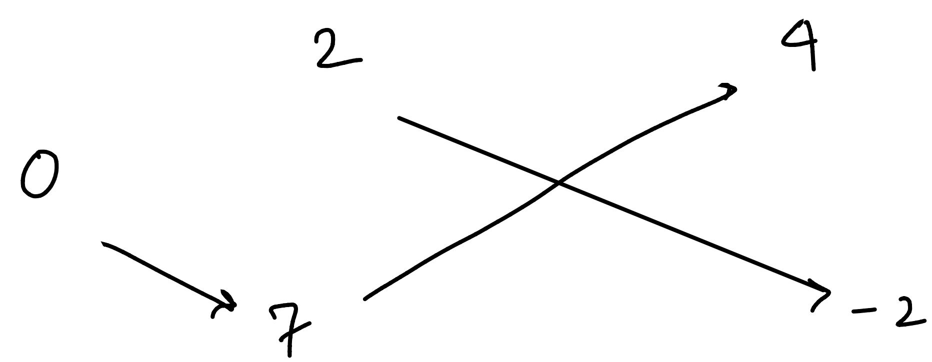
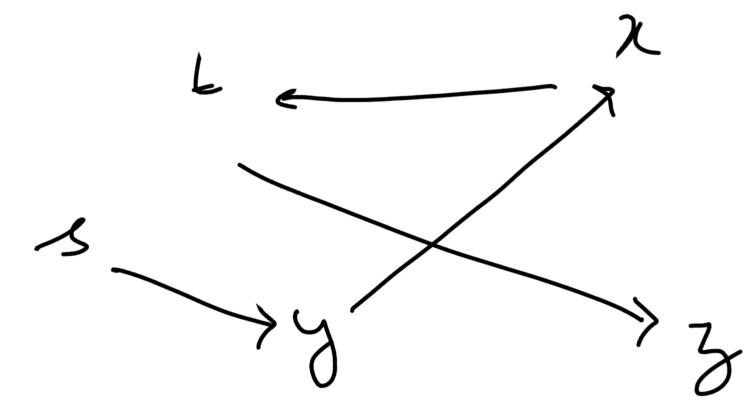
Step3





Shortest Path Tree

Time : $VE \rightarrow V^2$

Proof:

- $p = \langle \underset{s}{\underset{\shortparallel}{v_0}}, \ldots, \underset{v}{\underset{\shortparallel}{v_k}} \rangle$

- $k \leq |V| - 1$

- each iteration we relax every edge

- so in the i^{th} iteration we will relax (v_{i-1}, v_i)

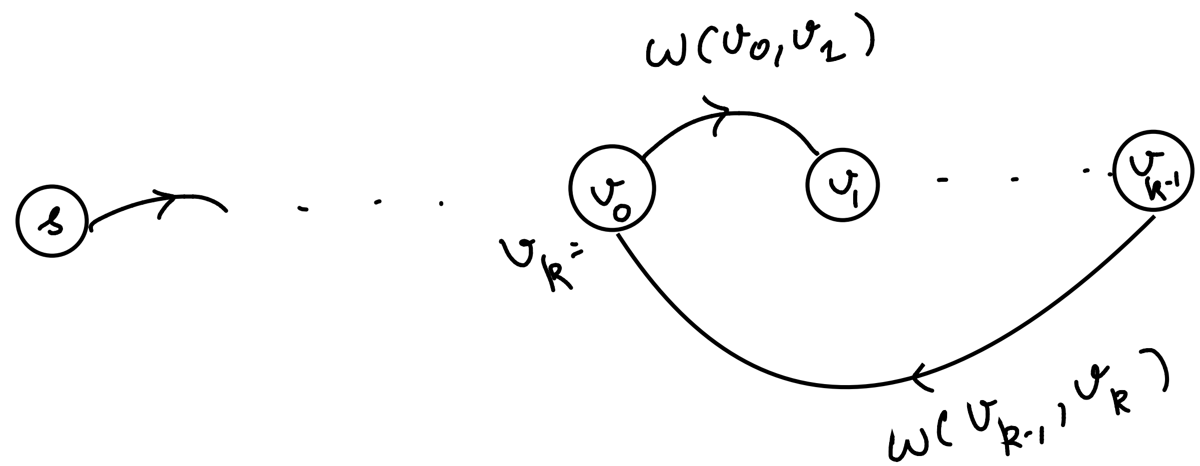
$$d[v_i] = \delta(s, v_i)$$

$\Rightarrow$ after k iterations $d[v_k] = \delta(s, v)$

- If v is not reachable $d[v] = \infty$ (value at initialisation)

- Say there is a negative cycle $c : \langle v_0, \ldots, v_k \rangle$, $v_k = v_0$

$$\Rightarrow \sum_{i=1}^{k} w(v_{i-1}, v_i) < 0$$



To contradict, say the algorithm does not detect the negative cycle

$$d[v_1] \leq d[v_0] + w(v_0, v_1)$$

$+$

$$d[v_2] \leq d[v_1] + w(v_1, v_2)$$

$$\vdots$$

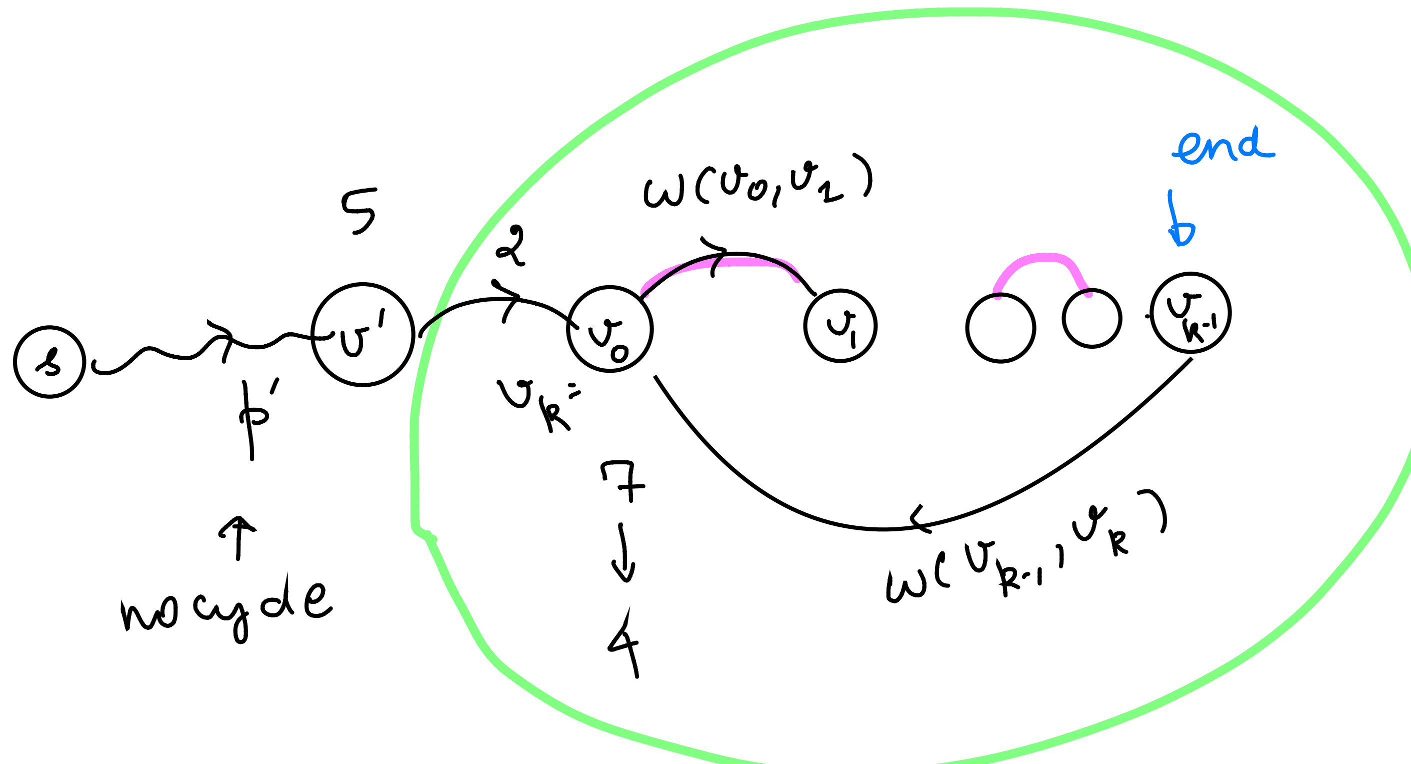
$+$

$$d[v_0] \leq d[v_{k-1}] + w(v_{k-1}, v_k)$$

(no improvement for vertices that are in the cycle)

$$\cancel{d[v_0] + d[v_1] + \dots + d[v_{k-1}]} \leq \cancel{d[v_0] + d[v_1] + \dots + d[v_{k-1}]} + \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

$$\sum_{i=1}^{k} w(v_{i-1}, v_i) \geq 0$$



$\delta(s, v_0) = -\infty$

negative cycle weight $= -3$

at some step $< |V| - 1$, $d[v'] = \delta(s, v')$

step+1 $\leq |V| - 1$ $\quad d[v] = \delta(s, v') + w(v', v)$

step' we reach v_{k-1}

First time it is reached : say $d[v_0] = 7$, and -ve cycle wt $= -3$

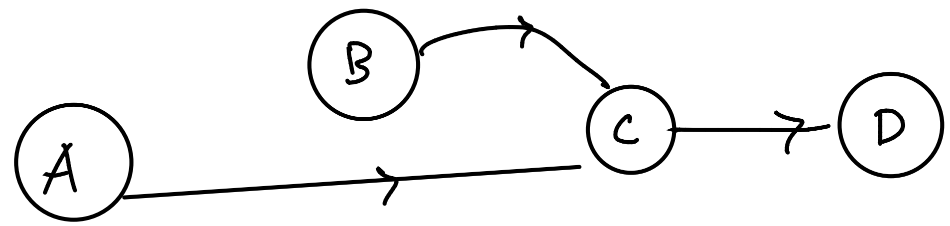
Second time reached : $d[v_0] = 4$

and second time reached : every vertex in the cycle improves.

TOPOLOGICAL SORT:

Directed Acyclic Graphs : Dependencies, Order of which should come before.

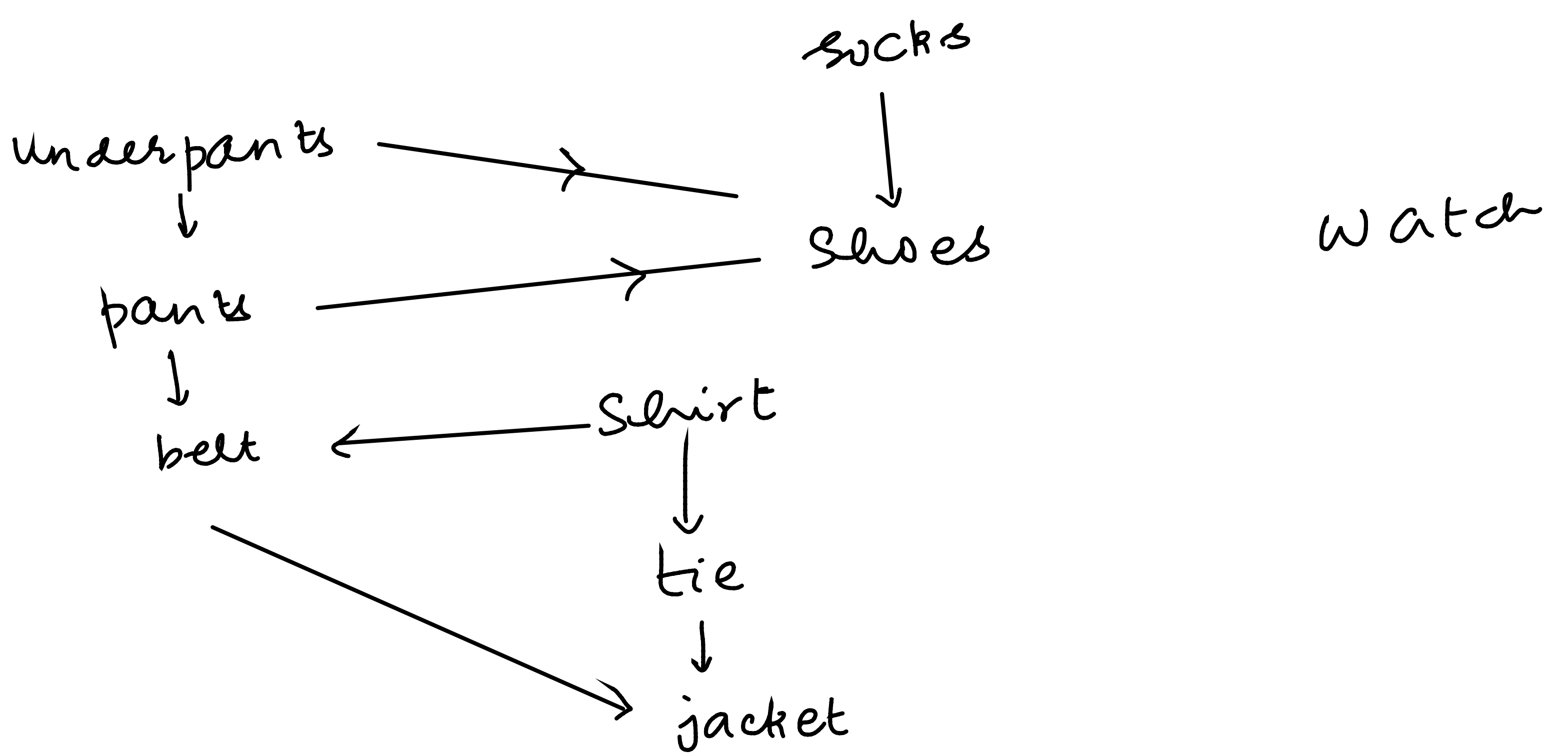
eg: curriculum of subjects



Need to know A, B to learn C.

Topics in a course:

wearing cloths (CLRS example)



Topological sort is an linear ordering of vertices of a DAG such that all edges are only from left to right

Underpants → pants → belt watch shirt → tie → jacket socks → shoes

Exercise: Complete the topological sort for below

shirt → tie underpants → pants → belt → jacket

DFS(G)

for each vertex $u \in V[G]$ (marking everything unexplored)

do colour[u] $\leftarrow$ WHITE

π[u] $\leftarrow$ NIL

time $\leftarrow$ 0

order is arbitrary

for each vertex $u \in V[G]$ (loop 1)

do if colour[u] = WHITE

then DFS-VISIT(u)

DFS- VISIT (u) (Discovery phase)

colour [u] ← Gray

time ← time + 1 (discovery time)

d [u] ← time

for each v ∈ Adj [u] } exploration of u

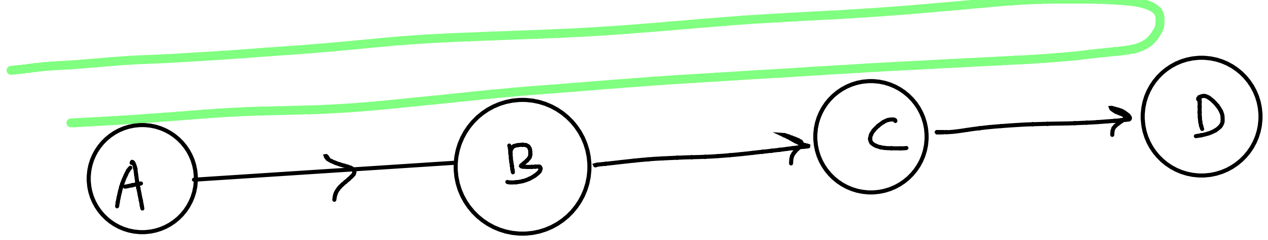
then π [v] ← u

DFS - VISIT (v)

colour [u] ← black

time ← time + 1

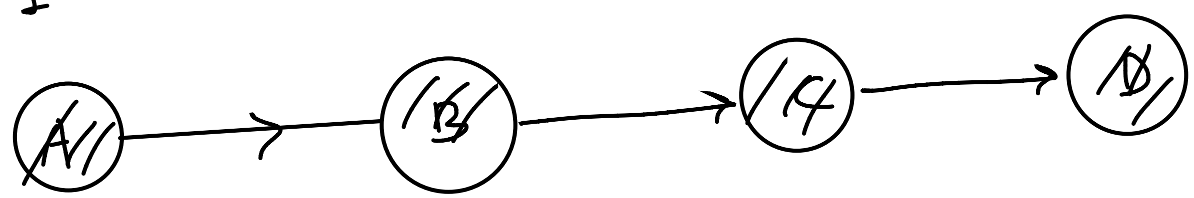
f [u] ← time (finishing time)



Vertex set order is A, B, C, D.

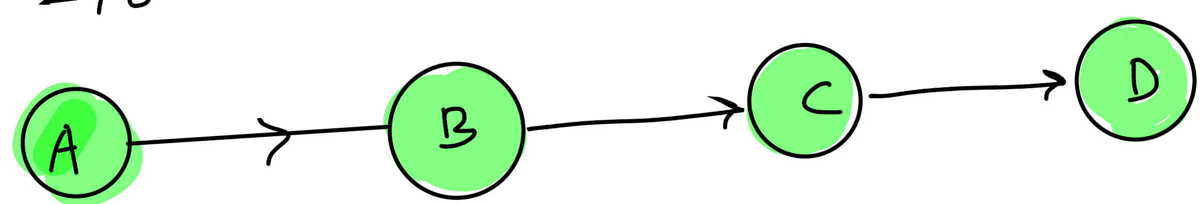
(loop 1)

d / f : 1 2 3 4



colour	Gray	Gray	Gray	Gray
time	1	2	3	4

d/f : 1/8 2/7 3/6 4/5



colour	Black	Black	Black	Black
time	8	7	6	5

LIST :

A	B	C	D
8	7	6	5

$A \rightarrow B \rightarrow C \rightarrow D$

Vertex set order is D, C, B, A

(loop 1)

		A	B	C	D
d/f	:	7/8	5/6	3/4	1/2
colour		Gray	Gray	Gray	Gray
time		7	5	3	1

$\cancel{A} \rightarrow \cancel{B} \rightarrow \cancel{C} \rightarrow \cancel{D}$

		A	B	C	D
d/f	:	7/8	5/6	3/4	1/2
colour		Black	Black	Black	Black
time		8	6	4	2

$A \rightarrow B \rightarrow C \rightarrow D$

LIST : A B C D
8 6 4 2

As and when a node finishes I will add to the front of a list