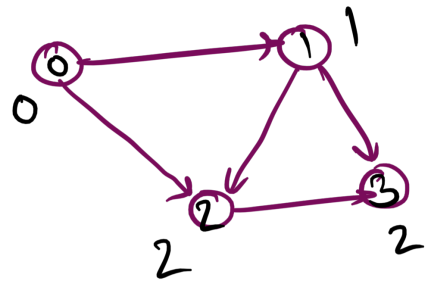
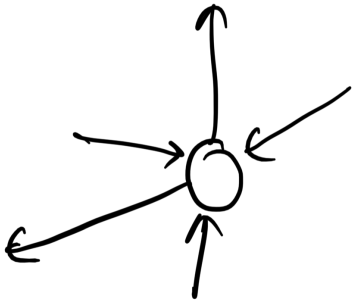
Graphs - III

InDegree of a Directed Graph

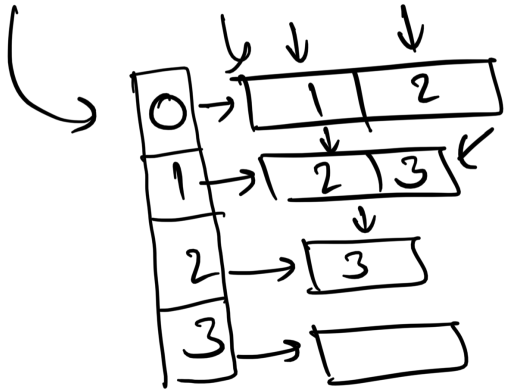


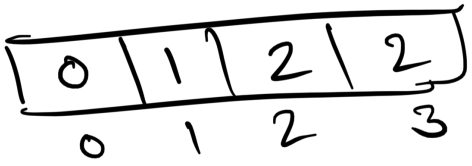
0 1 2 2.



indegree → 3

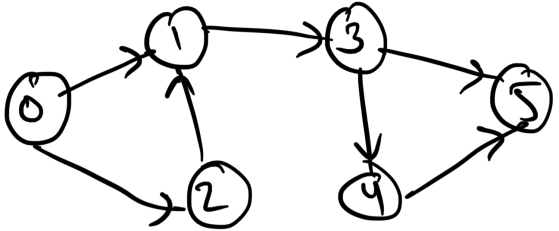
out degree → 2





indegree[]

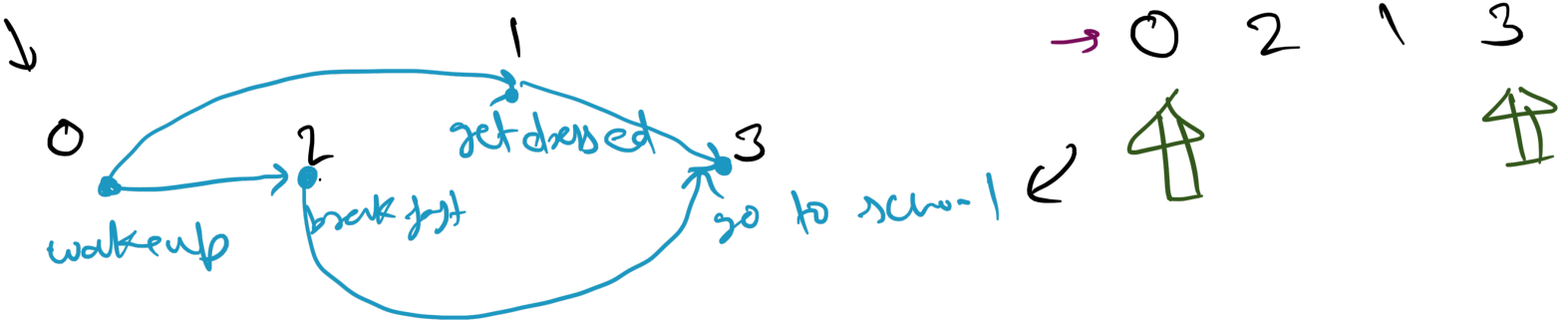
Topological Sorting in a Graph

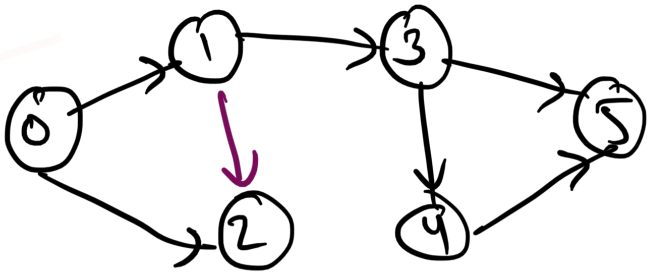


b depends on a.

→ 0 1 2 3

→ 0 2 1 3





0 2 1 3 4 5

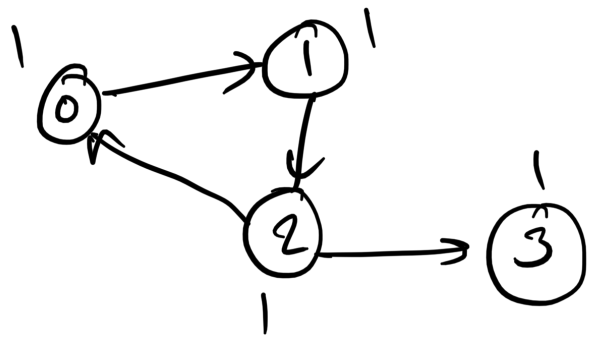
a → b

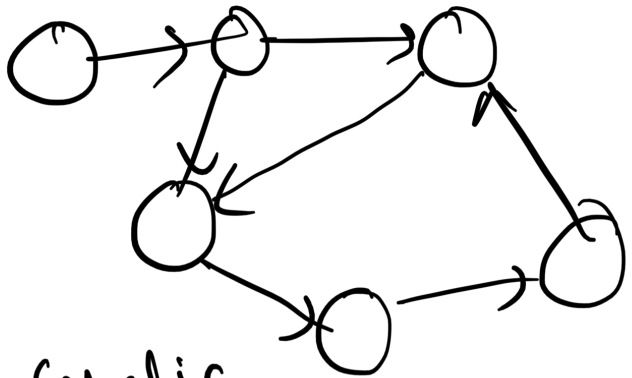
0 1 2 3 4 5

0 1 3 4 5 2 , 0 1 3 2 4 5

0 1 3 4 2 5

* Topological sorting: for every edge a → b, a appears before b.





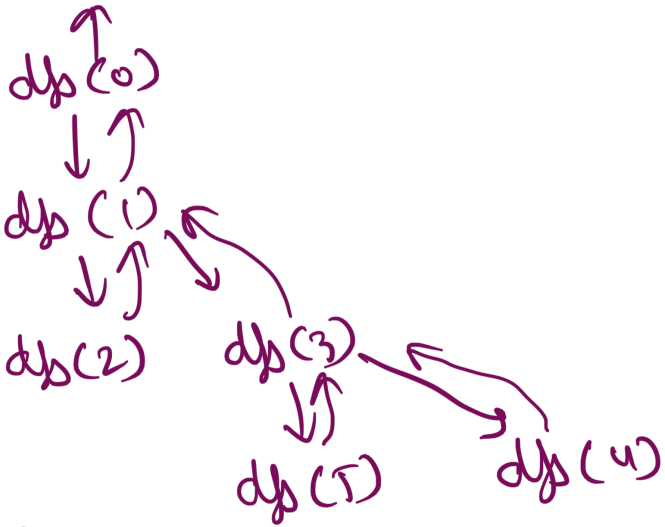
Directed cyclic Graph.

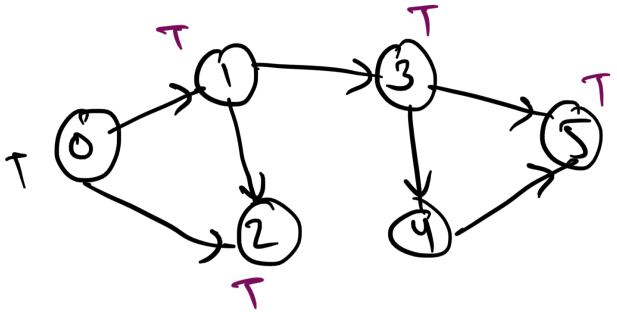
Topological sorting → Directed Acyclic Graph (DAG)

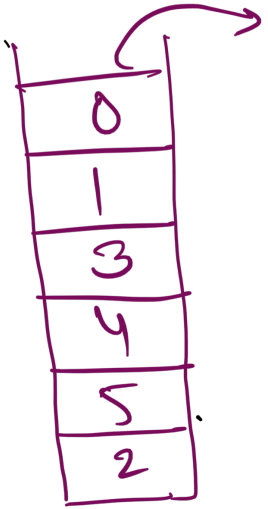
Both condition should be met for a DAG.

- At least one vertex with indeg = 0
- At least one vertex with outdeg = 0.

Topological Sort using DFS



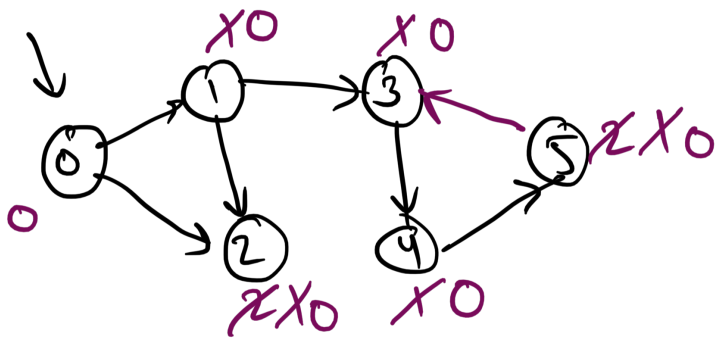




$O(V+E)$

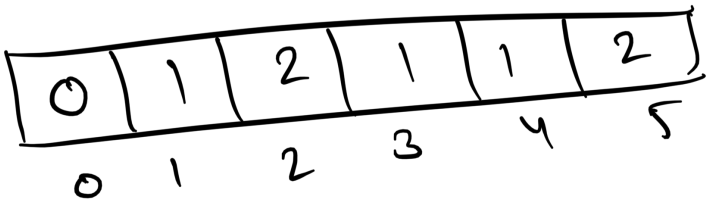
Space $\rightarrow O(V)$

Kahn's Algorithm - Topological Sort using BFS

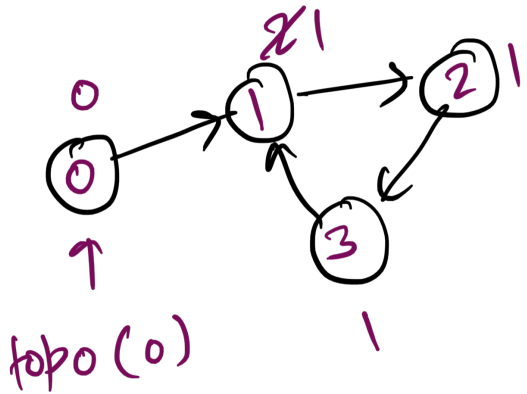


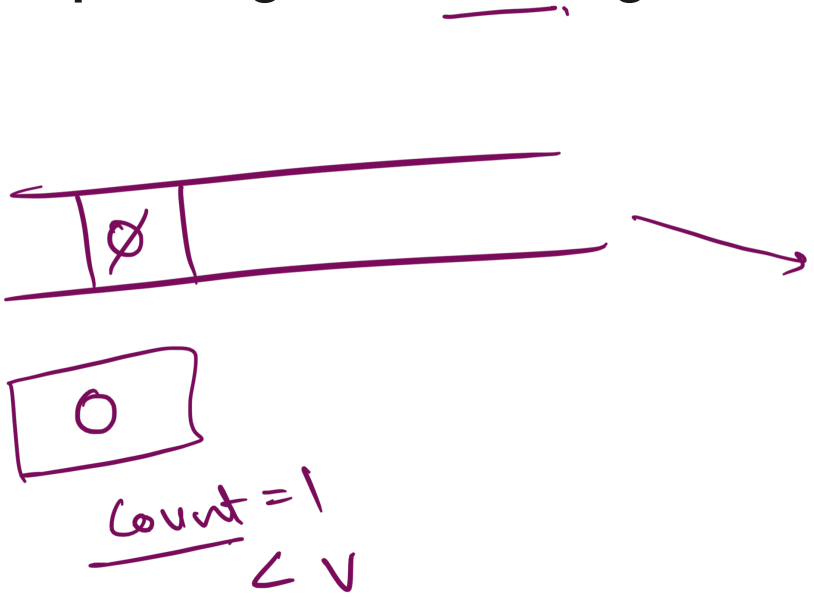
DAG.

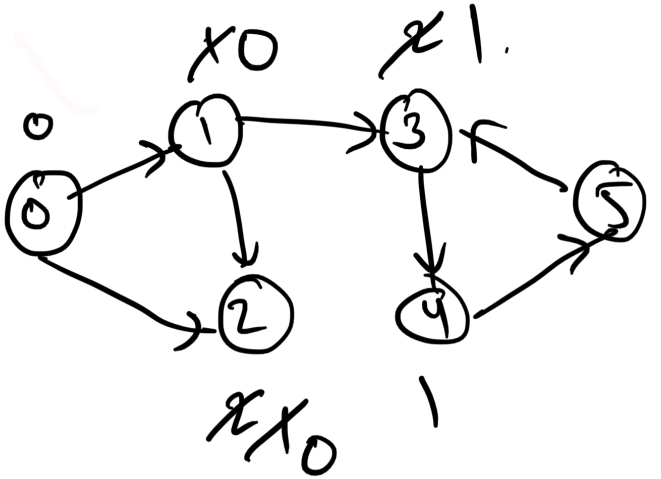
0 1 3 2 4 5

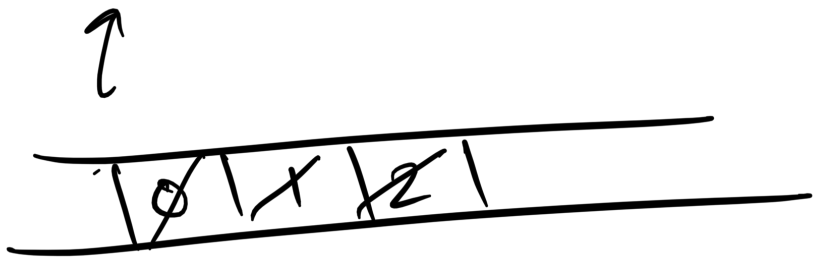


Detect a cycle in a Directed Graph using the Kahn's Algorithm



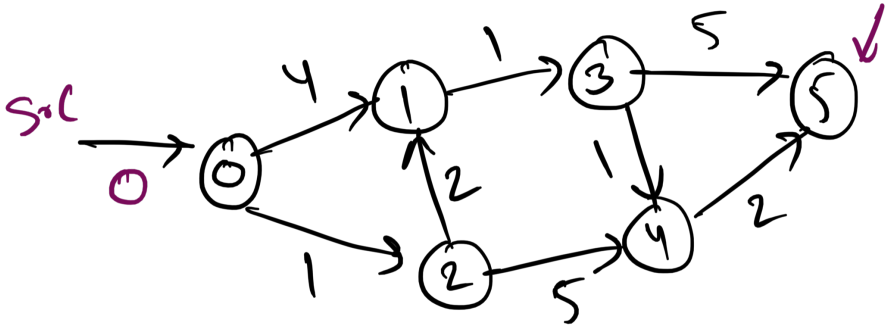






Count = 0 1 2
3

Shortest Path in a Directed Acyclic Graph



0	3	1	4	5	7
0	1	2	3	4	5

final dis

T.S. →

dis →

0	2	1	3	4	5
0	1	2	3	4	5

0	3	1	4	5	7

Rough

$4 > 1+2$

$\infty > 1+5$

$\infty > 3+1$

$\infty > 4+5$

```
for (every u in Topological sort) {
    for (every neighbor(v) of u) {
        if (dis[v] > dis[u] + edge(u,v)) {
            dis[v] = dis[u] + edge(u,v);
        }
    }
}
```

$6 > 4+1$

$9 > 5+2$

$O(V+E)$

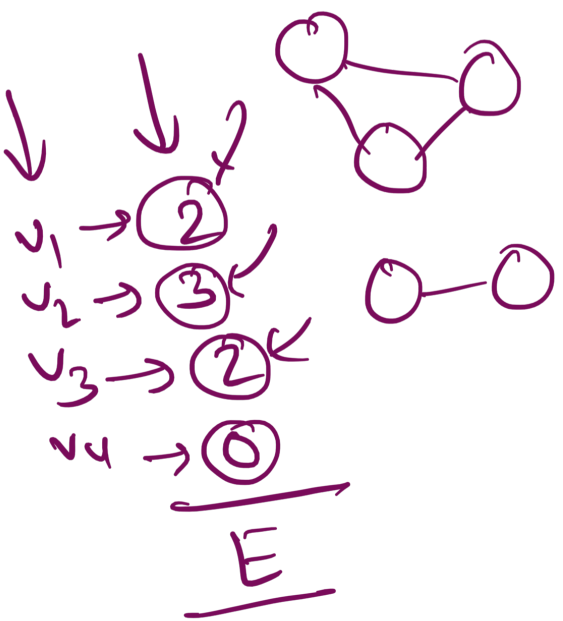
Practice.

Code This

$O(V) + O(E)$

$\Downarrow$

$O(V+E)$



$O(V * E)$ ✗

$O(V + E)$

$O(V)$, $O(E)$ ✓✗