# Topological Sort

$3 \longrightarrow 6$

3 must be completed before 6

Works only on graph with no cycles

# Directed Edges

$0 \rightarrow 5$

$0 \rightarrow 1$

$3 \rightarrow 5$

$5 \rightarrow 2$

$6 \rightarrow 0$

$1 \rightarrow 4$

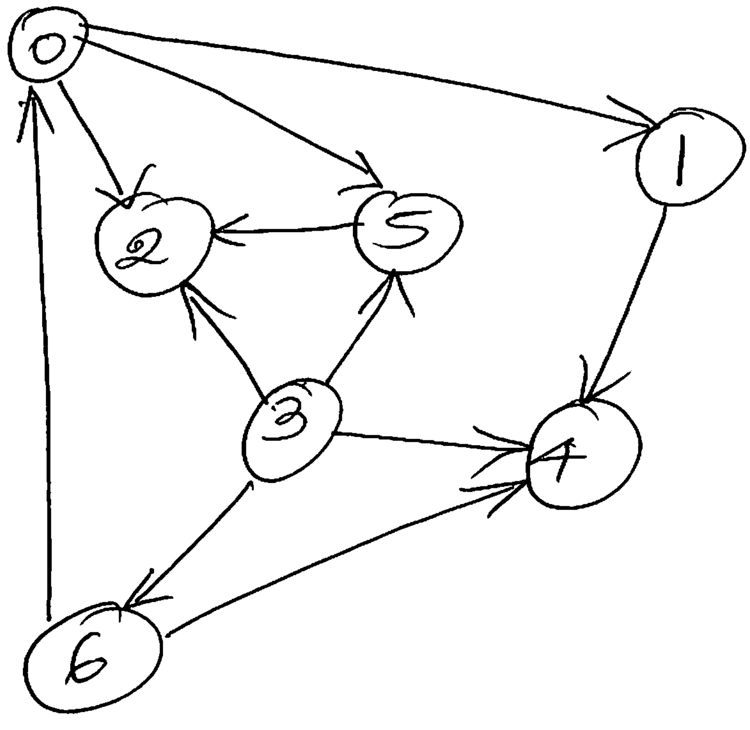
$0 \rightarrow 2$

$3 \rightarrow 6$

$3 \rightarrow 4$

$6 \rightarrow 4$

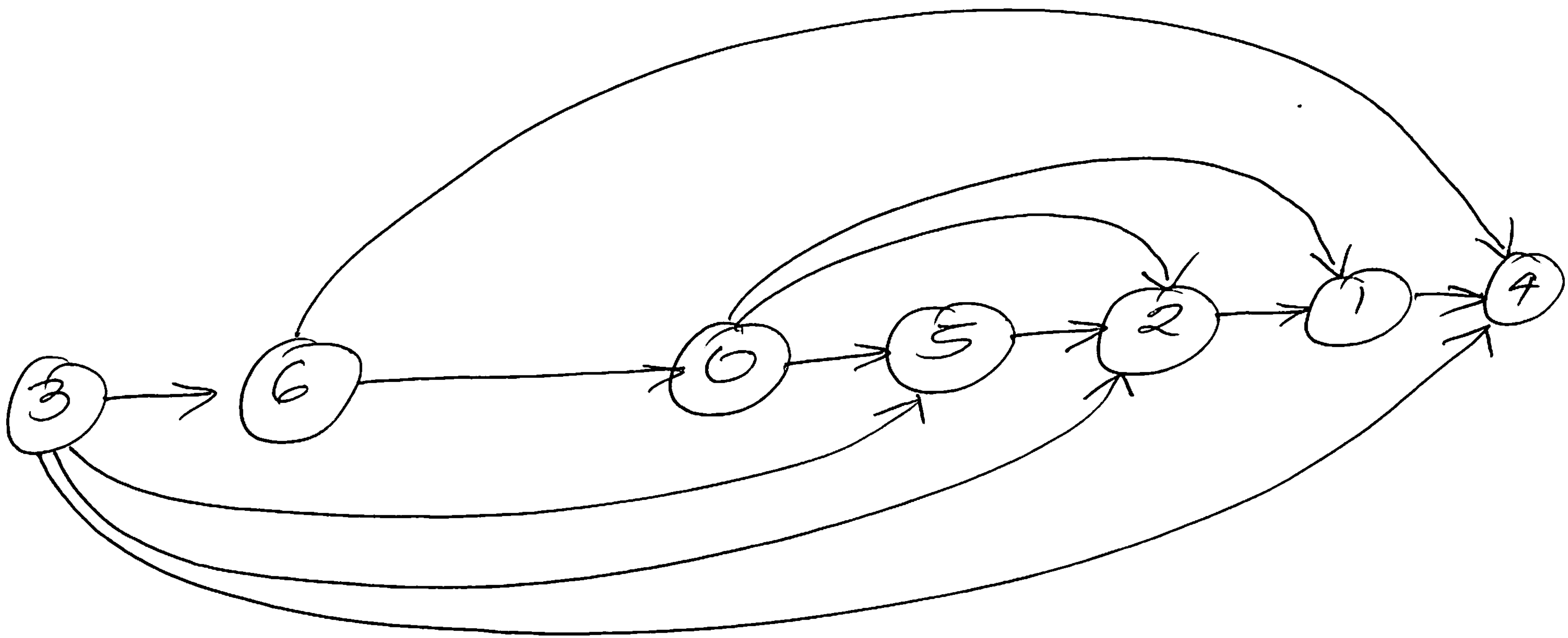
$3 \rightarrow 2$




Directed Acyclic Graph

Edges = 11 Vertices = 7

# Topological Order



Solution

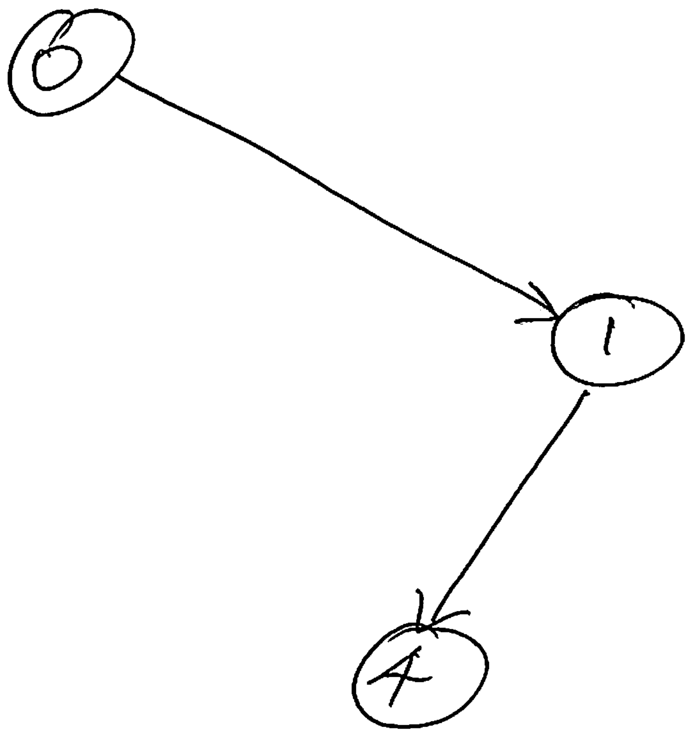
DFS

Run depth first search

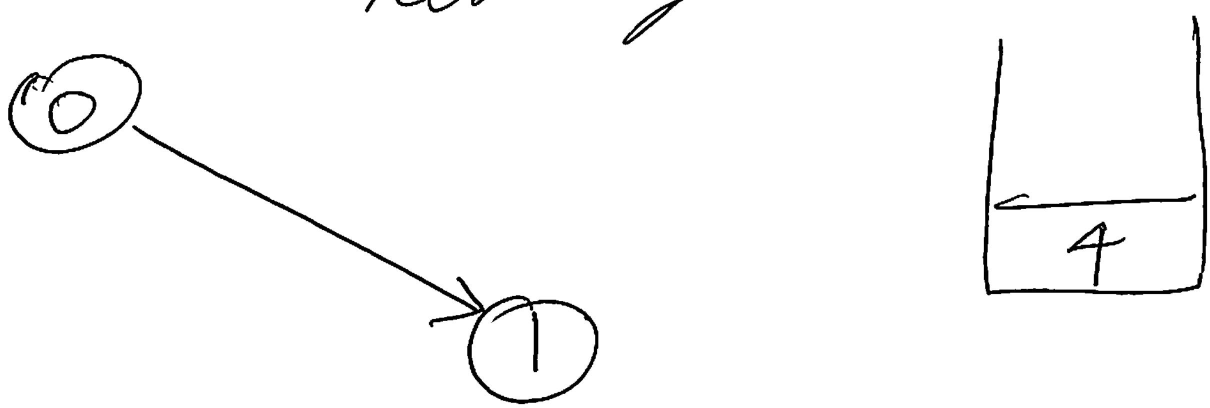
Which vertex should we pick first?

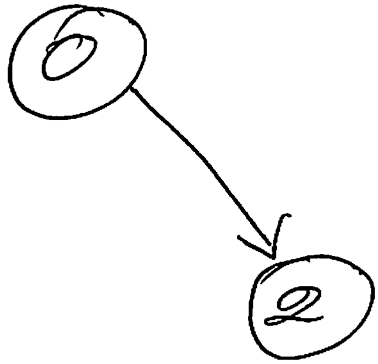
Can we pick any vertex?

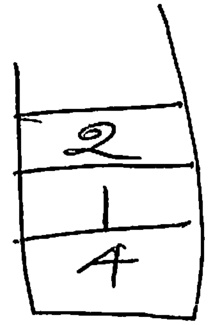
Calling phase



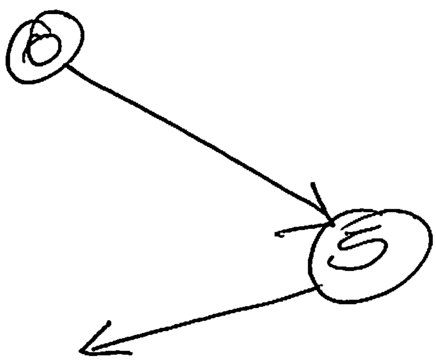

Returning Phase



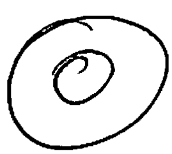



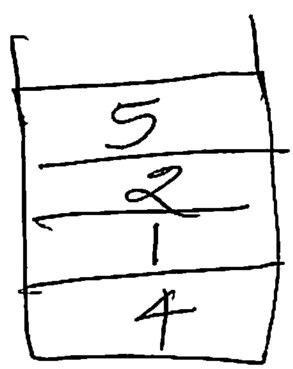








2 already visited



No more outgoing edges from 0.

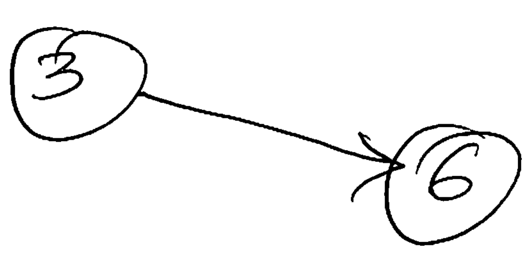
| |
|---|
| 0 |
| 5 |
| 2 |
| 1 |
| 4 |

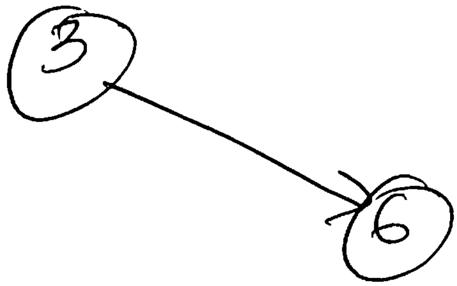
0, 1, 2, 3

↑

unvisited

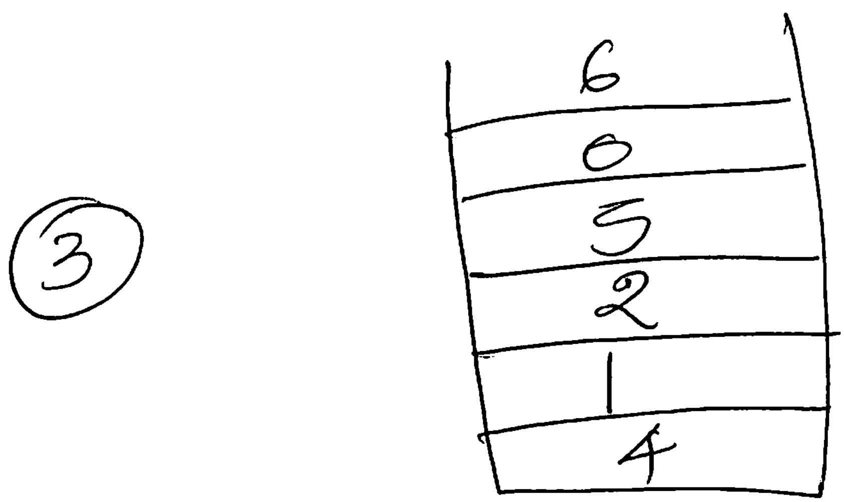
2, 4, 5. already visited



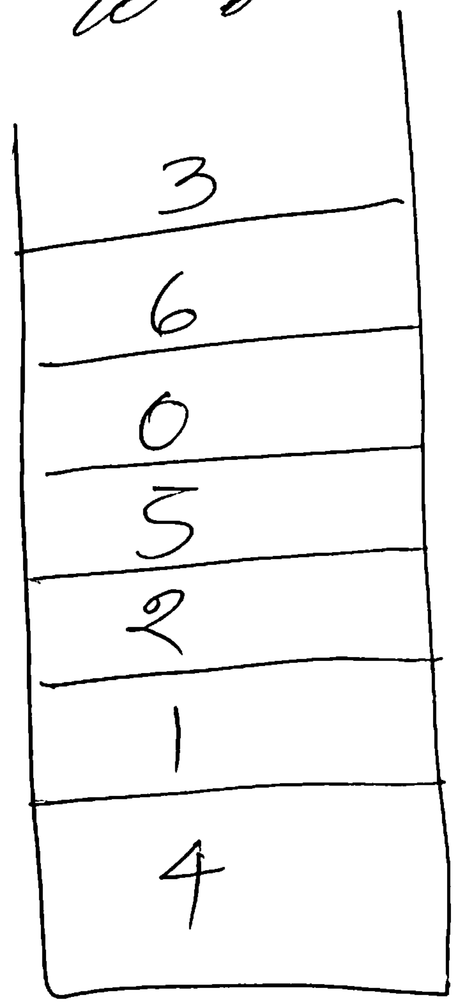



0 and 4 are already visited

No more outgoing edges from 6, add it to stack



No outgoing edges from 3
add it to stack




Answer : 3, 6, 0, 5, 2, 1, 4

The vertices are named

$0$ to $n-1$

where $n$ is the number

of vertices

We start from $0$, mark

$0, 1, 2, 3, 4, 5, 6.$

visited vertices so that we

don't explore its neighbors

We pick the next vertex

that is not visited to explore its neighbors.

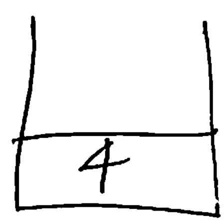
The vertex 3 is next in line for exploration

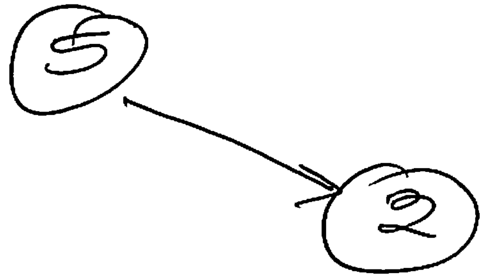
The next dfs starts from 3.

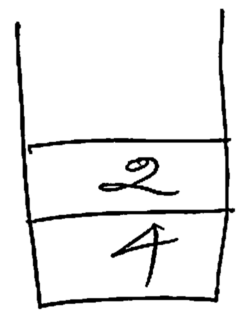
We start from 0 because its easy to go from 0 to $n-1$.

We can pick any vertex to explore. It does not have to be 0. It will still work.

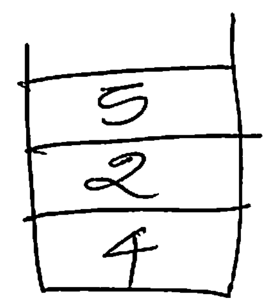
Let's start with 4, it will immediately go to stack

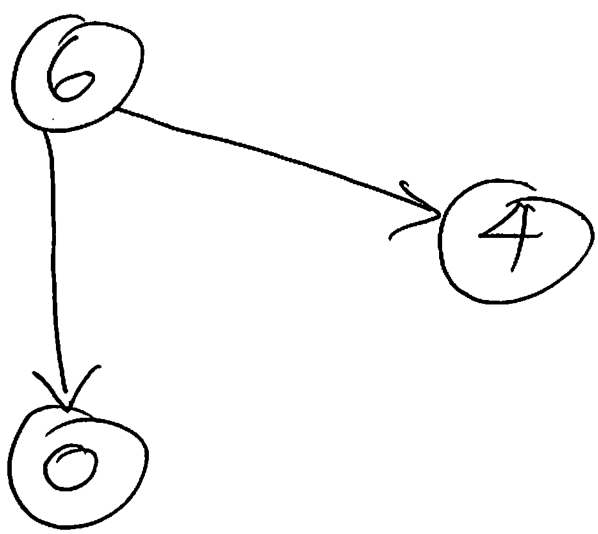



5
2

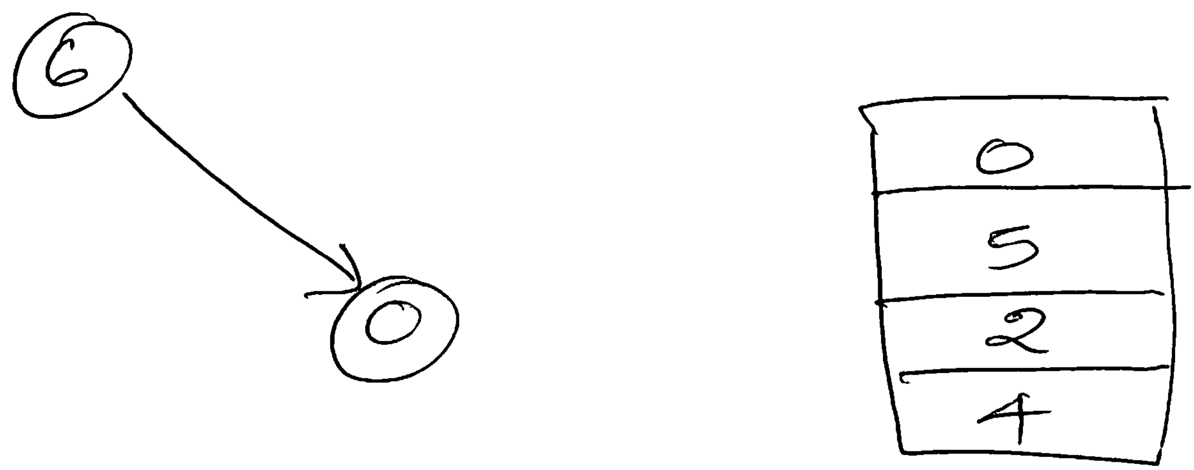

2
4


5


5
2
4


6
4
0

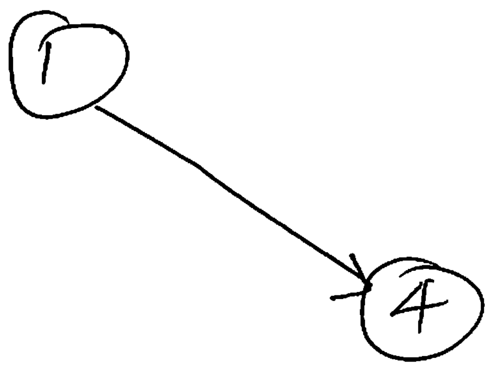
4 is already on the stack



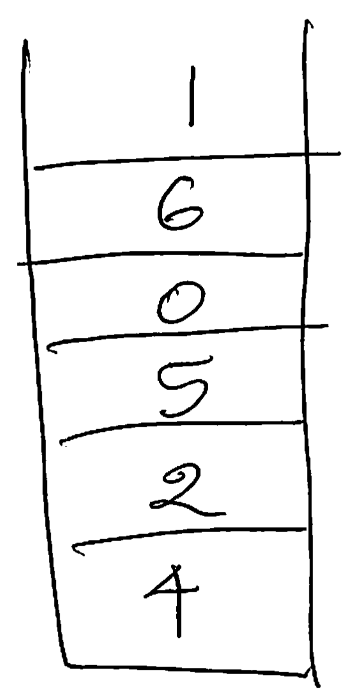


0, 1, 2, 3, 4, 5, 6

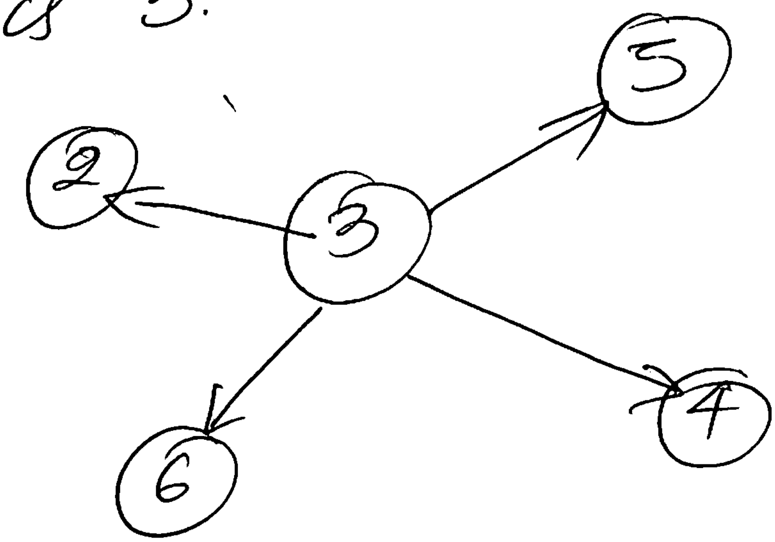
Pick 1




4 is already on stack

Last element to explore is 3.




All its neighbors are already on the stack

| |
|---|
| 3 |
| 1 |
| 6 |
| 0 |
| 5 |
| 2 |
| 4 |

$3 \rightarrow \underline{1} \rightarrow 6 \rightarrow 0 \rightarrow 5 \rightarrow 2 \rightarrow 4$

We can have more than one topological sorting of a graph

We can think of an edge (a, b) as a has to come before b.

Thus an edge defines a precedence relation.

Topological order is an order of the vertices that satisfies all the edges.

$3 \to 1 \to 6 \to 0 \to 5 \to 2 \to 4$

$0 \to 5$

$0 \to 1$

$3 \to 5$

$5 \to 2$

$6 \to 0$

$1 \to 4$

$0 \to 2$

$3 \to 6$

$3 \to 4$

$6 \to 4$

$3 \to 2$

Does the order of the vertices satisfy all the edges?

$\longrightarrow$

must come before

$\boxed{\text{socks}} \longrightarrow \boxed{\text{shoes}}$

In DFS, the vertex finished last will be first in the topological order

Claim

The vertex finished last by DFS cannot have any incoming edges.

# A Standalone Algorithm

1. Find a vertex with no incoming edges, put it in the output
2. Delete all its outgoing edges
3. Repeat