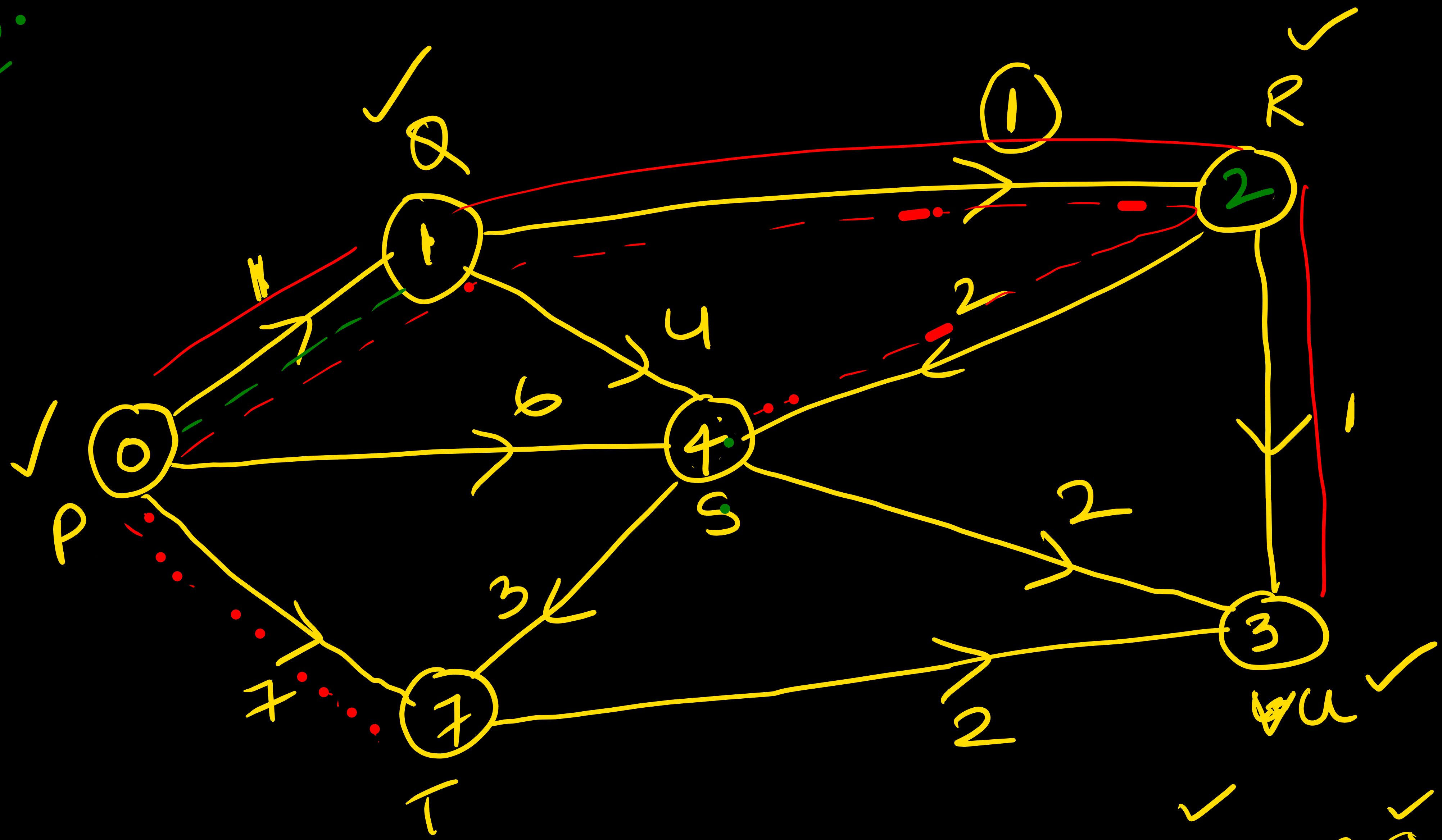
Gate :

a- source



a c e b d

Gate:



$S \xrightarrow{3} \cancel{T}$

$S \xrightarrow{2} UX$

P Q R U S T

Dijkstra

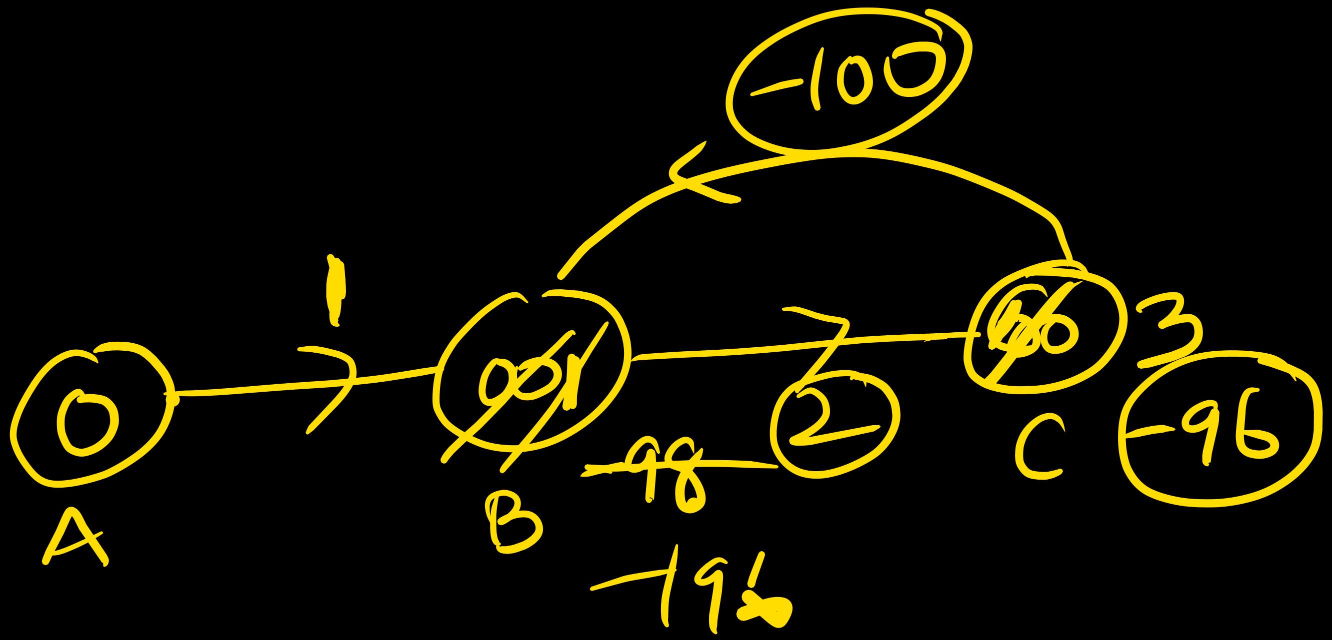
⇓

only for positive.

⇓

-ve edge get an answer provided that there is no negative weight cycle.

Example where dijkstra will fail:



Solution is Bellman Ford.

→ Find negative weight cycles.

Negative weight cycles - Dijkstra will fail.

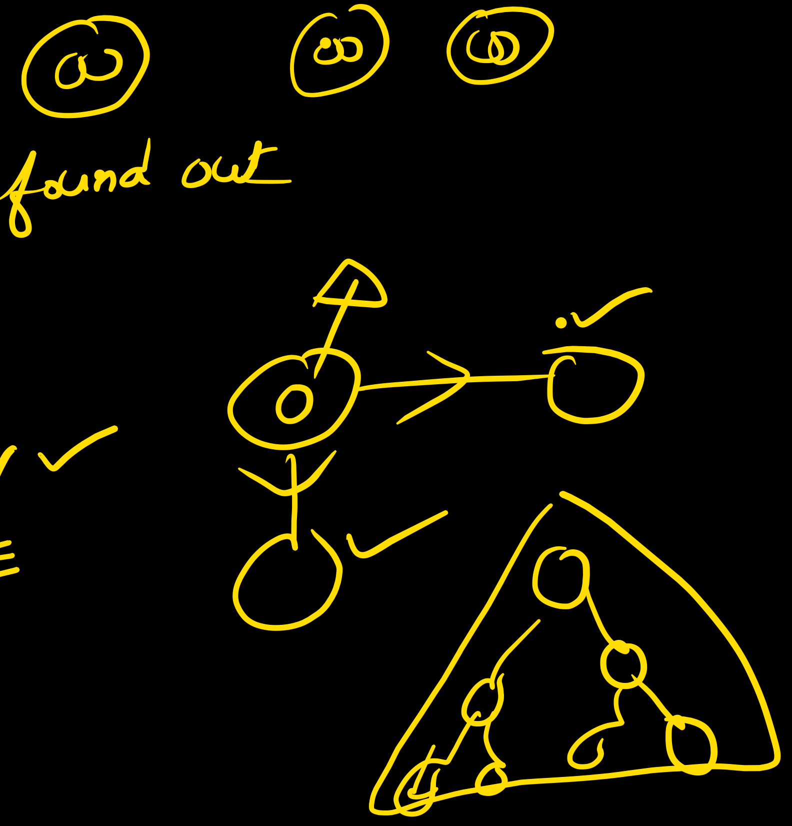
So Dijkstra will not allow -ve edges as it cannot find out -ve weight cycles.

Dijkstra (G, w, S)

1. initialize - single - source $(G, S) \rightarrow O(V)$

2. $S = \phi$ → nodes whose shortest path is found out

3. $Q = G.V$ → build a min heap.



while $(Q \neq \phi)$

$u = \text{extract-min}(Q) \Rightarrow O(\log V) \times V$

$S = S \cup \{u\}$

for each vertex $v \in G.Adj[u]$

Relax $(u, v, w) \Rightarrow$ decrease key

$E \times (\log V) \Rightarrow O(E \log V)$

$O(V \log V) + O(E \log V)$

$O(E \log V)$.

Bellman Ford

- NO (~~∞~~ −∞ cycle)
- Yes (NO −ve weight cycle).

Dijkstra

- NO X −ve cycle
- Yes (SP)

Cannot detect −ve weight cycle

n ~~ds~~ nodes ⇒ ~~shortest~~ shortest path → × at most (n-1) edges.

n → vertices ✓

I pass → Shortest paths containing at most 1 edge.

II pass → " " " × at most 2 edges

III pass → " " " × 3 edges

IV " " " ~~4~~ 4 ed

⋮

n-1 pass →

at most n-1 ✓

V

I pass → shortest path containing at most 1 edge.

II 〃 2 edge

⋮

(V-1) pass 〃 (V-1) edge.

one more pass → detect ~~no~~ negative weight cycles.

(V-1) ⇒ passes ⇒ all shortest paths are found.

⇓

1 more pass → if the distance decrease ⇒ -ve weight cycle.

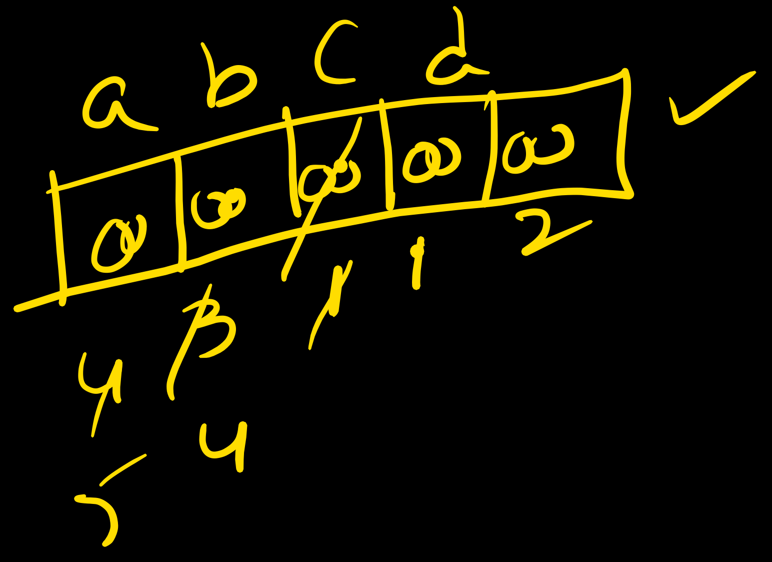
Algo: I pass ⟹ relax all edges once

II pass ⟹ " " "

III pass ⟹ " " "

⋮

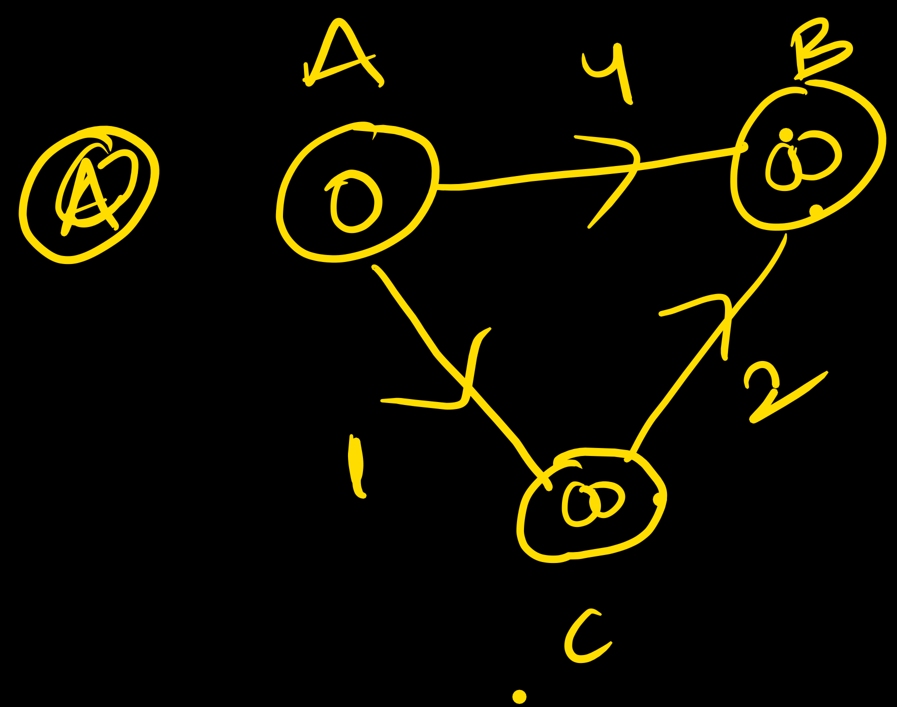
V−1 pass ⟹ " "



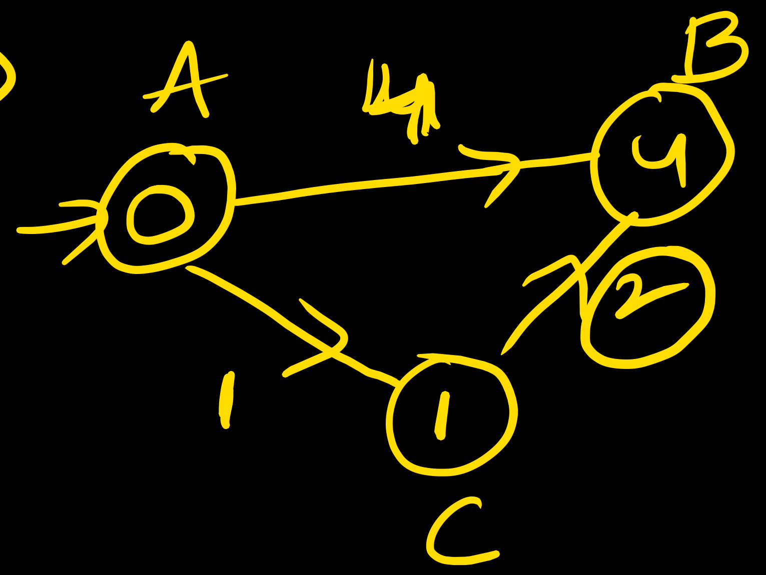
(V−1) E relaxing ⟹ in any order ⟹ no need of minheap

⟹ an array is sufficient

O(1)   O(V E) ✓



I pass



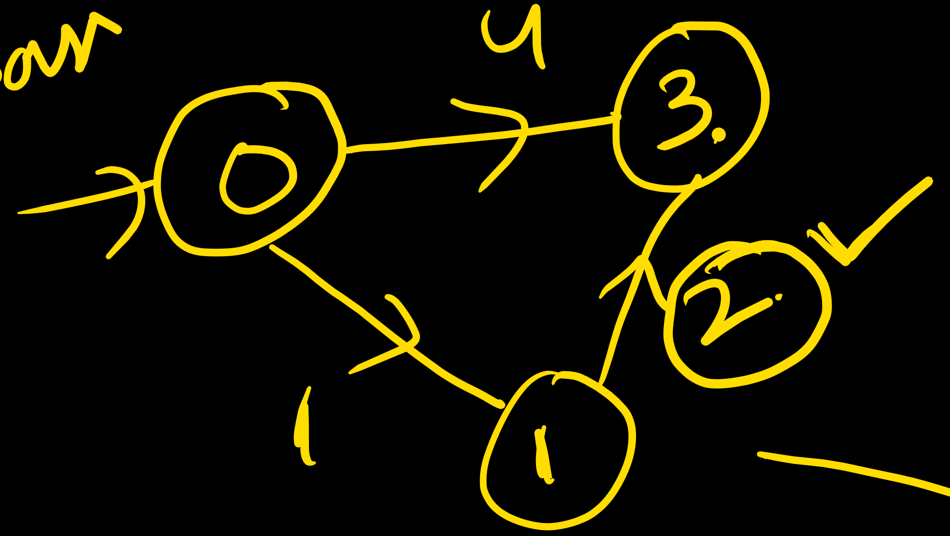
3-nodes
↓
2 pass

1 edge.

2 pass ⇒ shortest path

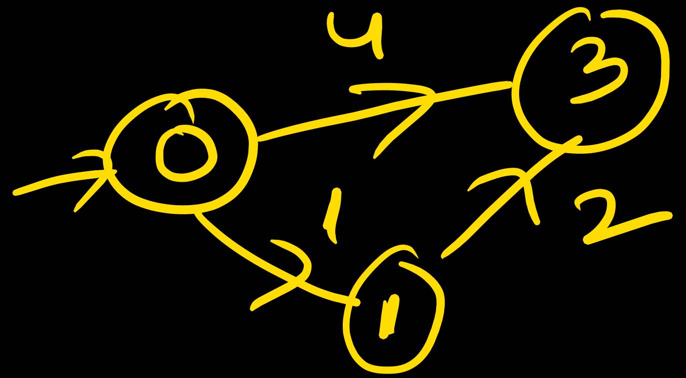
1 pass ⇒ -ve wt cycle.

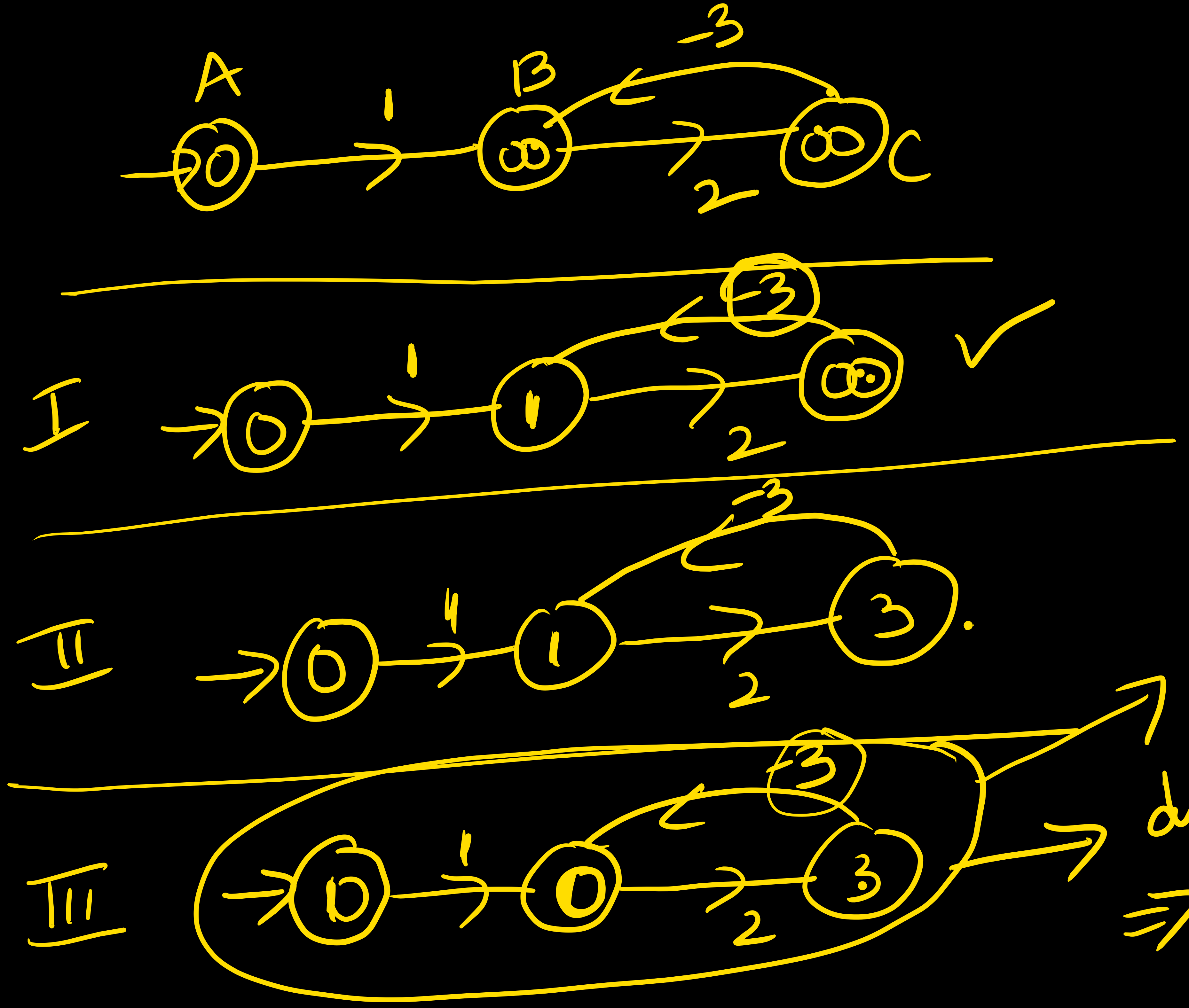
II pass



n - vertices
⇒ (n-1) passes
+ 1

III





3 nod 2 pass

-1
-3
-5
-7
-8

distance is changing.

⇒ -ve weight cycle

Bellman Ford (G, ω, S)

{ G → graph, ω → matric , S → Source

1	2	3	4	5	6
0	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~

Initialize single Source(G, S)

for (i = 1 to (V − 1)) → (V − 1)

For each edge (u, v) ∈ E → (E)

Relax (u, v, ω) → $O((V-1)(E)) = O(VE)$

For each edge (u, v) ∈ E

if (v.d > u.d + ω (u, v) } ⇒ checking if the distance is changing

return false

return true.

−ve weights are allowed in bellman ford

→ it will report if there is a −ve weight cycle.