RECAP

— Agents

— Perception, Interaction, Learning & Rationality

System	Percept	Intent	Learning	Rationality
Roomba	Cameras, Radar/Lidar	Movement, Obstacle avoid, Clean	Rules: Spiral until wall, then sweep OR Build grid-search	Cover most ground, without redundancy
Youtube/ Netflix/ Amazon Recommendations	Browsing History Text or Graph	Predictions	Build Similarity-based models of user	Maximize Time/Money spent on platform

- Fully observable v/s partially observable
- Single-agent v/s Multi-Agent

↓

Do agents' performance depend on each other

- Deterministic v/s Non-Deterministic

↓

Next State is defined completely by present state & action

Randomness in environment or agents' behaviour or both

↓

Stochastic ⇒ Explicit Probabilities.

- Episodic v/s Sequential

↓

Current State & Action have no impact on future states

Assembly Line defect check

Driving → short term decisions affect future states

- — Static v/s Dynamic

- — Discrete v/s Continuous

- — Known v/s unknown

Knowledge abt 'rules' of the Environment rather than state of the env.

Solitaire → Known but Partially Observable

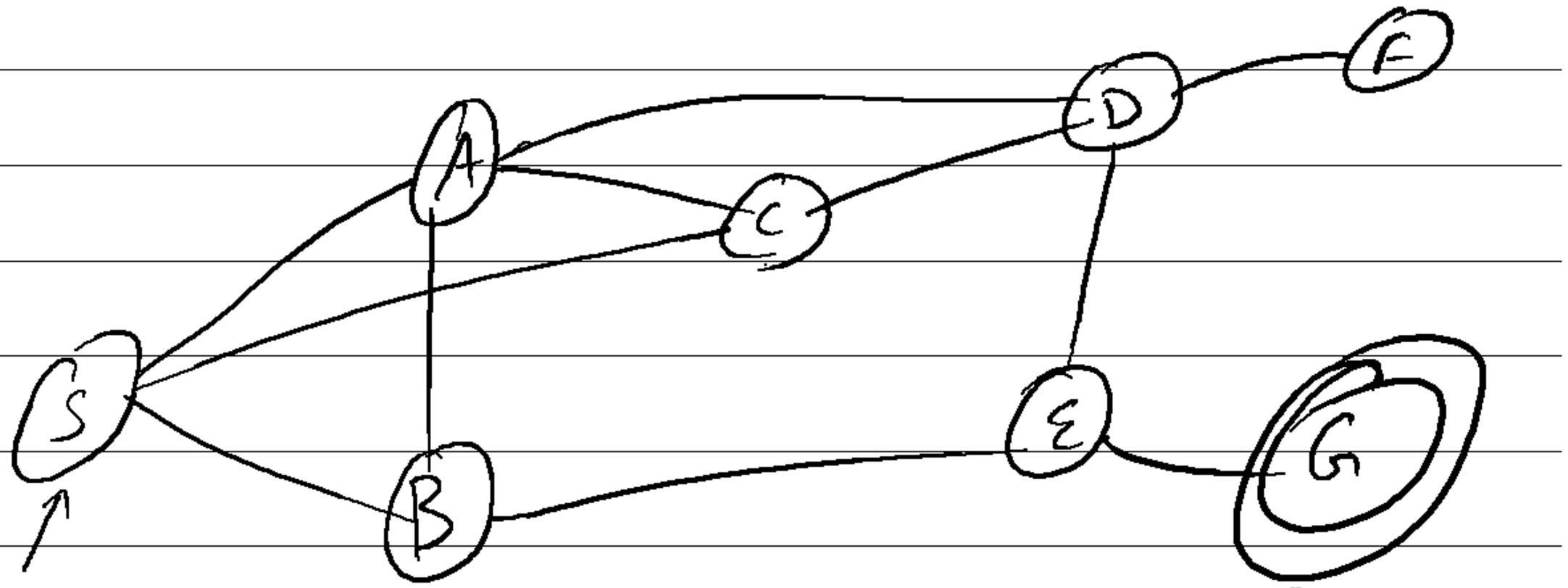
Tesla FSD in a city it wasn't trained in

SEARCH

Fully Observable, Static, Deterministic, Known Env.

The Solution is a fixed sequence of actions.

Route Planning ? → Example on Board



Initial State

Action / Transition

Cost

Goal State

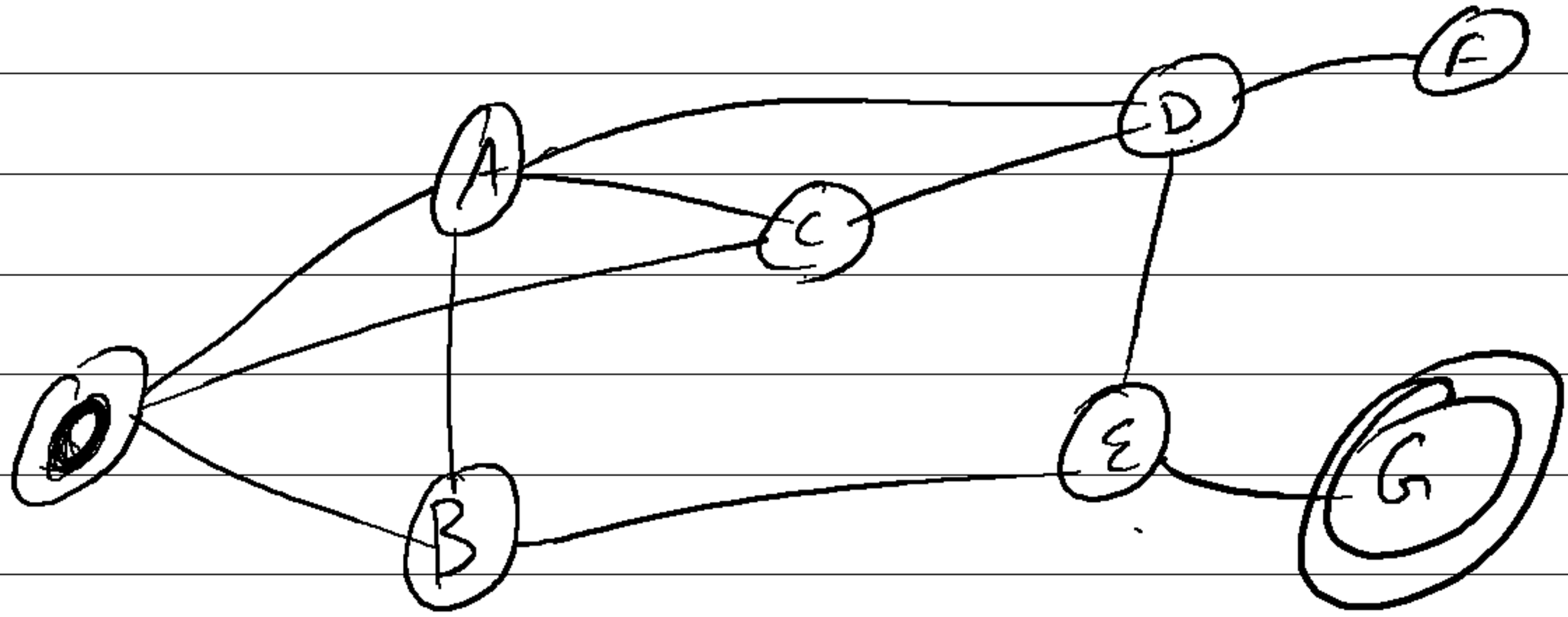
Mazes as a search problem

Example in Slides

8-Puzzle

Basic DFS over search Tree

Implement using Stacks or Recursion



O

/

A

1

B

1

E — G

1

D

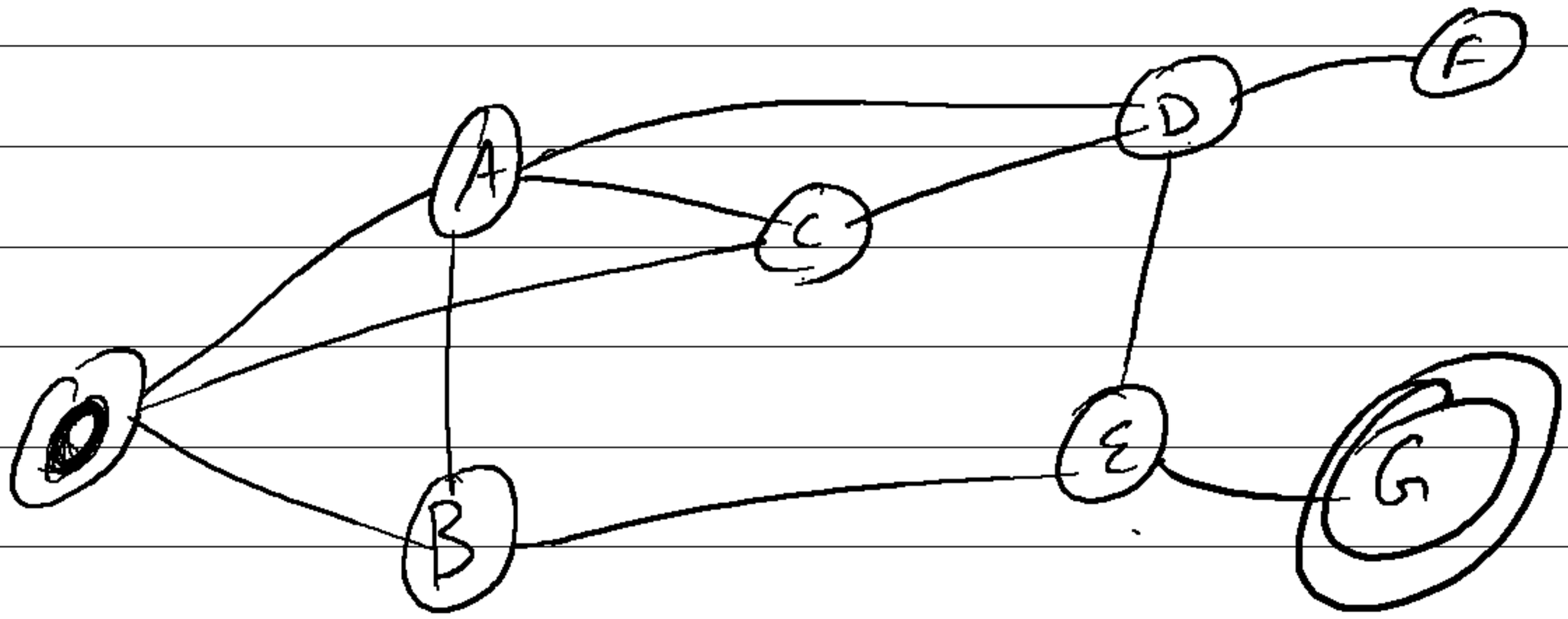
1

C F

Visited :

Path :

Stack :

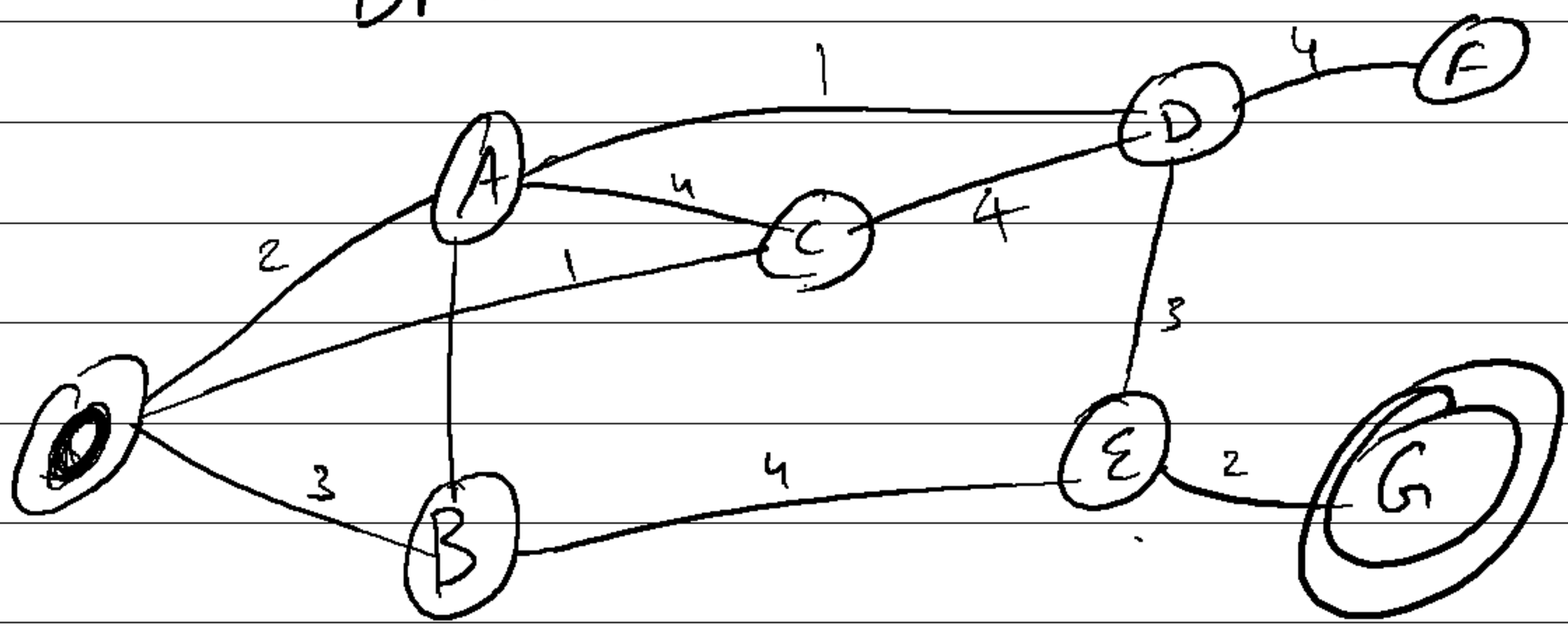


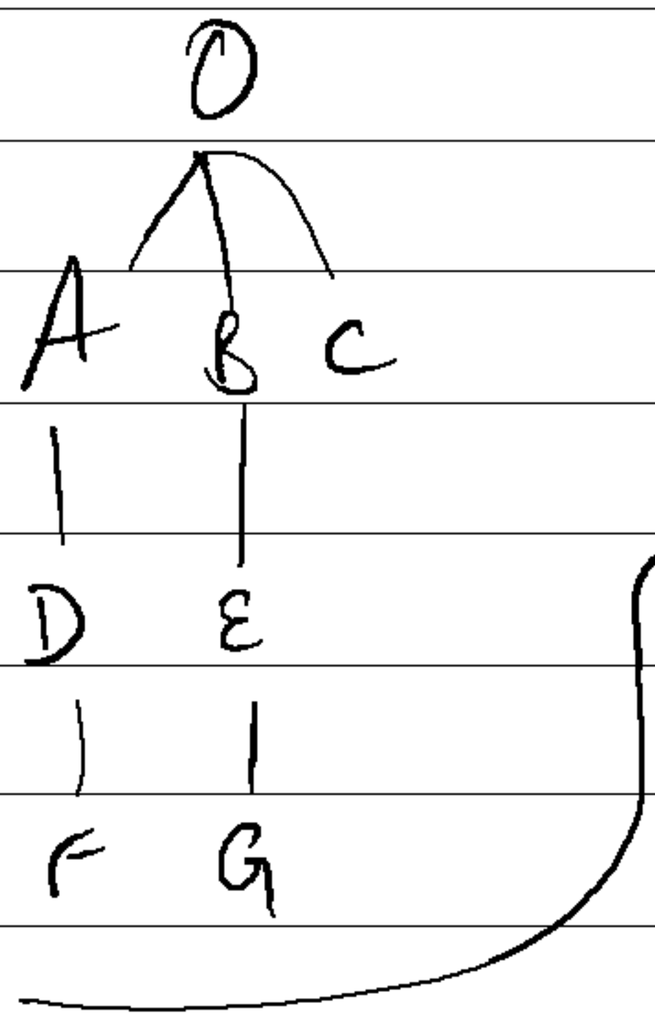
RECURSION

Visited = { }

```
dfs(G, V, Vis):
    If v ∉ Vis:
        Vis.append(v)
        ∀ x ∈ N(v)
            dfs(G, x, Vis)
```

BFS





Queue:

Visited: O A B C D E F G