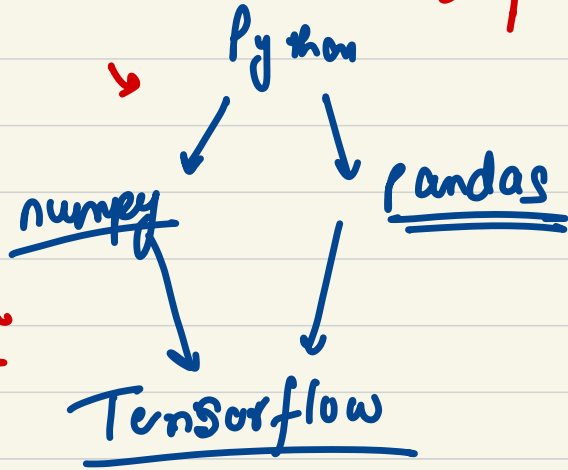
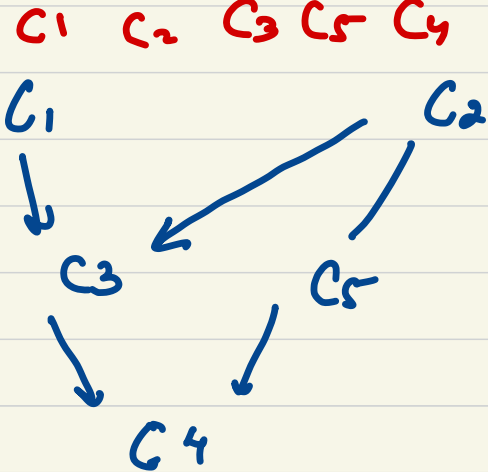




python,  
numpy,  
pandas,  
tensorflow



Dependency graph → Directed

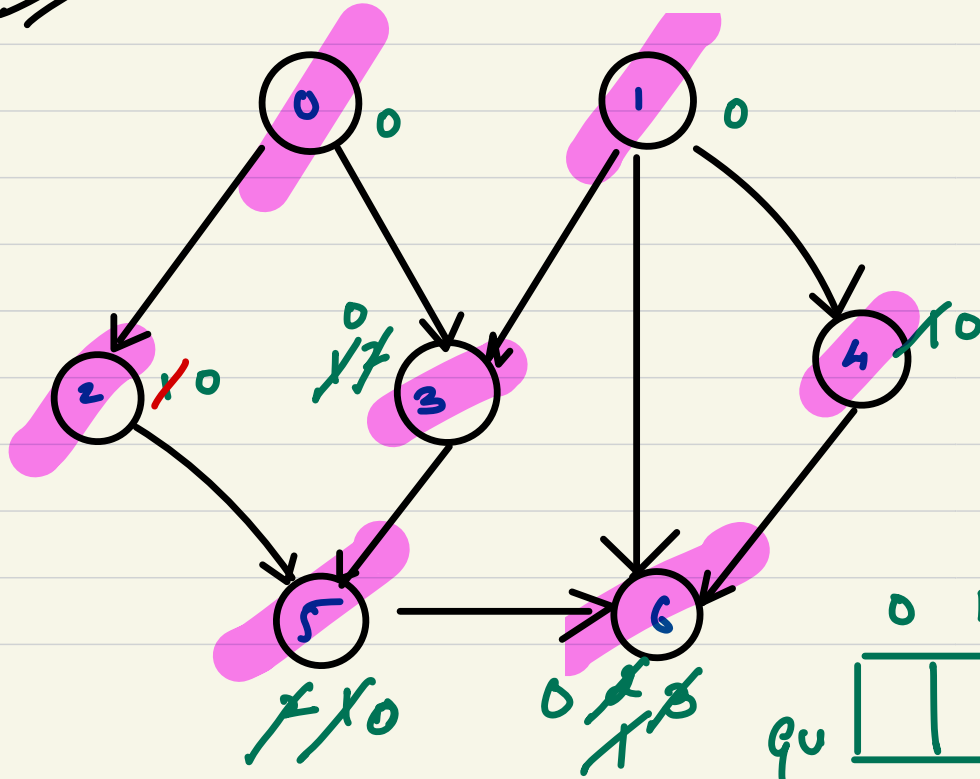


we have a sense of dependency that one item  
need to be resolved before the other  
indegree → no. of incoming edges.

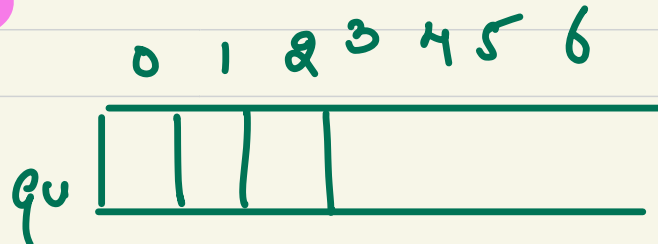
(Kahn's algo)  $\rightarrow$  bfs

Topological Sorting

(directed graph)

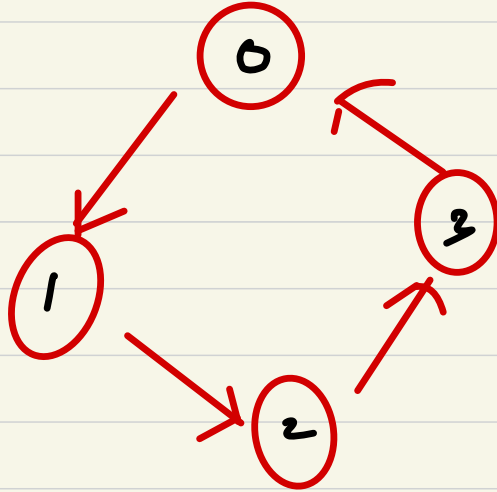


we should resolve those nodes whose indegree is 0. Add them to the bfs queue.



we will not be able to apply topological sort if there is cycle in the directed graph.

Dead lock

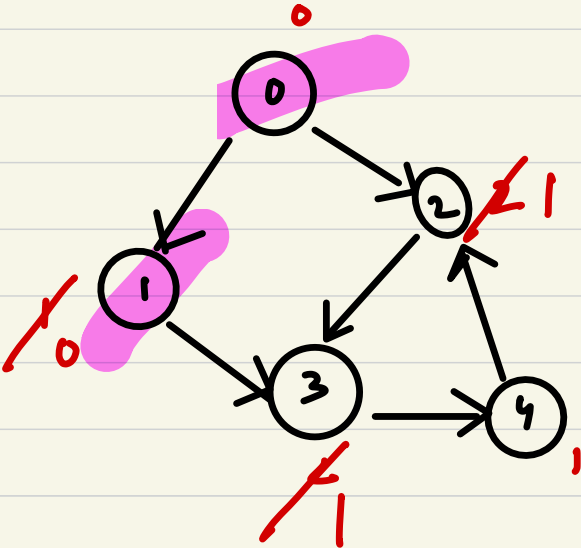


topo sort can be only applied in

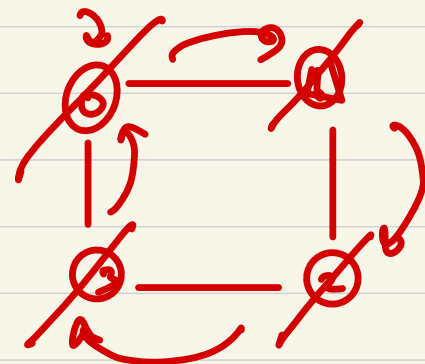
DAG.

To detect a cycle in your graph using topological sort follow these steps →

- 1) Start applying topo sort
- 2) If the no. of nodes resolved by topo sort is less than the total no. of nodes in the graph, then there is cycle.



dfs / bfs



$$\rightarrow \underline{a < b} \quad \underline{b > d} \quad \underline{e < f} \quad \underline{a > c}$$

Can we allocate any set of int values to these inequalities  
 Such that all of them can be resolved:  $\rightarrow$  TopoSort

$$\underline{a < b} \quad \underline{b < c}$$

$$\begin{aligned} a &\rightarrow 10 \\ b &\rightarrow 20 \\ c &\rightarrow 30 \end{aligned}$$

True

$$(a < b \quad b < c \quad c < a)$$

<sup>d</sup>  
false

Shortest path algorithm

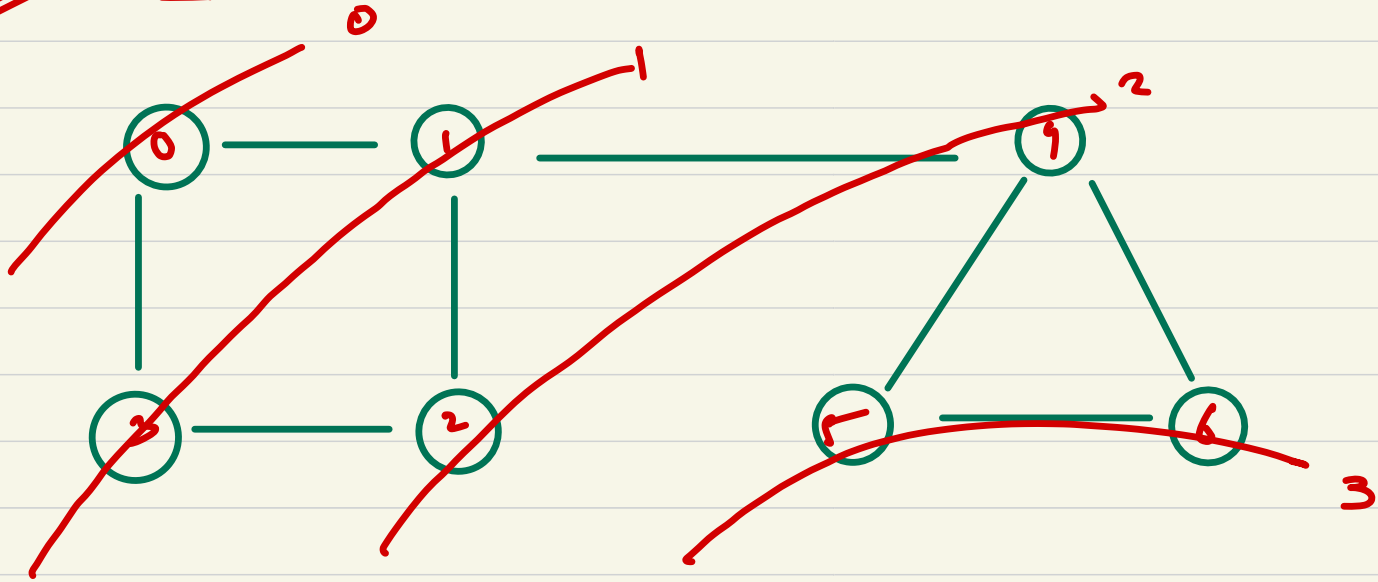
1 1  
Dijkstra's



↓  
Single src shortest  
paths

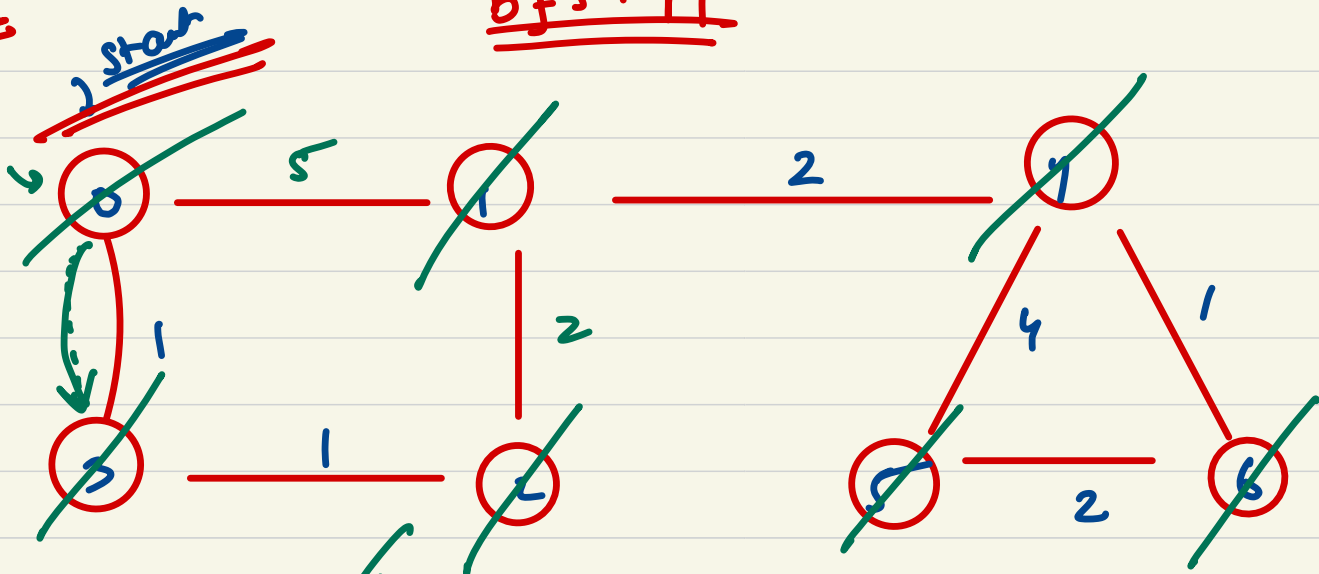


unvisited → bfs



weighted

bfs + pq



mapping

vertex	wt	vis
0	0	2
1	5	3
2	2	1
3	1	
4	6	
5	9	
6	7	



min heap  
pq → wt

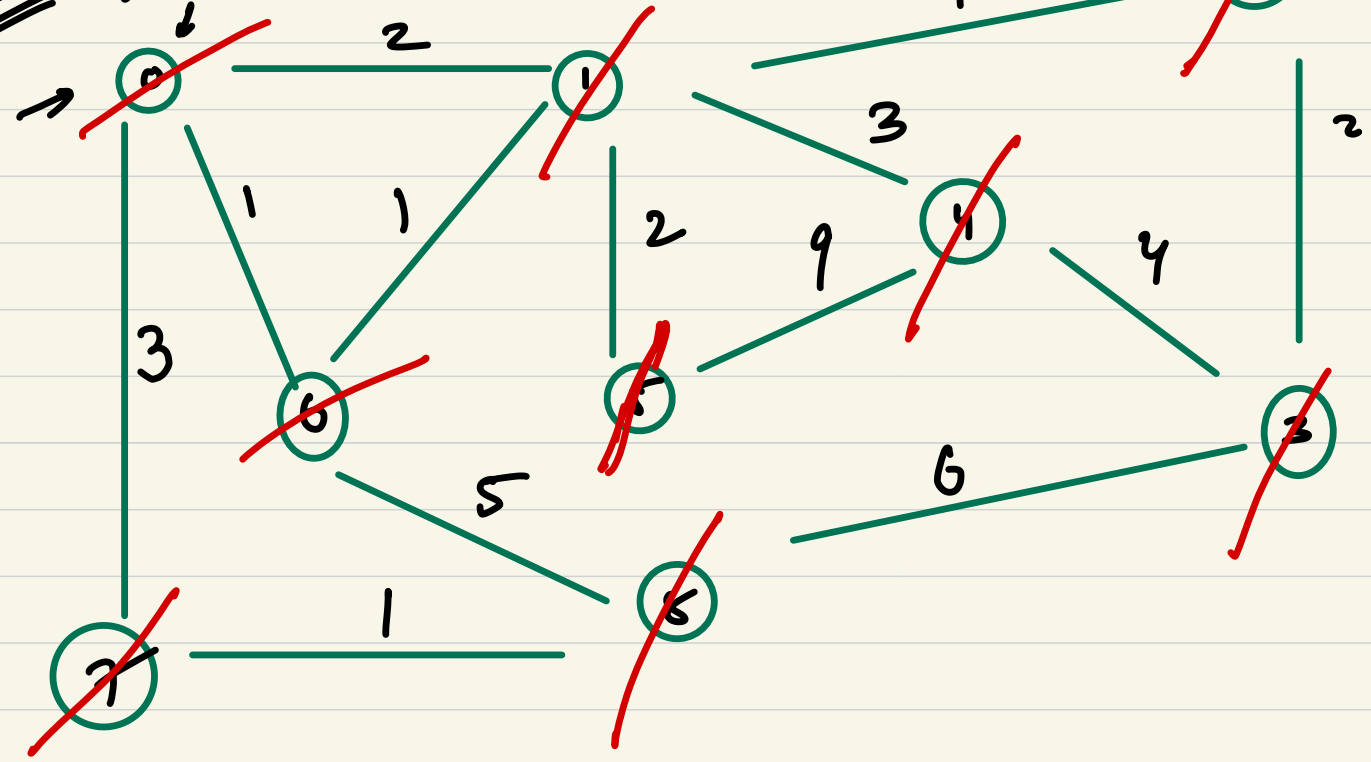
(6, 7)

(5, 9)

(5, 10)

apply  
dijsktra

source



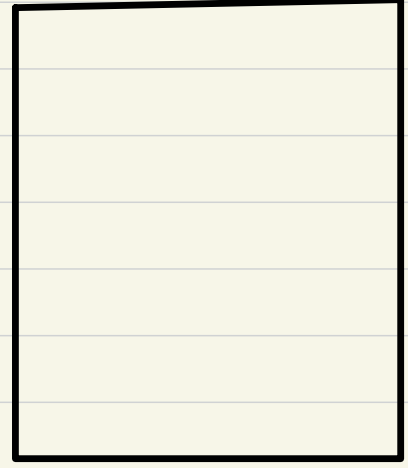
wt.  
mapping

0	—	0
1	—	<del>2</del>
2	—	<del>3</del>
3	—	<del>5</del>
4	—	<del>5</del>
5	—	<del>4</del>
6	—	<del>1</del>
7	—	<del>3</del>
8	—	<del>4</del>

via  
mapping

0	—	-1
1	—	0
2	—	1
3	—	2
4	—	1
5	—	1
6	—	0
7	—	0
8	—	<del>6</del> 7

pq → min heap → wt



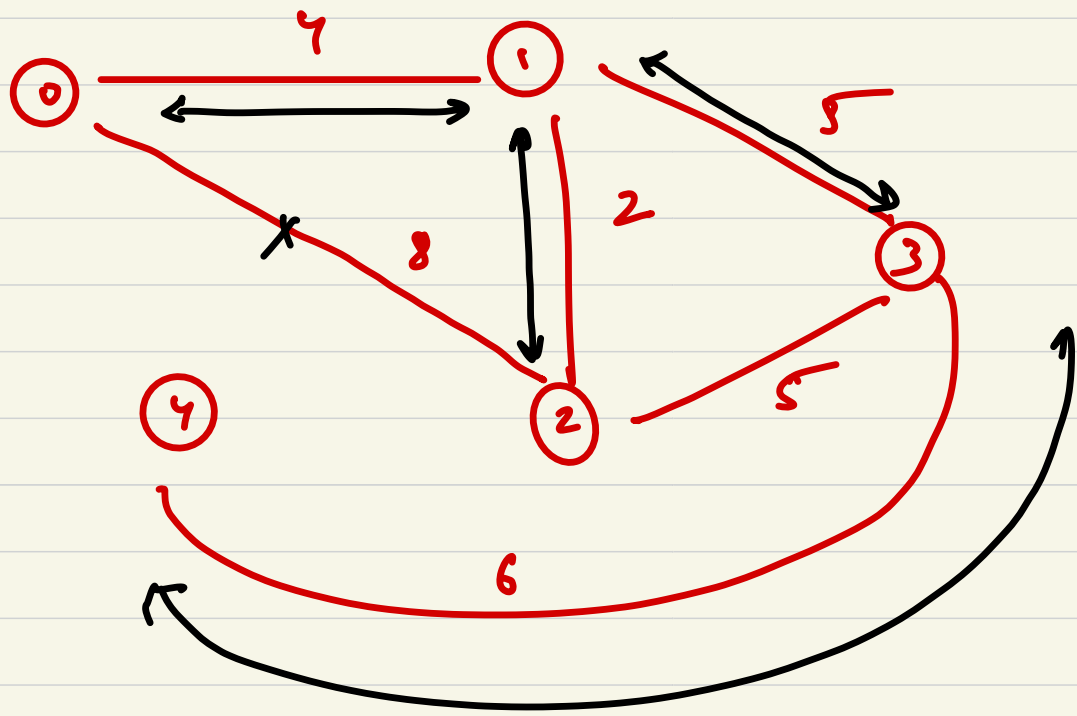
(6, 1) (1, 2)

(3, 5) (4, 5)

(5, 1)

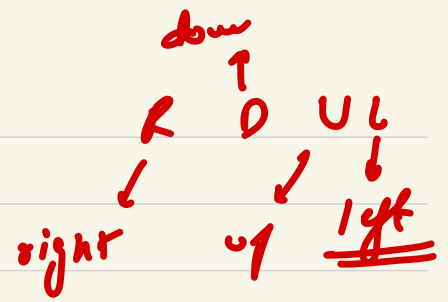
(8, 4)

(7, 3) (2, 3)



2 unit cost

R	U	D	L	D
.	.	.	R	U
.	.			



min cost is  
 reached from top  
 left to bottom  
right