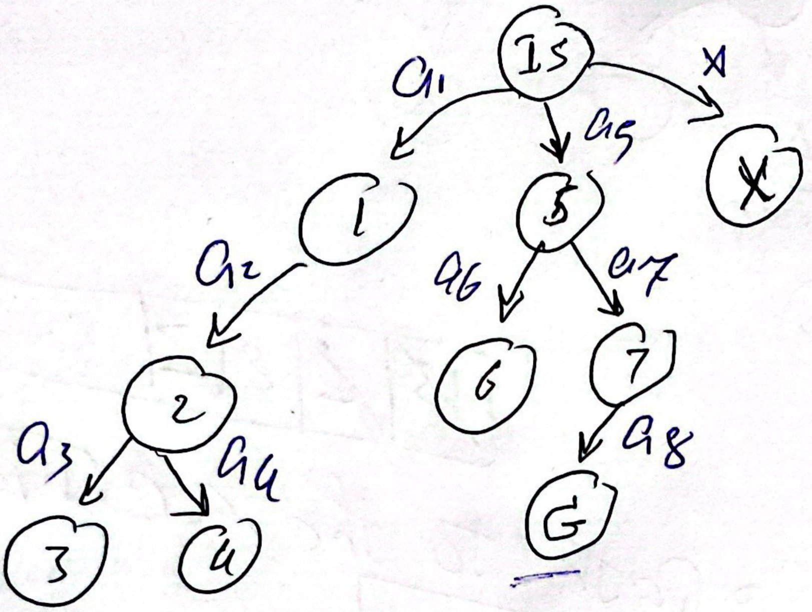
Dfs



• we will not visited this node .

Vis

IS	1	2	3	4	5	6	7	G

the solution path should be:

$\boxed{a8 \; a7 \; a5}$ reversed

• So we simply will apply the Dfs logic & concatenate the actions if the return was True.

DFS with Solution Algorithm:(problem, initialState)

```
3. frontier = deque() No need
1. if (problem.is_goal(initial state))
        1. return []
2. Path = None
4. explored = map which contains the privious visited nodes
5. In Queue, map which " " nodes inside the Queue.
6. Path = dfs (explored, In Queue, problem, initialState)
7. return Path reversed
```

```
DFS ( Problem, ~~initial~~ state, grey, black & Path)
    1. if( grey (state) or black (state))
        1. return ~~None~~ false
    2. actions = problem.get_actions(state)
    3. for action in actions
        1. next_state = problem.get_successor(~~state~~ state, action)
black = True → 2. goal found = DFS(problem, next state, grey, black)
        3. if goal found
            1. Path.append (action)
            2. return True
    4. return false # No solution found
```

8/8

11/8/23