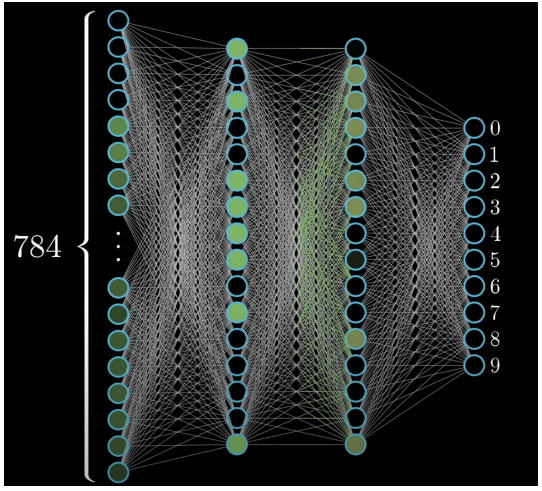
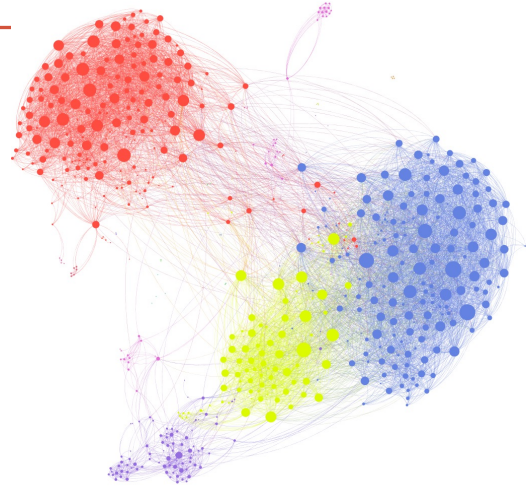
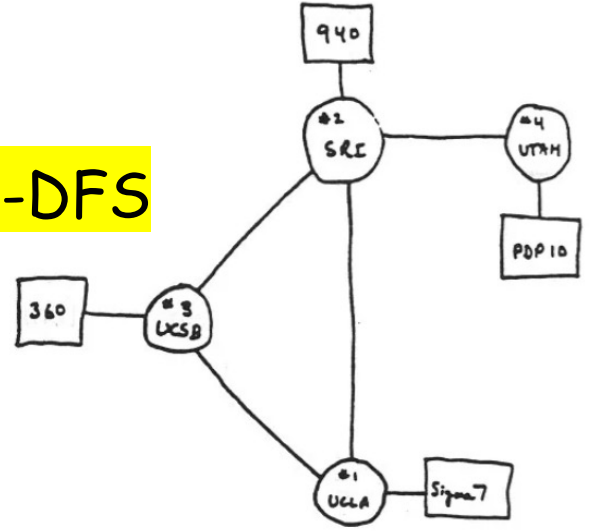
Link to Handout:
https://bit.ly/CS24-W24-Handout-Graphs-DFS

DEPTH FIRST SEARCH





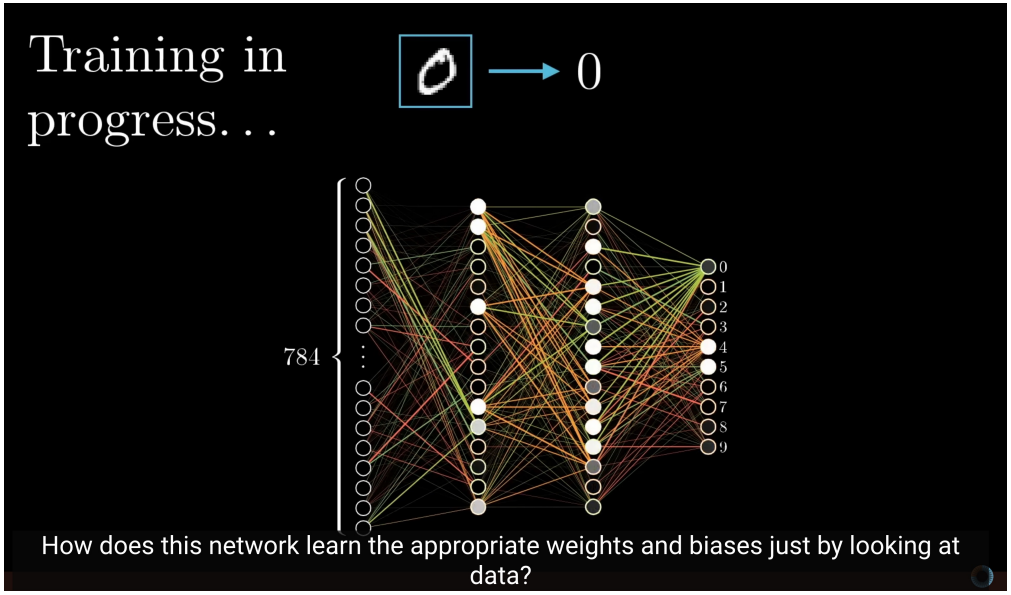


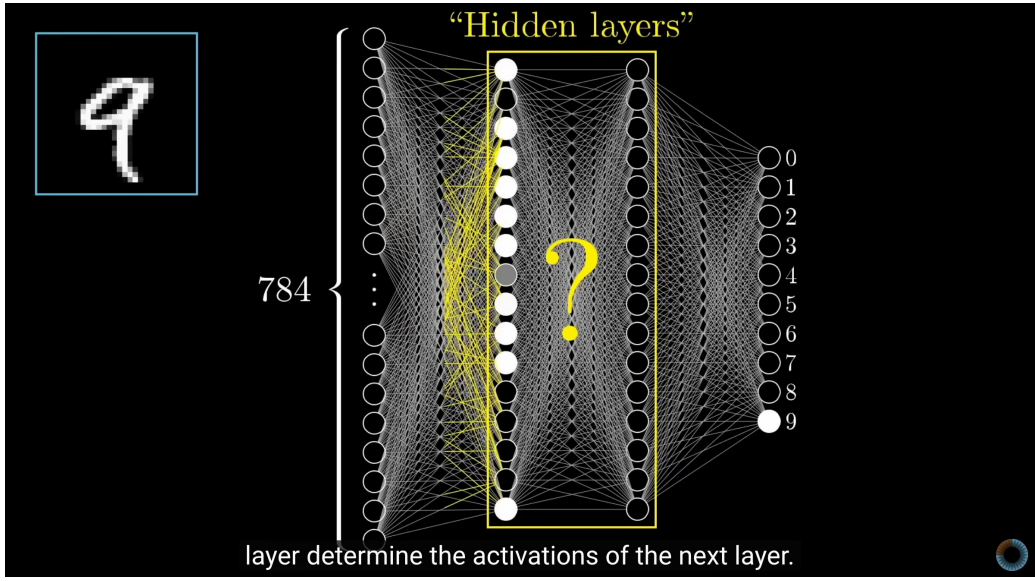
THE ARPA NETWORK

DEC 1969

4 NODES

How does information flow in a NeuralNetwork ?





Training
Learn network parameters

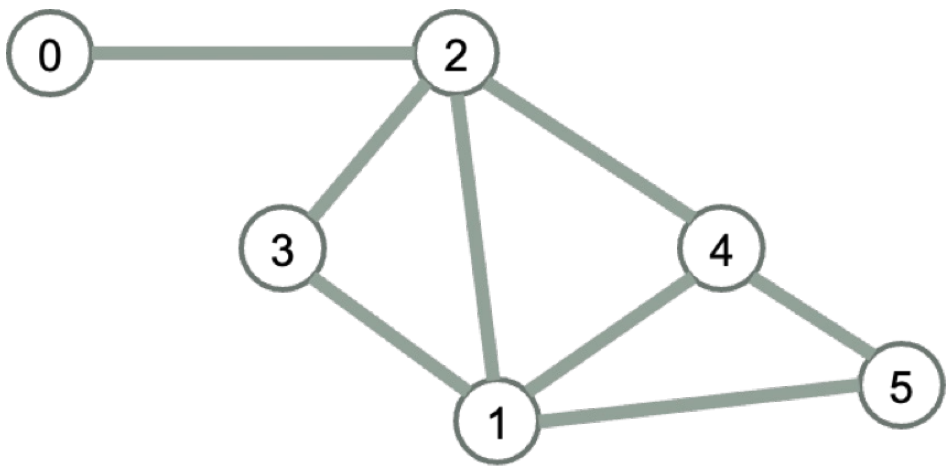
Evaluation/Prediction
Network produces outputs from inputs

Credits: 3Blue1Brown

Graph search: general approach

```
Keep track of all areas discovered

While there is an unexplored path, follow path
```

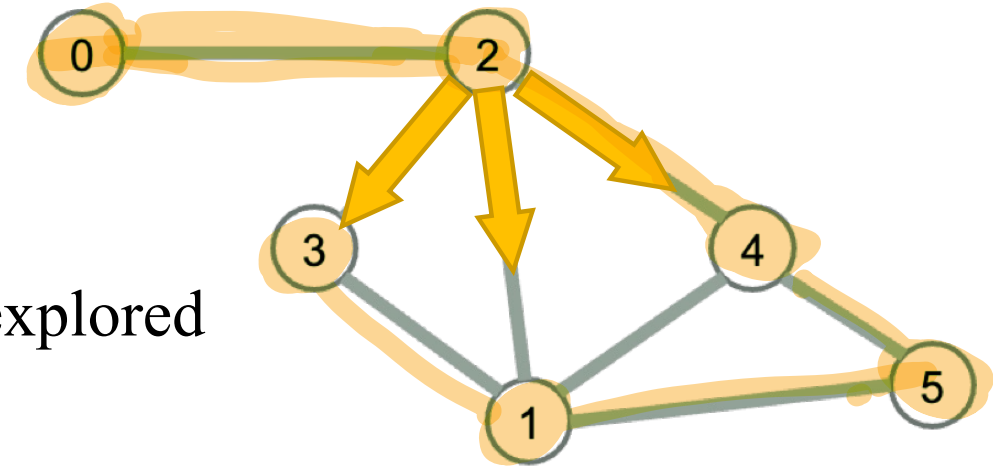


Slide credits: Professor Daniel Kane, UCSD

Systematize the Search

Need to keep track of:

- Which vertices discovered
- Which edges have yet to be explored



`exploreDFS` Algorithm will:

- Use a field `v.visited` to let us know which vertices we have seen.
- Store edges to be explored implicitly in the program stack.

Slide credits: Professor Daniel Kane, UCSD

Explore – Depth First

```
exploreDFS(v)
  v.visited ← true
  For each edge (v,w)
    If not w.visited
      exploreDFS(w)
```

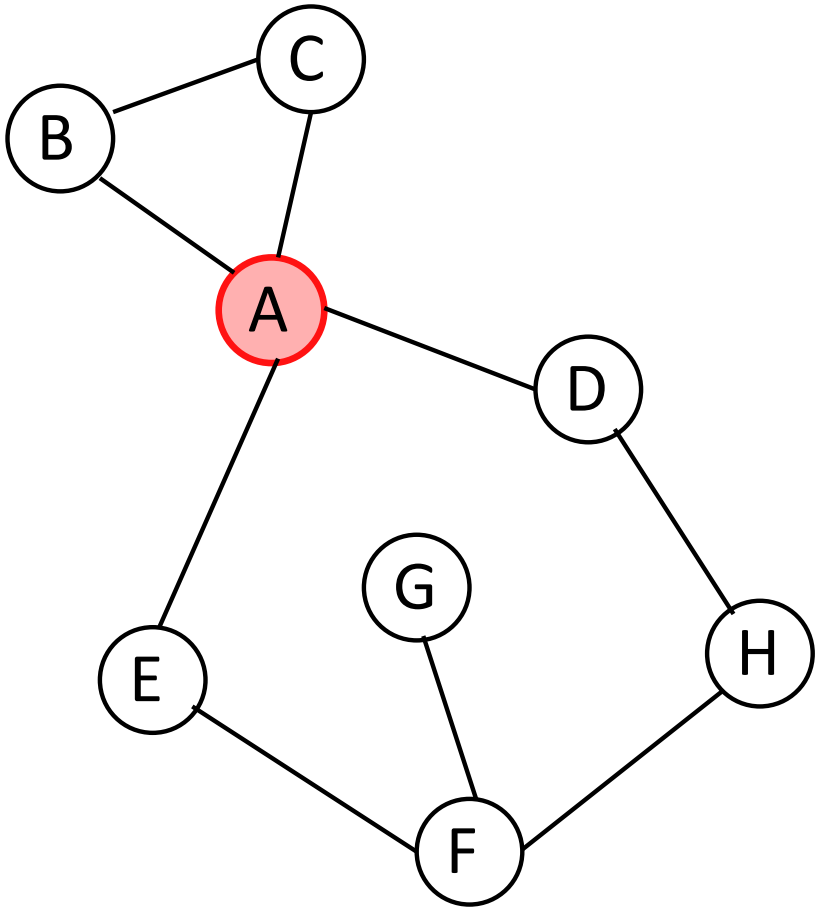
Slide credits: Professor Daniel Kane, UCSD

exploreDFS: Explore – Depth First

```
exploreDFS(v)
  v.visited ← true
  For each edge (v,w)
    If not w.visited
      exploreDFS(w)
      w.prev ← v
```

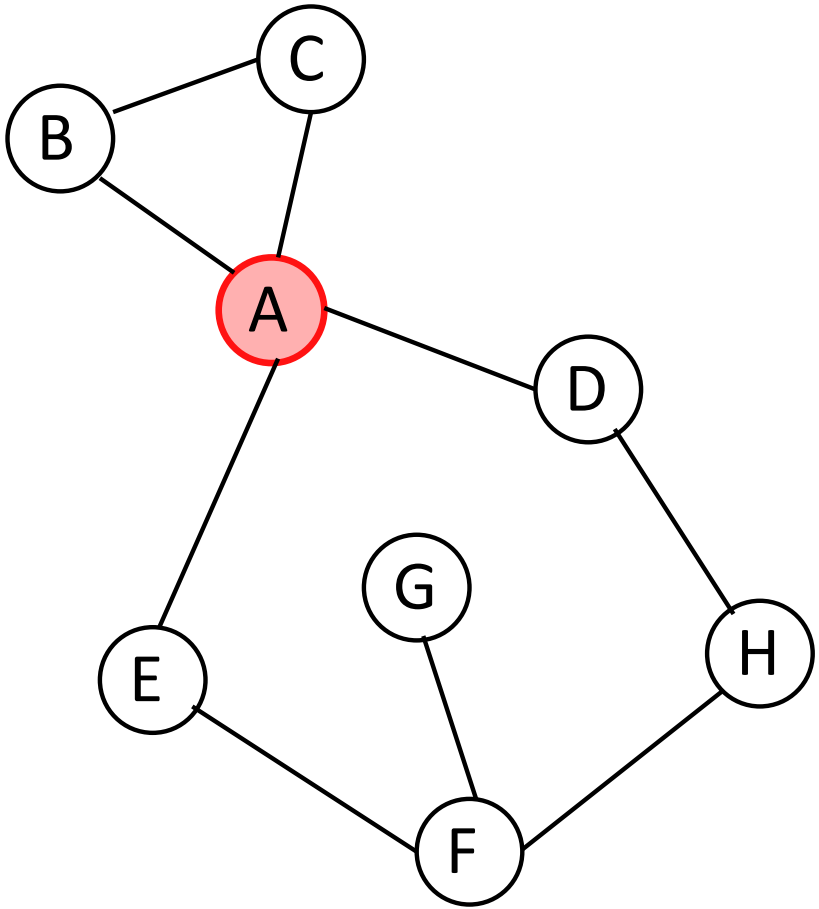
Slide credits: Professor Daniel Kane, UCSD

Explore (Depth First): Example




```
explore(A)
```

Explore (Depth First): Example



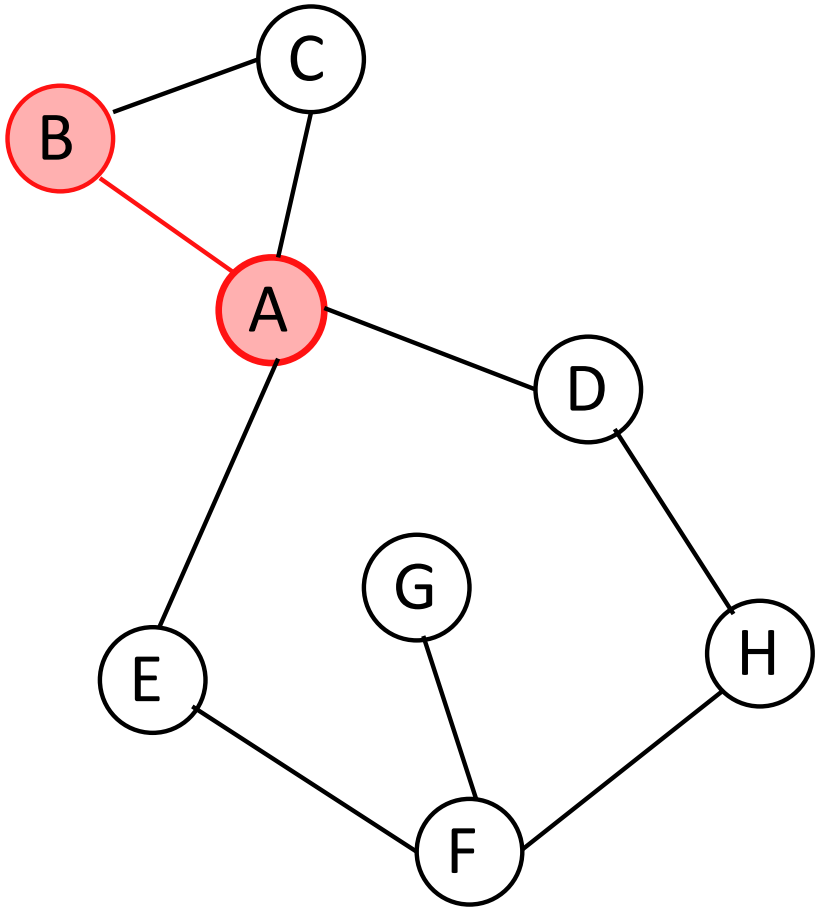

```
explore(A)
  explore(B)

  explore(C)
  explore(D)

  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



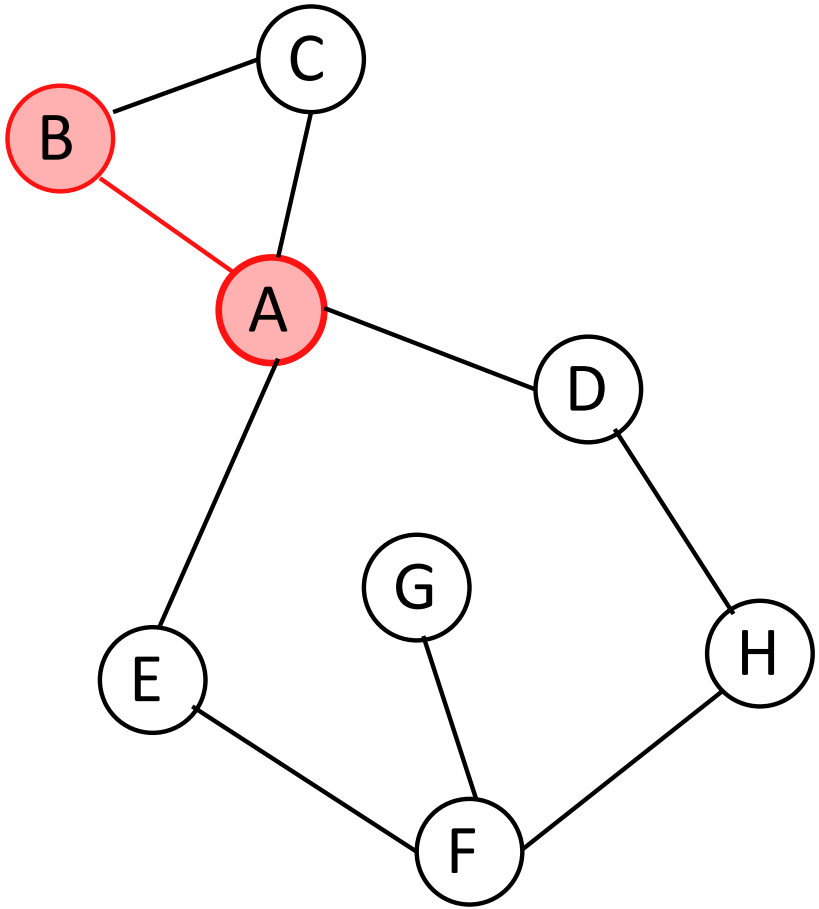

```
explore(A)
  explore(B)
```

```
explore(C)
explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



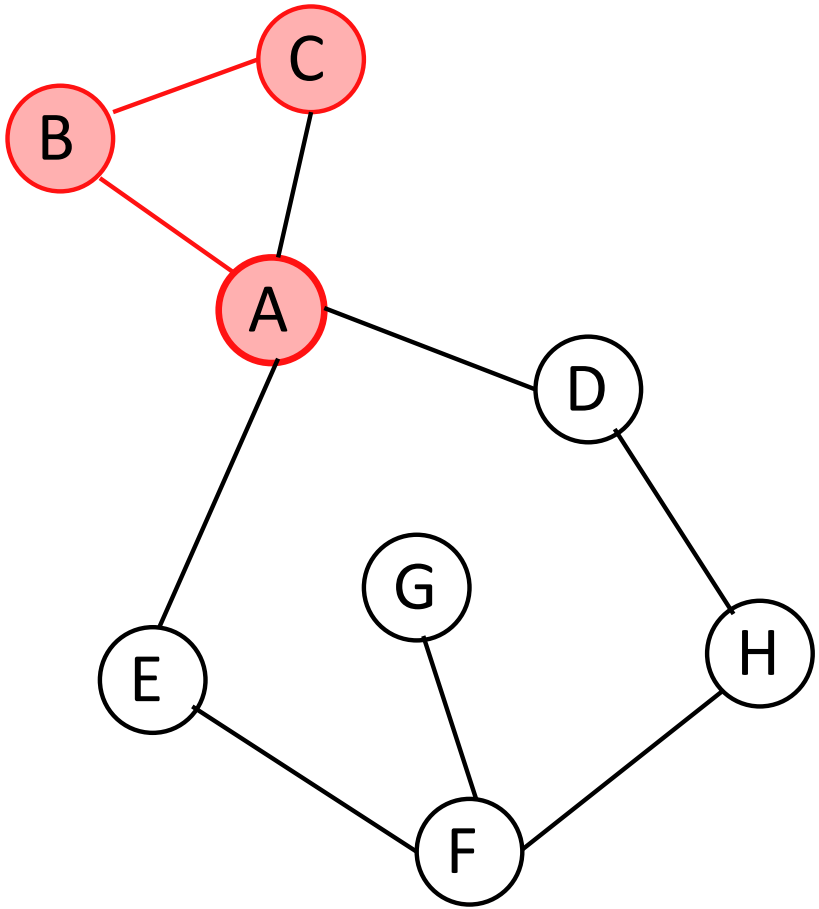

```
explore(A)
  explore(B)
    explore(A)
    explore(C)

  explore(C)
  explore(D)

  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



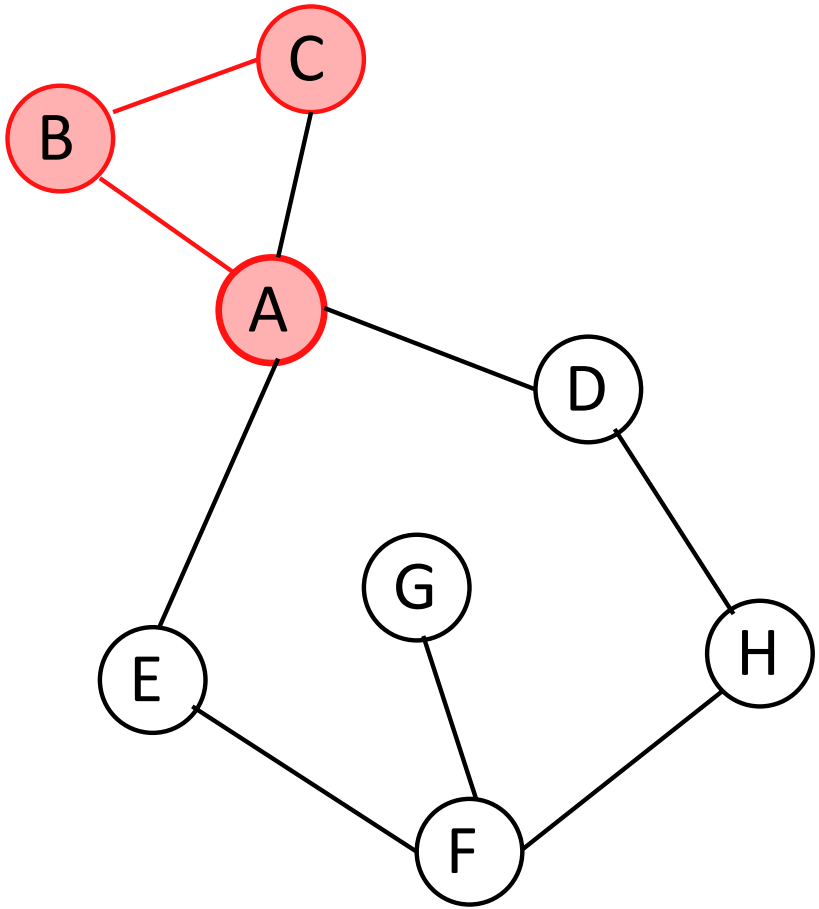

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
```

```
explore(C)
explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

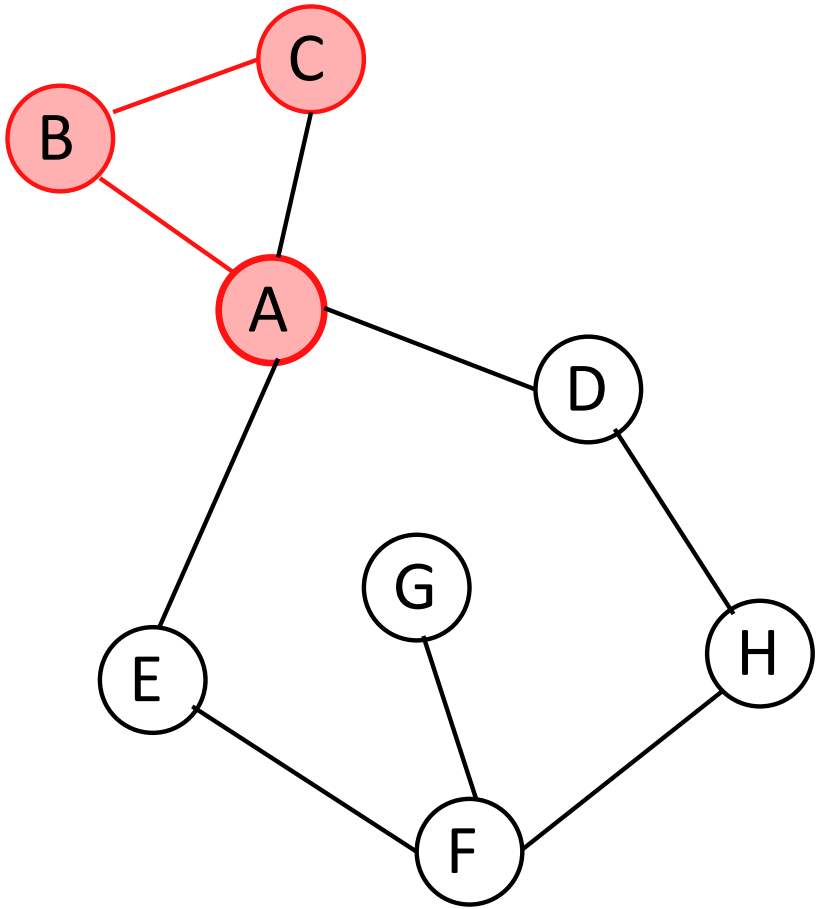



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

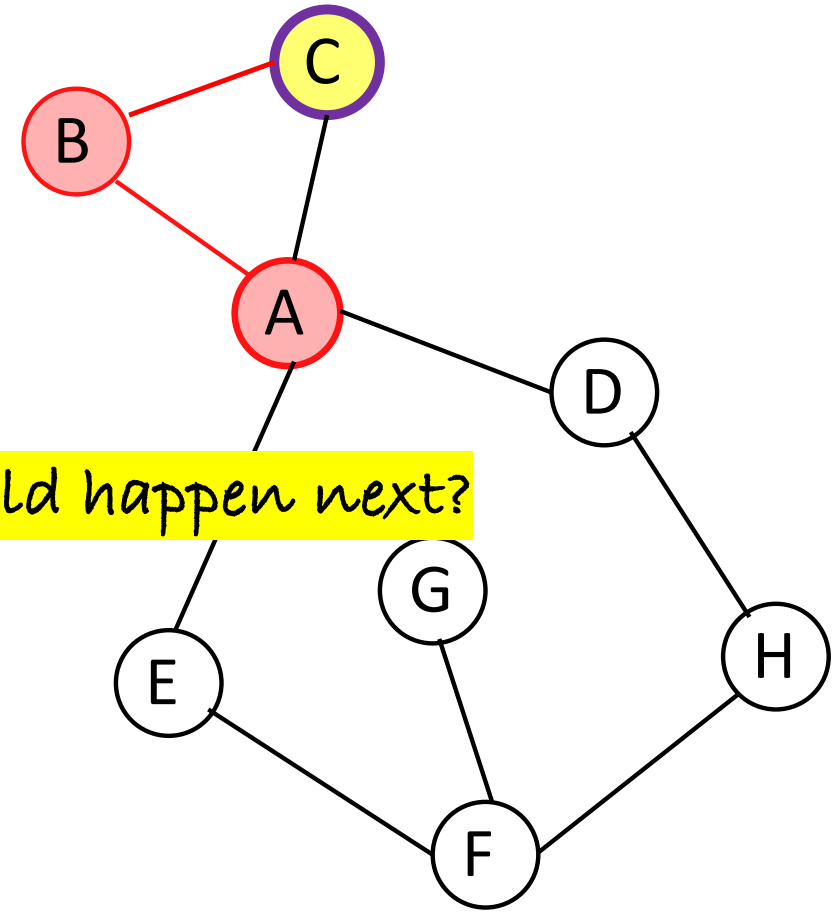



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



Dead end!

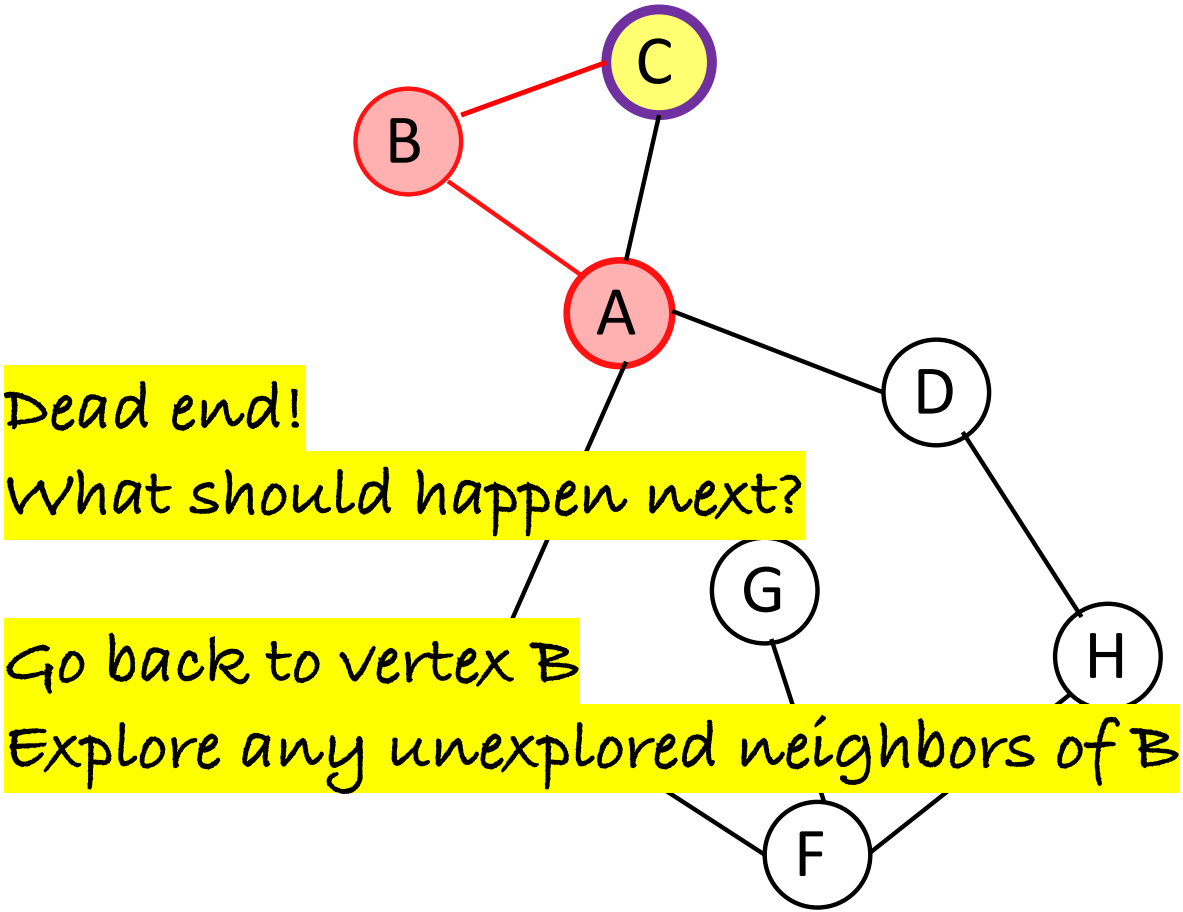
What should happen next?

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

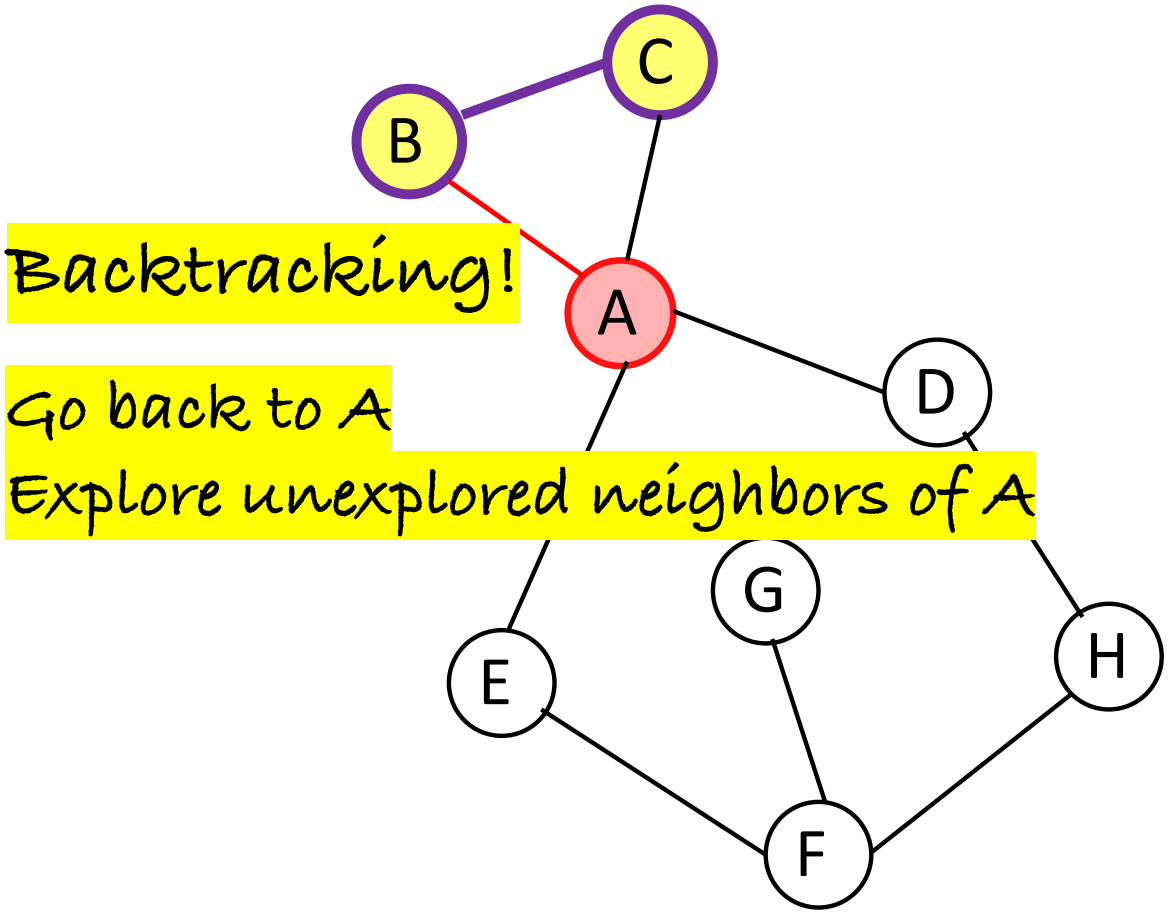



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

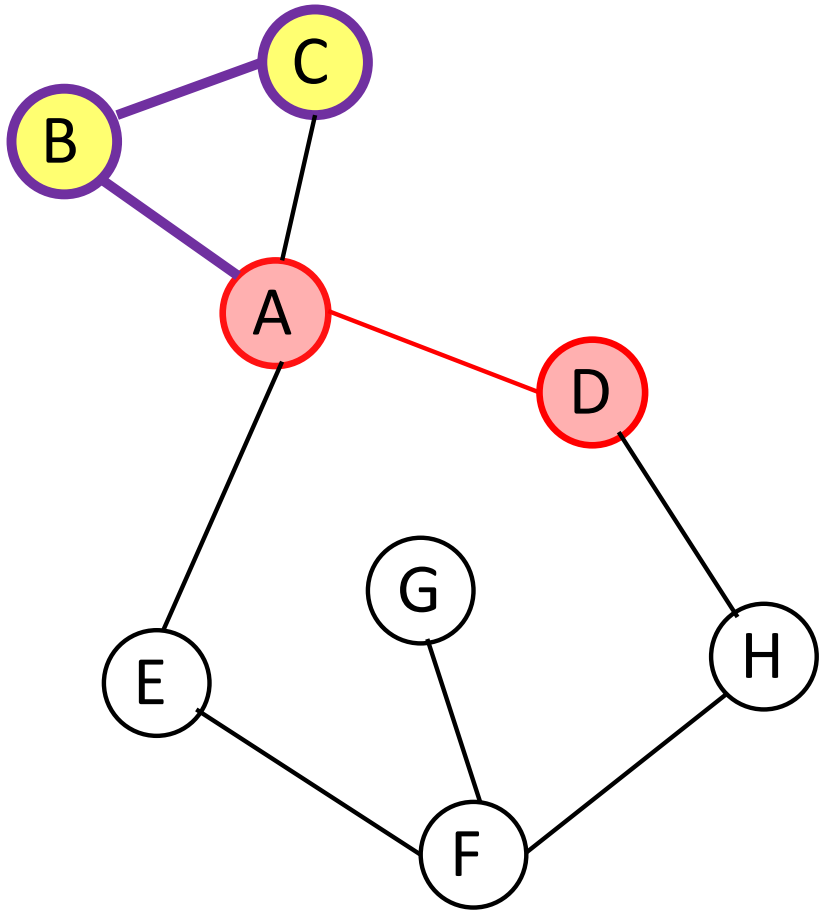
Explore (Depth First): Example




```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Explore (Depth First): Example

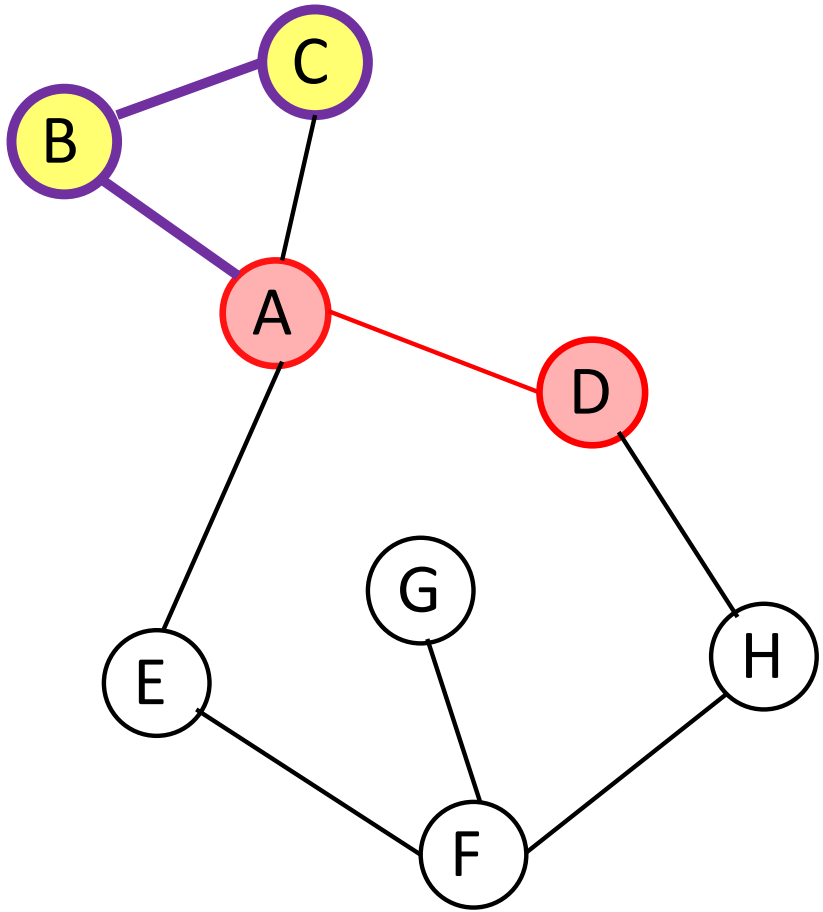



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

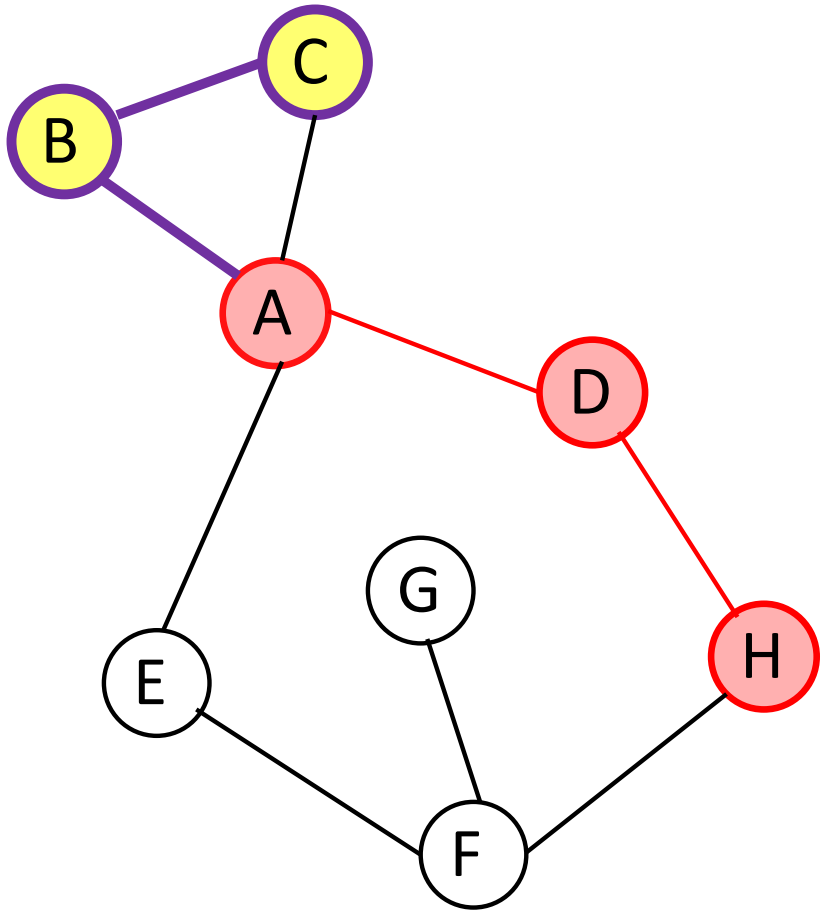



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

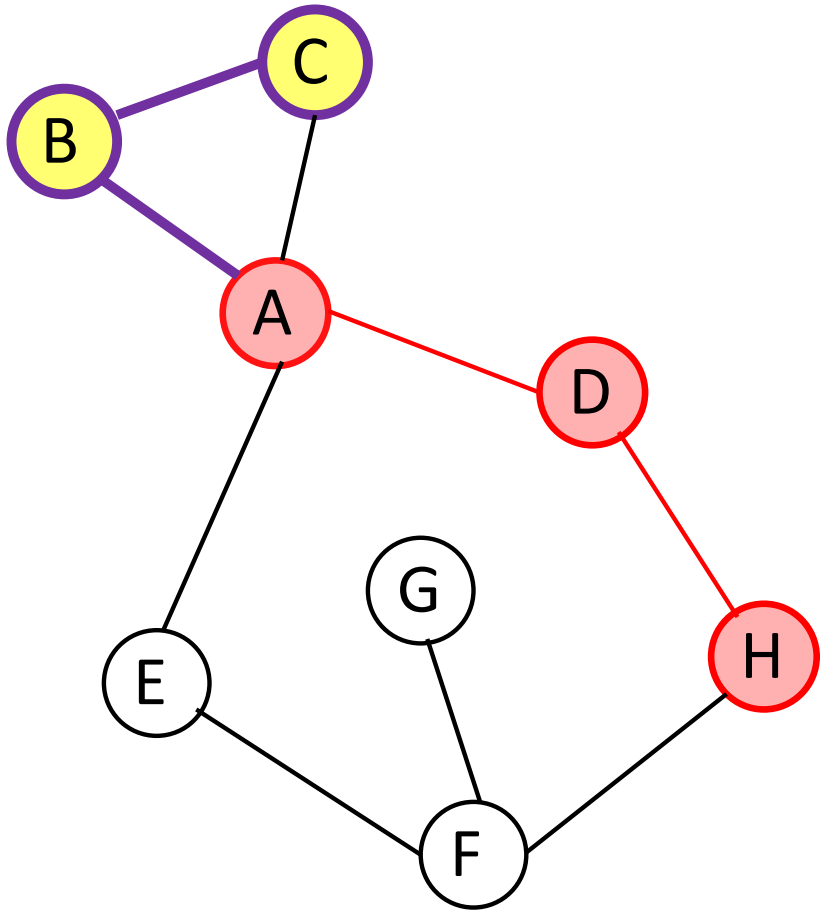



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

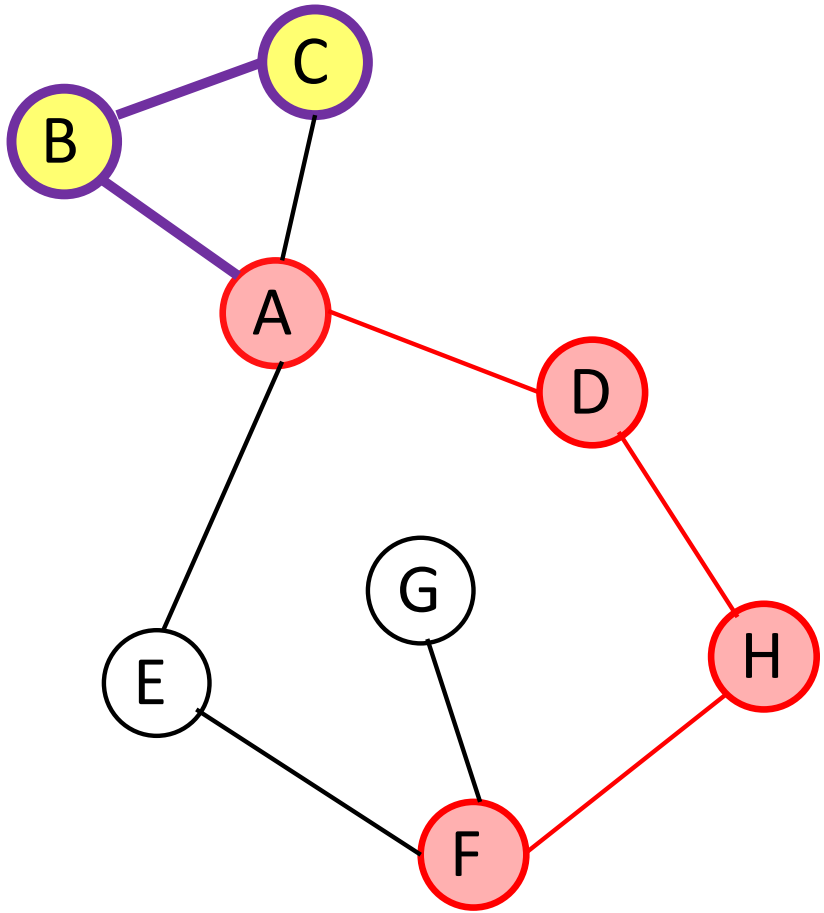



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

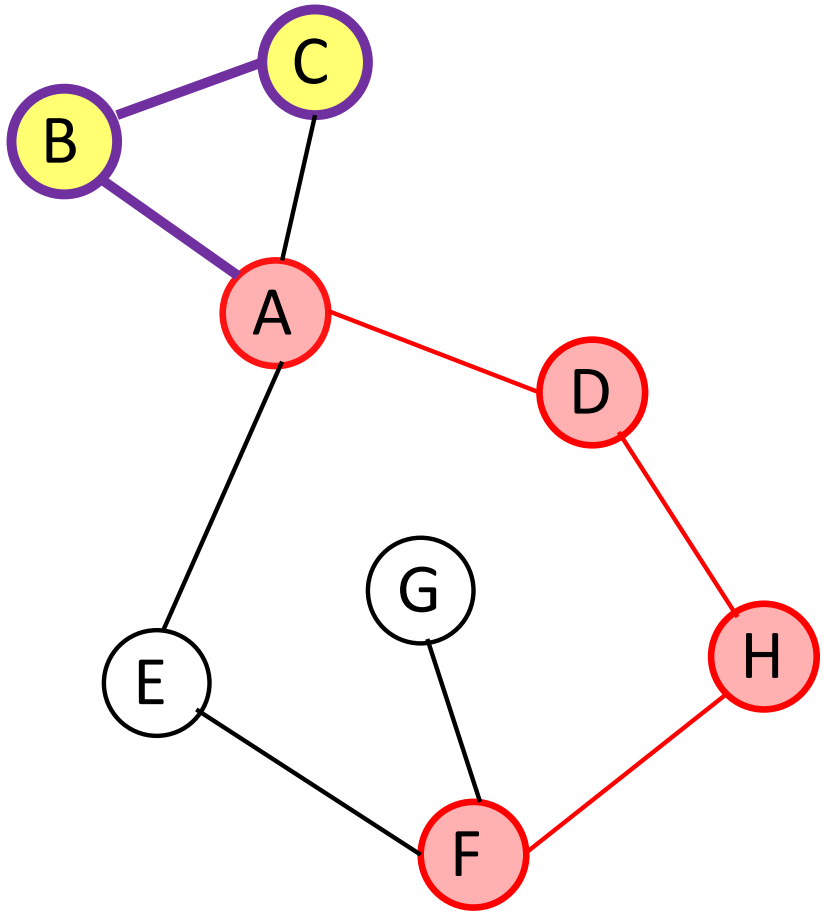



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
```

```
explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



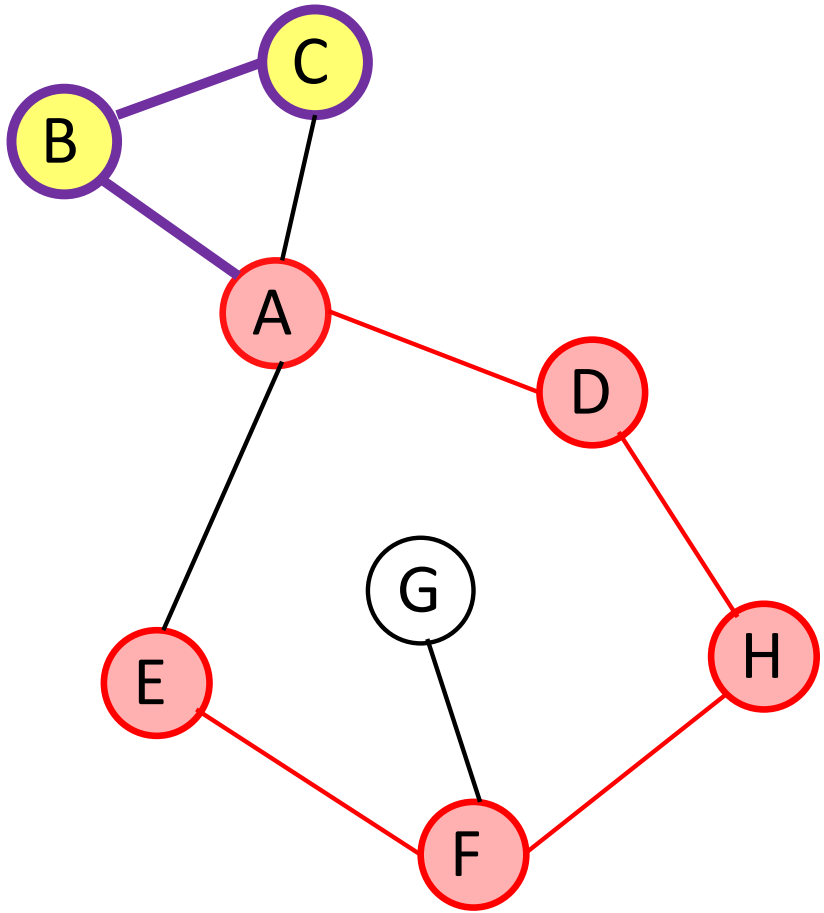

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)


        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



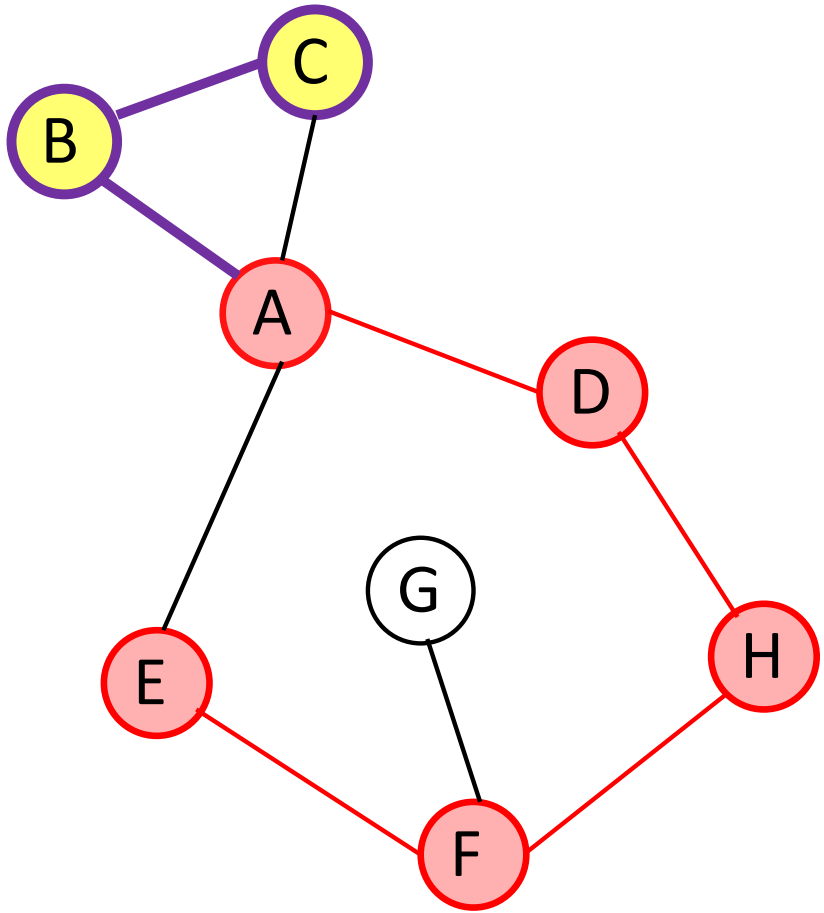

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)


        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

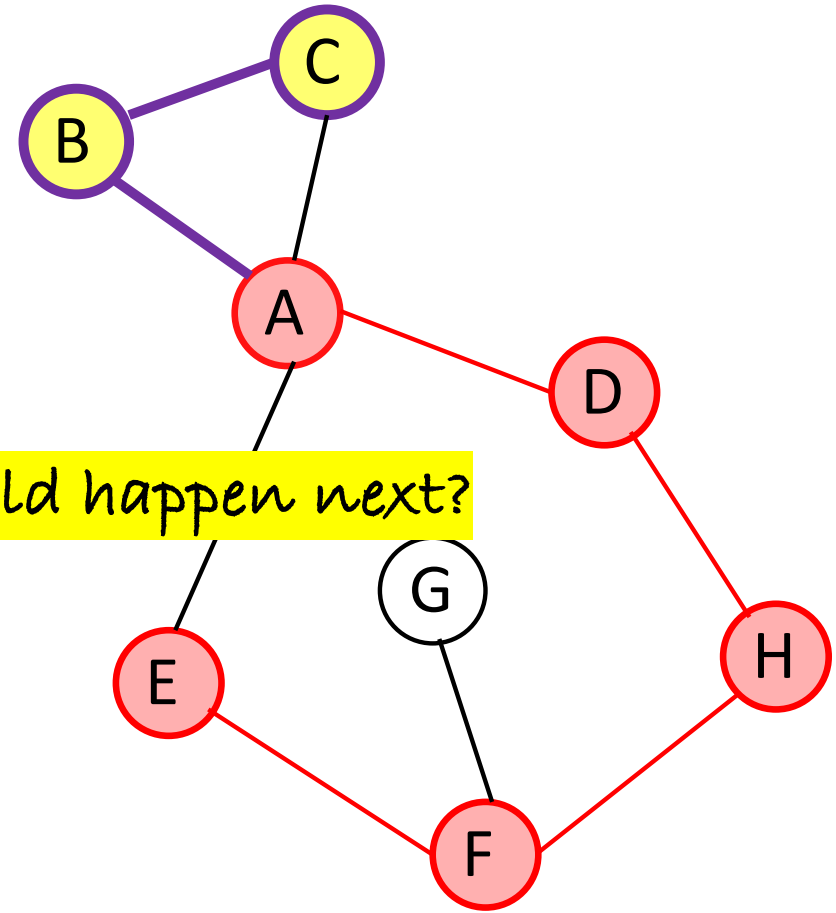



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



Dead end!

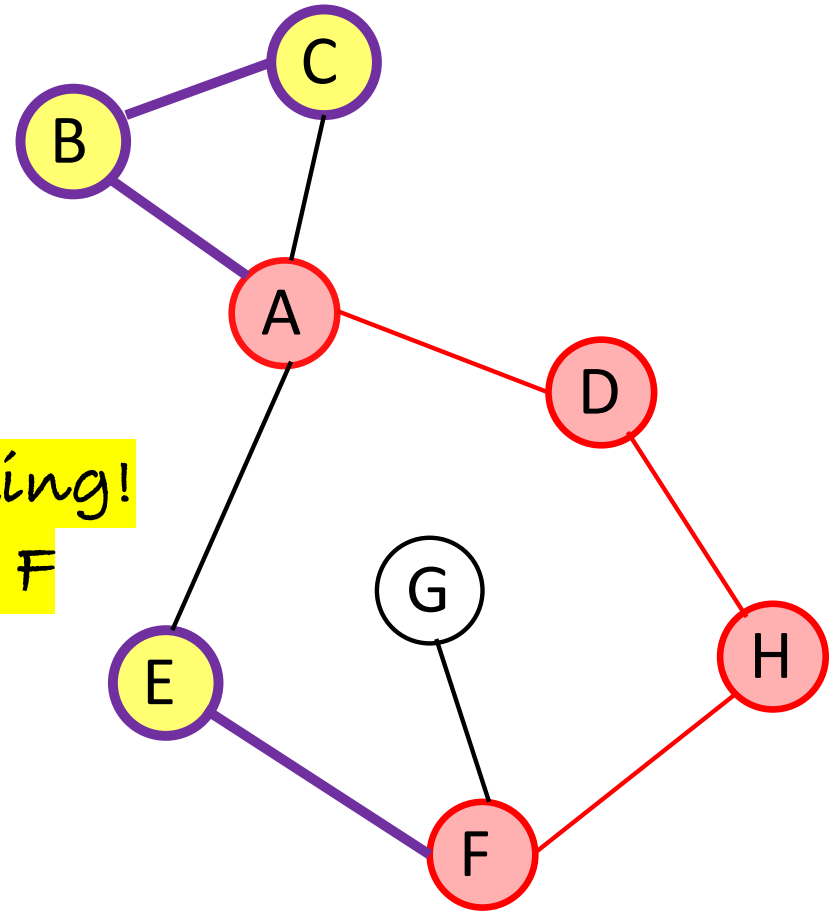
What should happen next?

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example



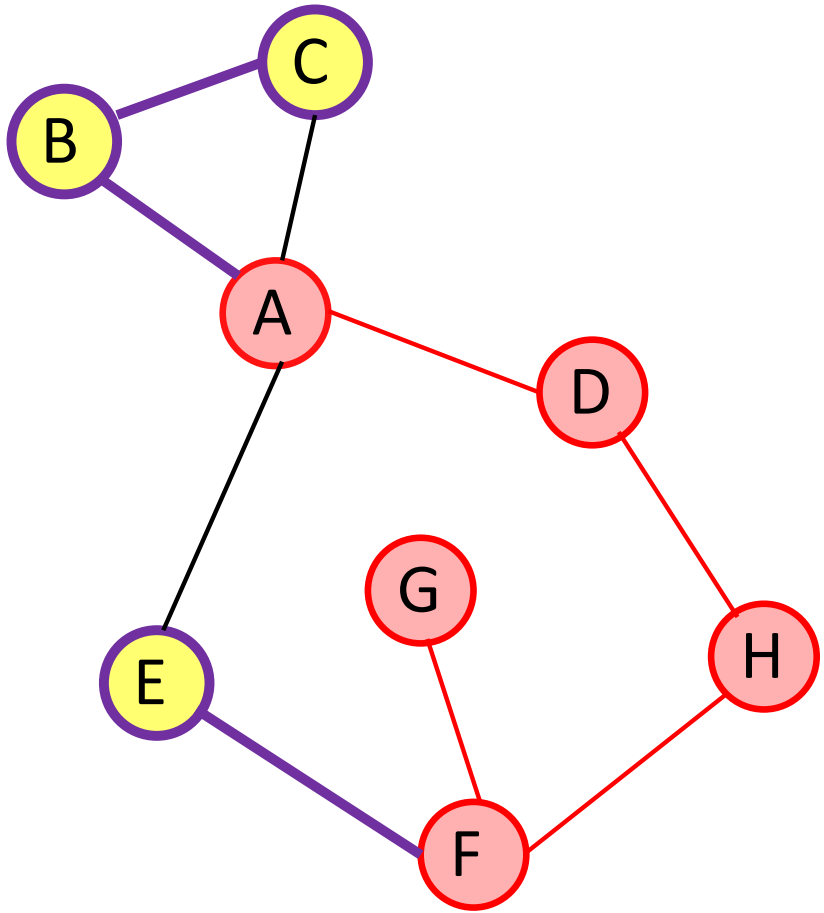
Backtracking!
Go back to F

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

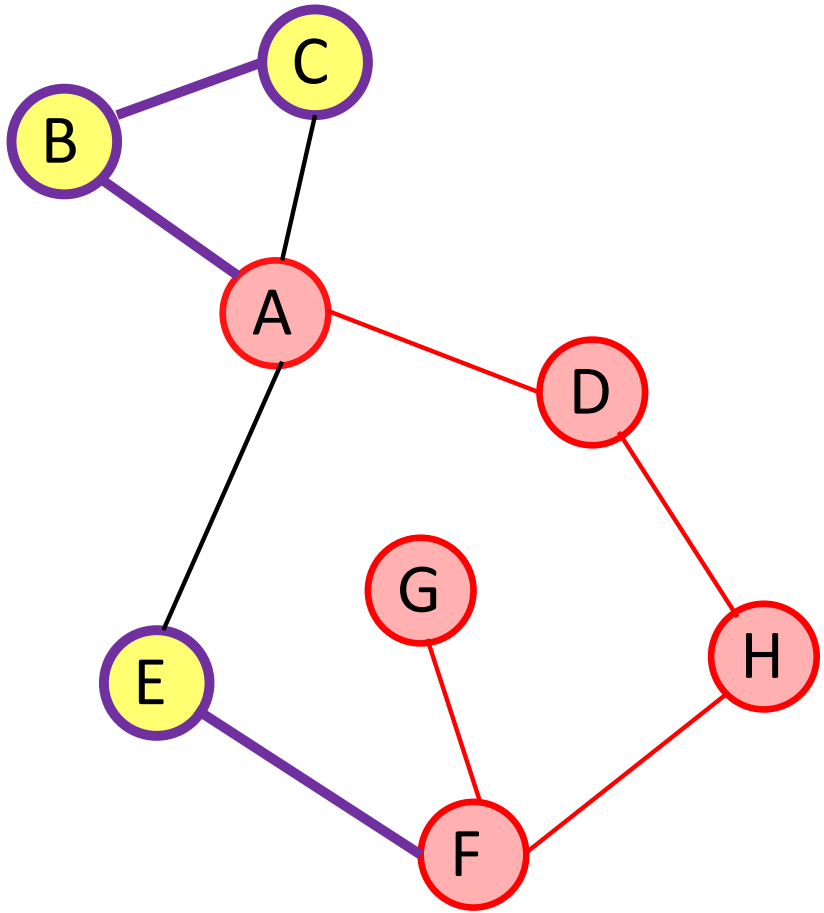



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)

        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

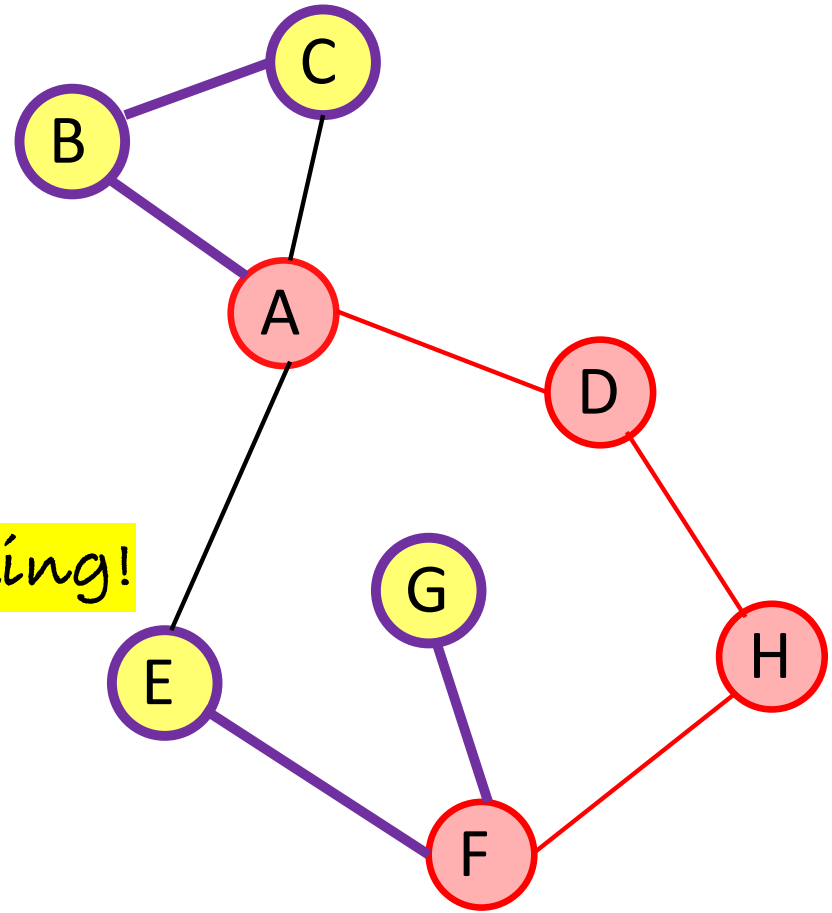
Explore (Depth First): Example




```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

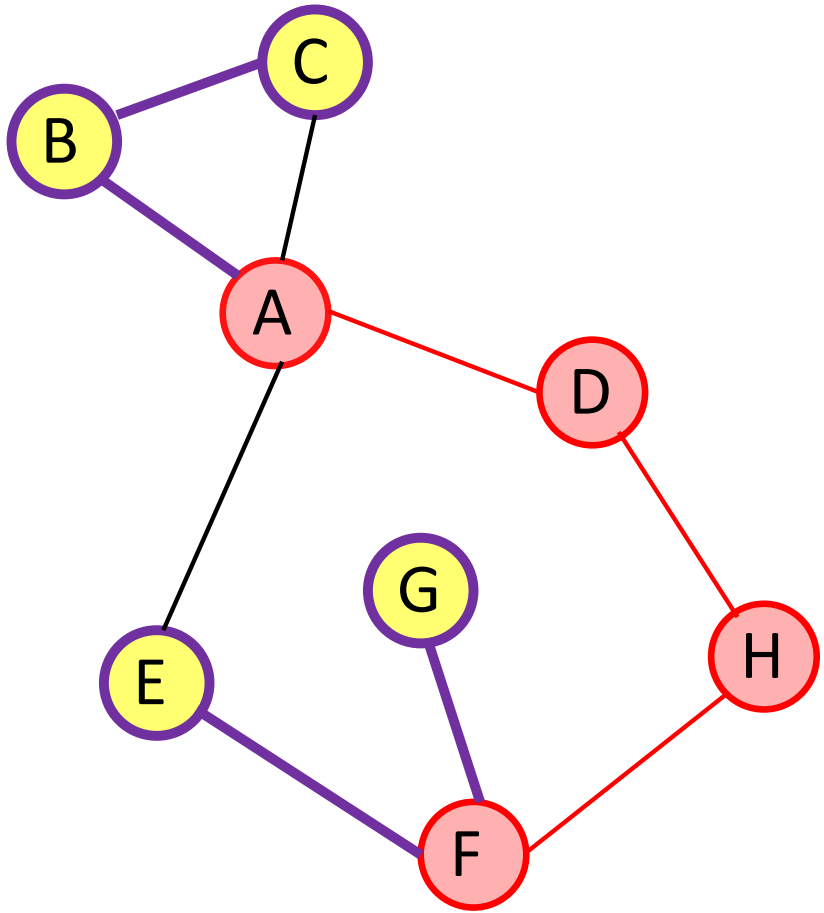


Backtracking!

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

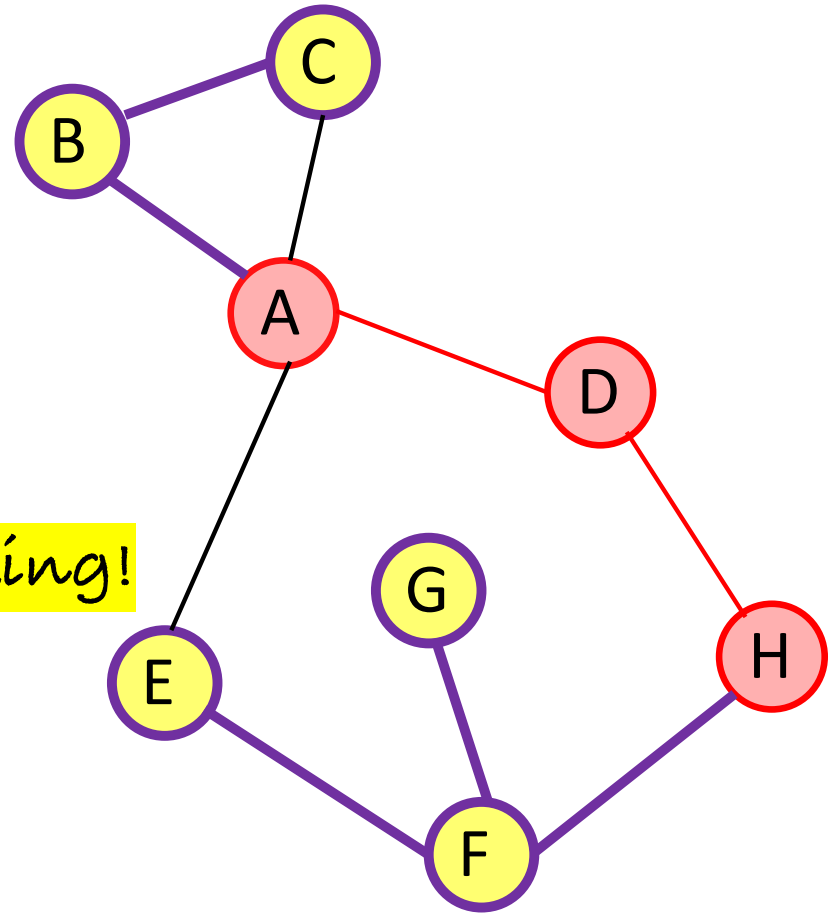
Explore (Depth First): Example




```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

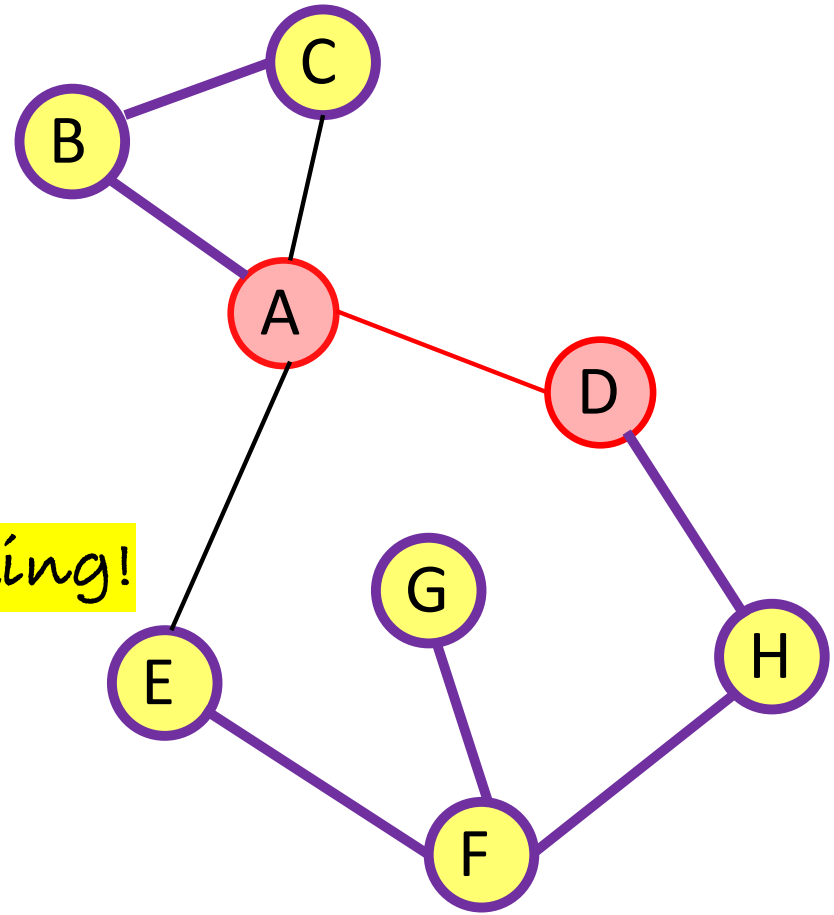


Backtracking!

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

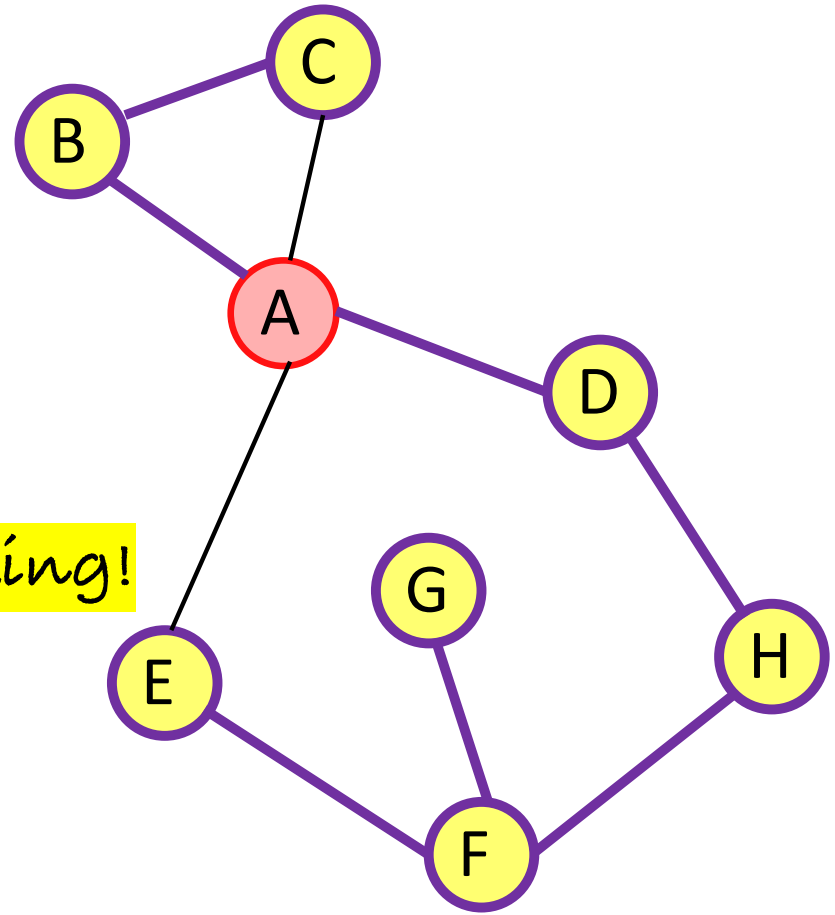


Backtracking!

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

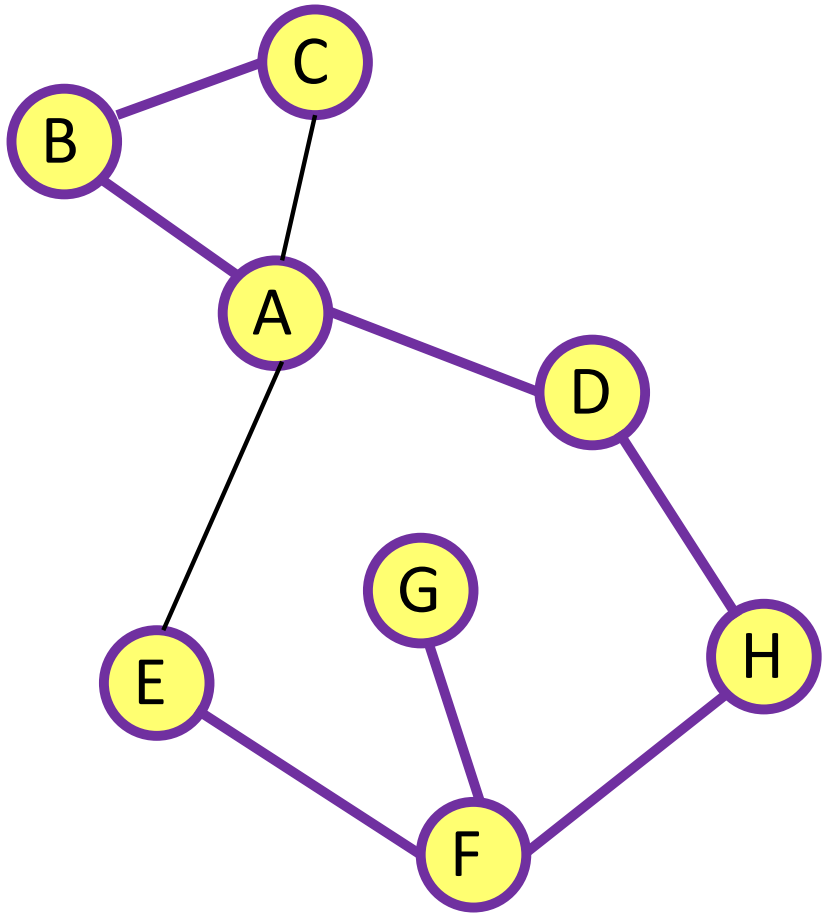


Backtracking!

```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First): Example

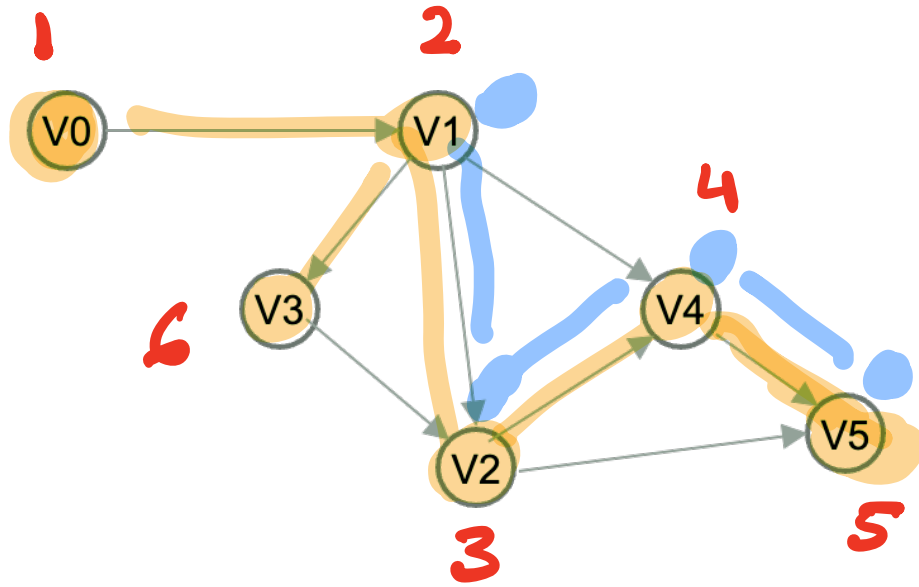



```
explore(A)
  explore(B)
    explore(A)
    explore(C)
      explore(A)
      explore(B)
  explore(C)
  explore(D)
    explore(A)
    explore(H)
      explore(D)
      explore(F)
        explore(E)
          explore(A)
          explore(F)
        explore(G)
          explore(F)
        explore(H)
  explore(E)
```

Based on slides by Professor Daniel Kane, UCSD

Explore (Depth First)

Search as far down a single path as possible, backtrack as needed



Assuming `exploreDFS` chooses the lower number node to explore first, in what order does `exploreDFS` visit the nodes in this graph?

A. V0, V1, V2, V3, V4, V5
B. V0, V1, V3, V4, V2, V5
C. V0, V1, V3, V2, V4, V5
D. V0, V1, V2, V4, V5, V3

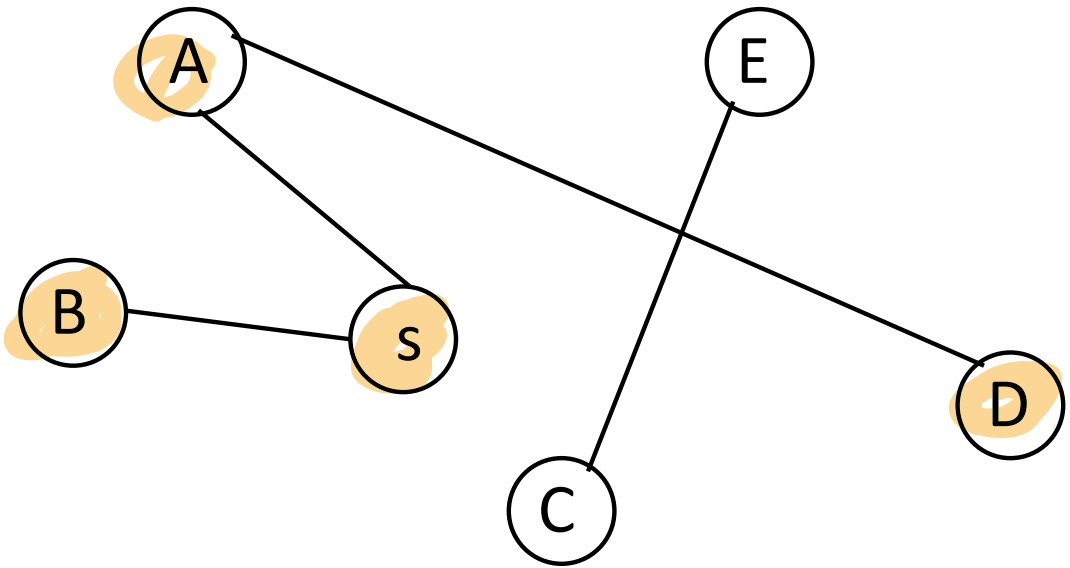
Question: exploreDFS

Which vertices does exploreDFS(s) mark as visited?

A. All the vertices

B. All vertices except C & E

C. None of the above



Slide credits: Professor Daniel Kane, UCSD

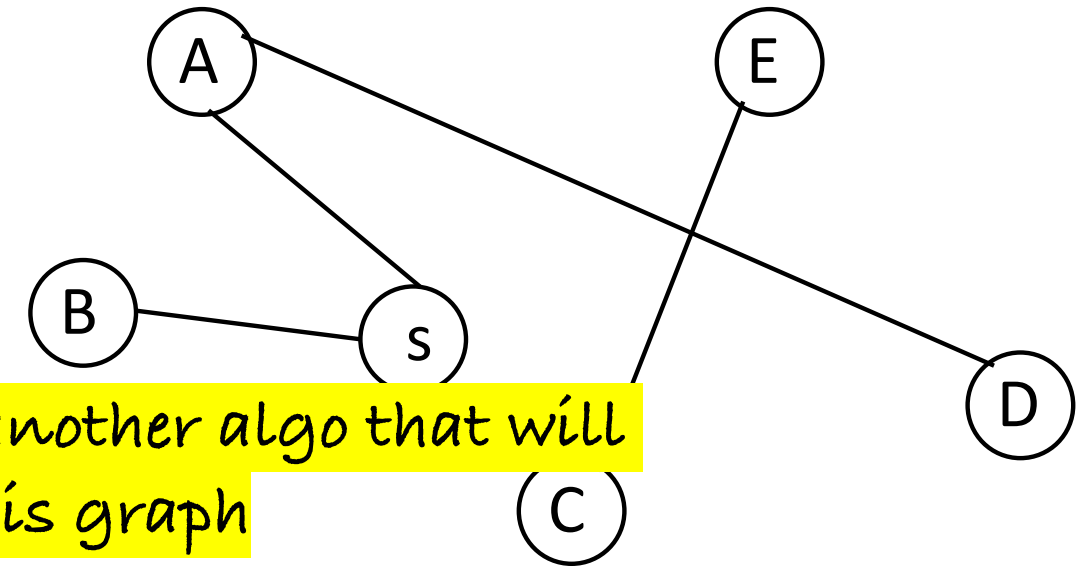
Question: exploreDFS

Which vertices does exploreDFS(s) mark as visited?

A. All the vertices

B. All vertices except C & E

C. None of the above



Use exploreDFS to write another algo that will visit all the vertices in this graph

Slide credits: Professor Daniel Kane, UCSD

Depth First Search

`exploreDFS` only finds the part of the graph reachable from a single vertex. If you want to discover the entire graph, you may need to run it multiple times.

```
DepthFirstSearch(G)
  Mark all v ∈ G as unvisited
  For v ∈ G
    If not v.visited, exploreDFS(v)
```

There are n rooms labeled from 0 to n - 1 and all the rooms are locked except for room 0. Your goal is to visit all the rooms. However, you cannot enter a locked room without having its key.

When you visit a room, you may find a set of distinct keys in it. Each key has a number on it, denoting which room it unlocks, and you can take all of them with you to unlock the other rooms.

Given an array rooms where rooms[i] is the set of keys that you can obtain if you visited room i, return true if you can visit all the rooms, or false otherwise.

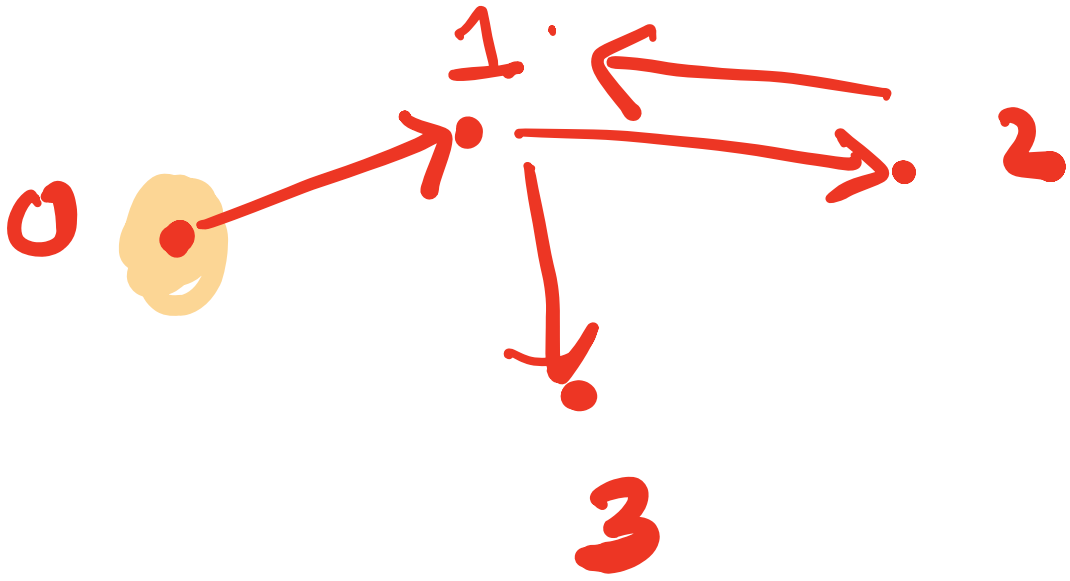
Can we solve this problem using depth-first search or exploreDFS?

A. Yes

B. No

https://leetcode.com/problems/keys-and-rooms/description/

Input: `rooms = [[1],[2, 3],[1],[]]`
Output: `true`



https://leetcode.com/problems/keys-and-rooms/description/

Complete Activity 4 in your handout

```
typedef std::vector<std::unordered_map<int, Connection> > AdjList;
```