Graph Traversals
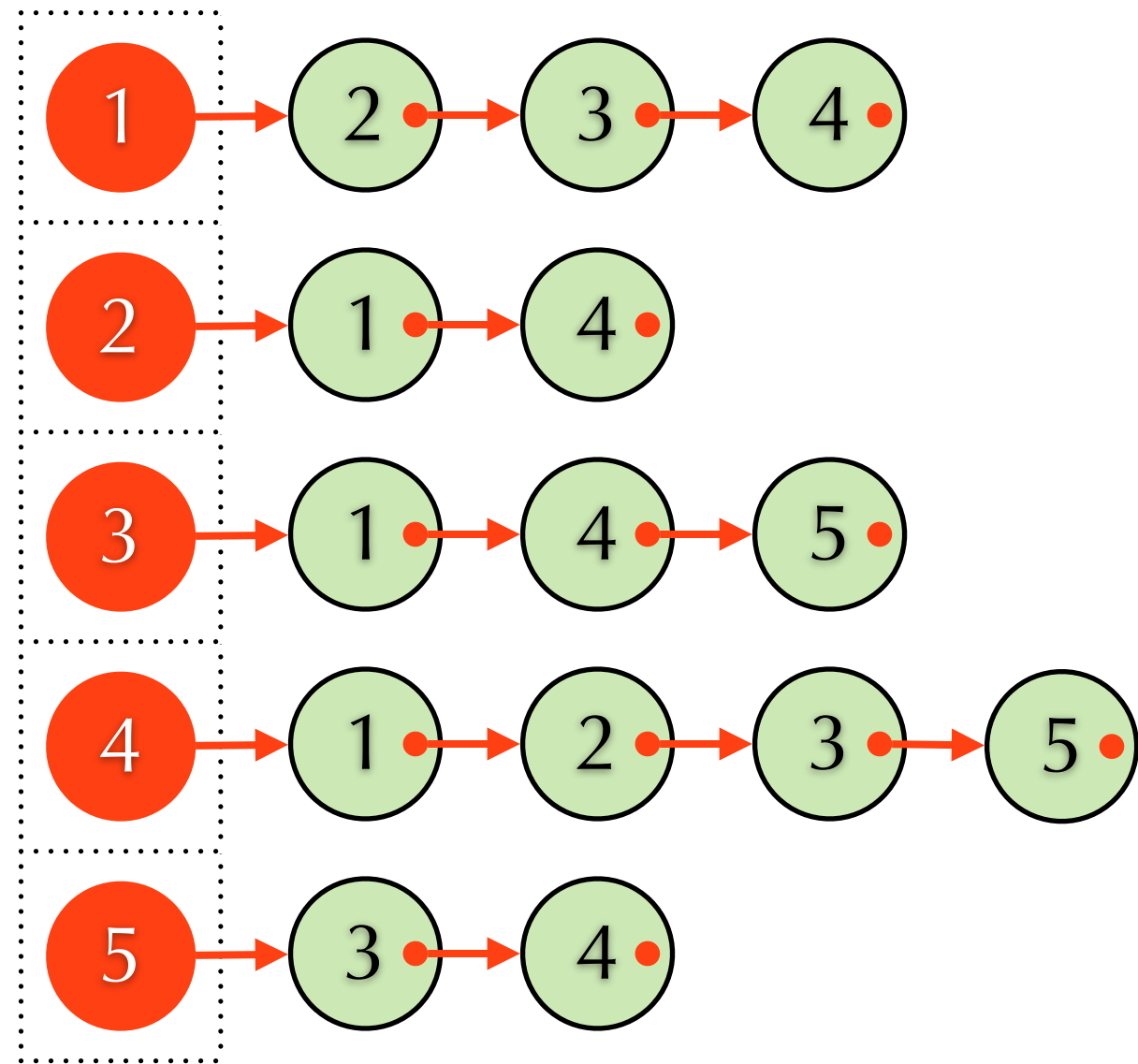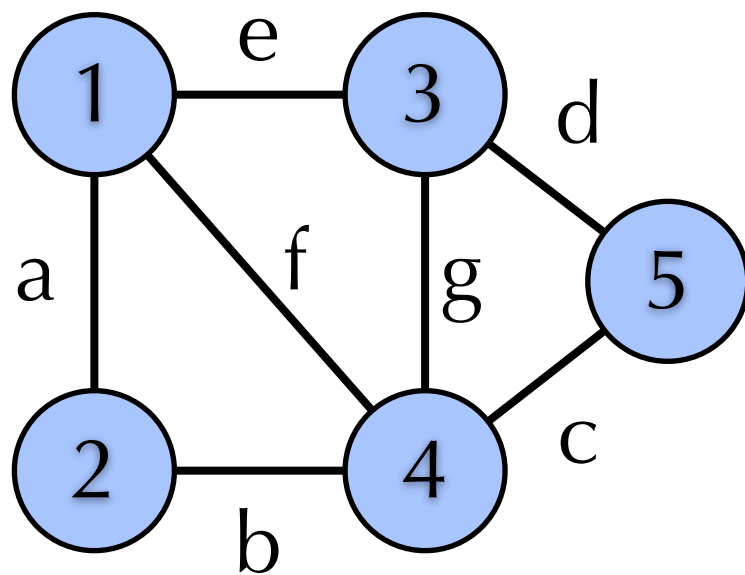
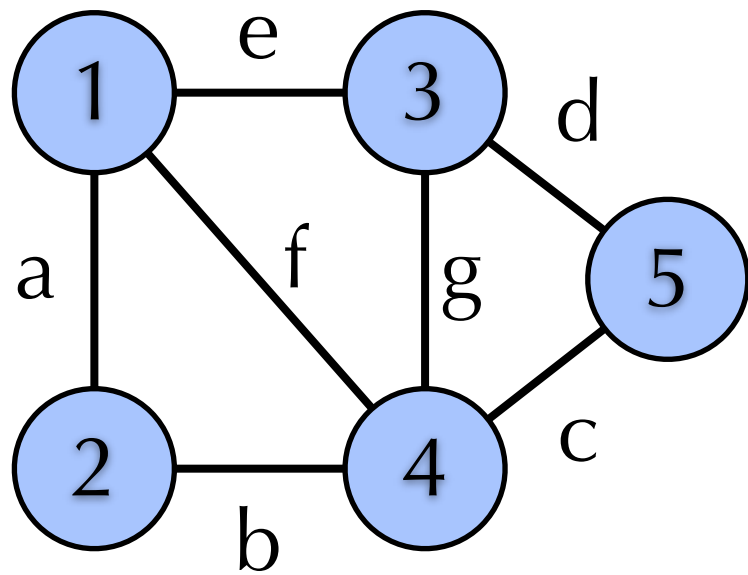# Graph Representation

‣ G=(V,E)

  ‣ V: set of vertices

  ‣ E: set of edges

    ‣ Directed edge: (u,v)

    ‣ Undirected edge: {u,v}

‣ Graph representations affect the complexity of algorithms

Adjacency List: Undirected
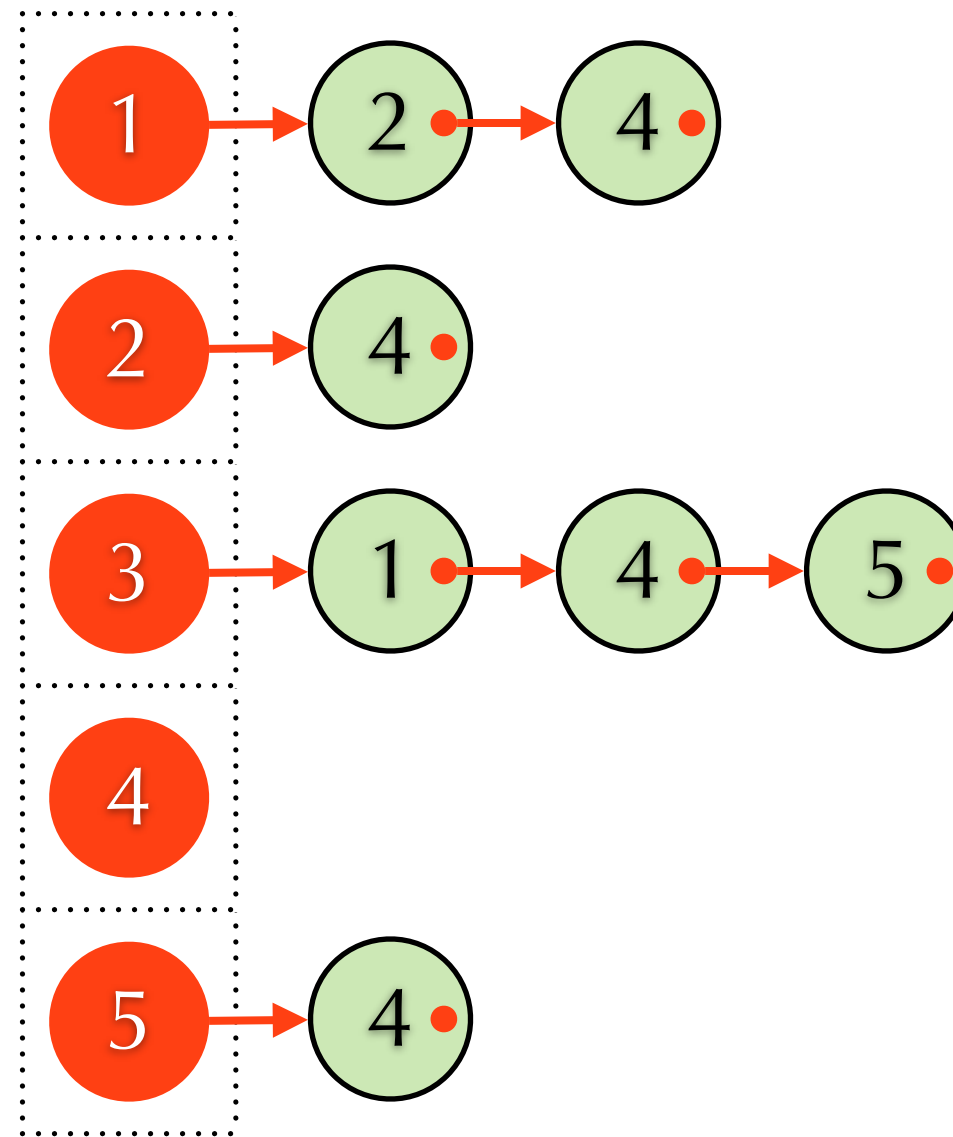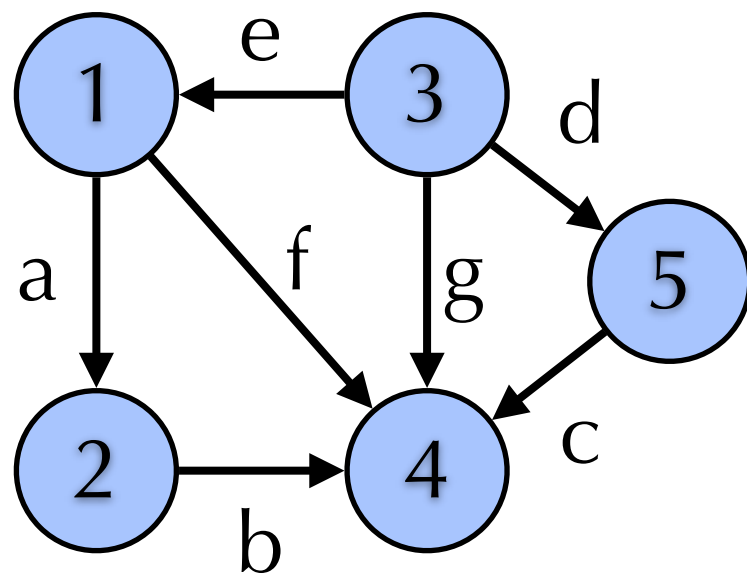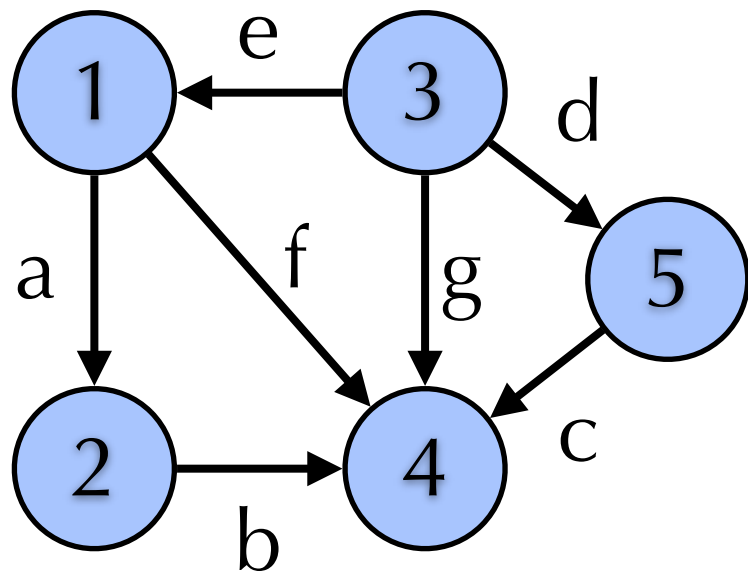
Adjacency Matrix: Undirected

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

Adjacency List: Directed

Adjacency Matrix: Directed

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

Graph Representation: Comparison

▸ Adjacency list
  ▸ Space: $\Theta(|V|+|E|)$
  ▸ Random access an edge: $O(d)$ d: degree
  ▸ Enumerate edges incident to a vertex: $O(d)$
▸ Adjacency matrix
  ▸ Space: $\Theta(|V|^2)$
  ▸ Random access an edge: $\Theta(1)$
  ▸ Enumerate edges incident to a vertex: $O(|V|)$

Graph Traversal

‣ Visit reachable nodes from a certain node s in a graph G=(V,E)

  ‣ Breadth-First-Search (BFS)

  ‣ Depth-First-Search (DFS)

‣ Application

  ‣ Finding the shortest paths in unweighted graphs or weighted DAGs

  ‣ Cycle detection

  ‣ Topological sort
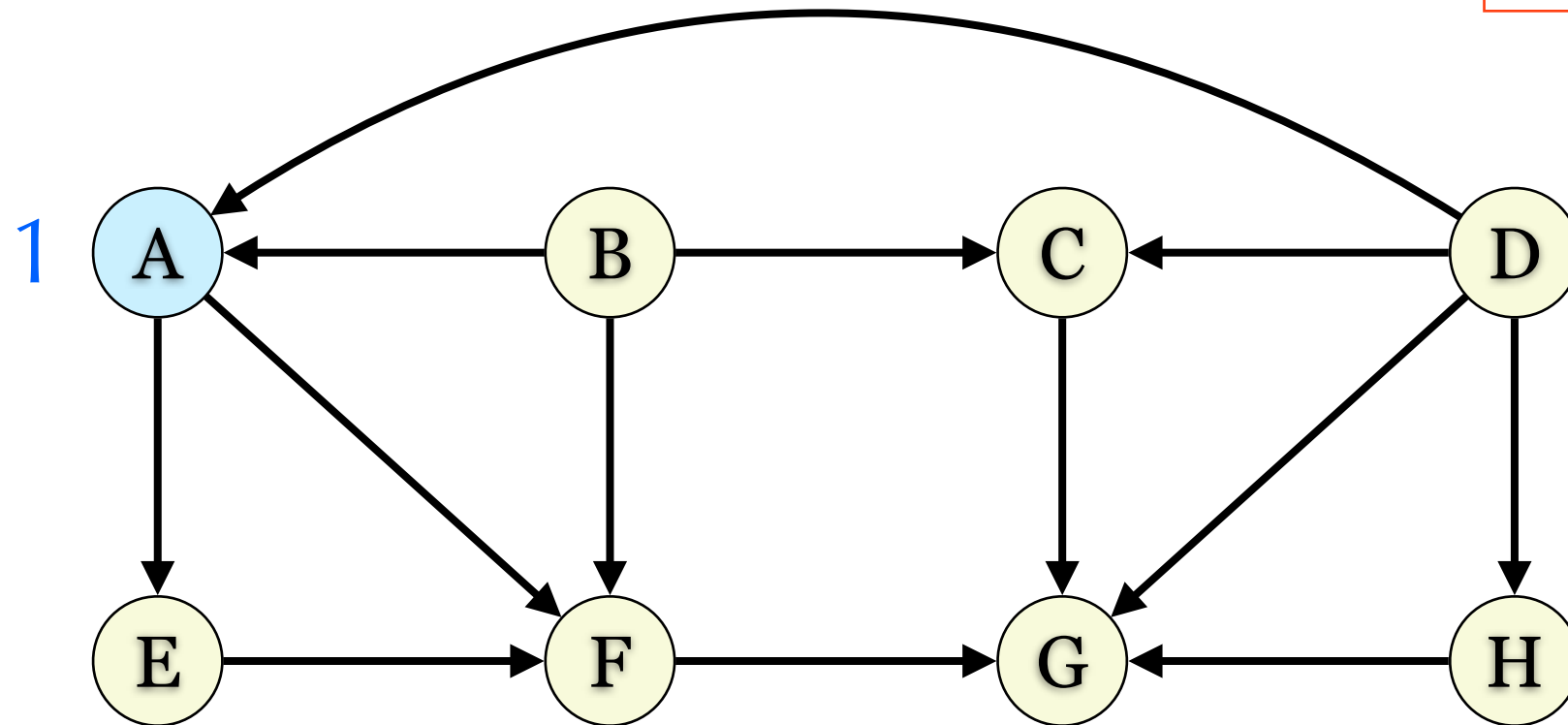
  ‣ Others

Breadth First Search

‣ Initialization:
  ‣ for each v∈V: enqueued[v]=visited[v]=false
  ‣ enqueued[s]=true, Q.enqueue(s)
‣ Main loop: while(Q≠∅)
  ‣ u=Q.dequeue()
    for each v s.t. (u,v)∈E
      if enqueued[v]==false
        enqueued[v]=true, Q.enqueue(v)
    visited[u]=true

Example: BFS

Queue | A | | | | | | |

Example: BFS

Queue: E F

Example: BFS

1 A ↔ B → C ← D

2 E → F → G ← H

3
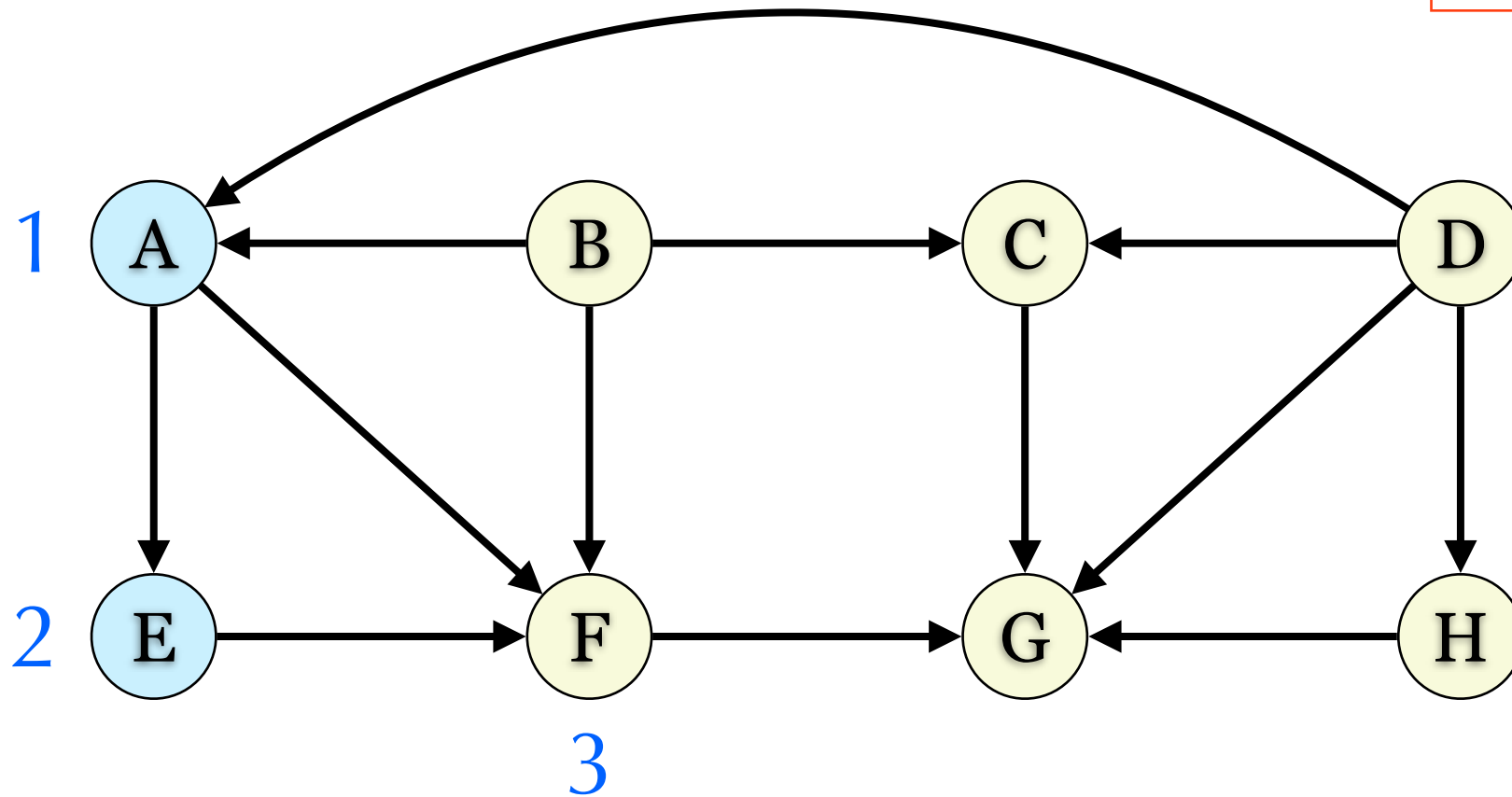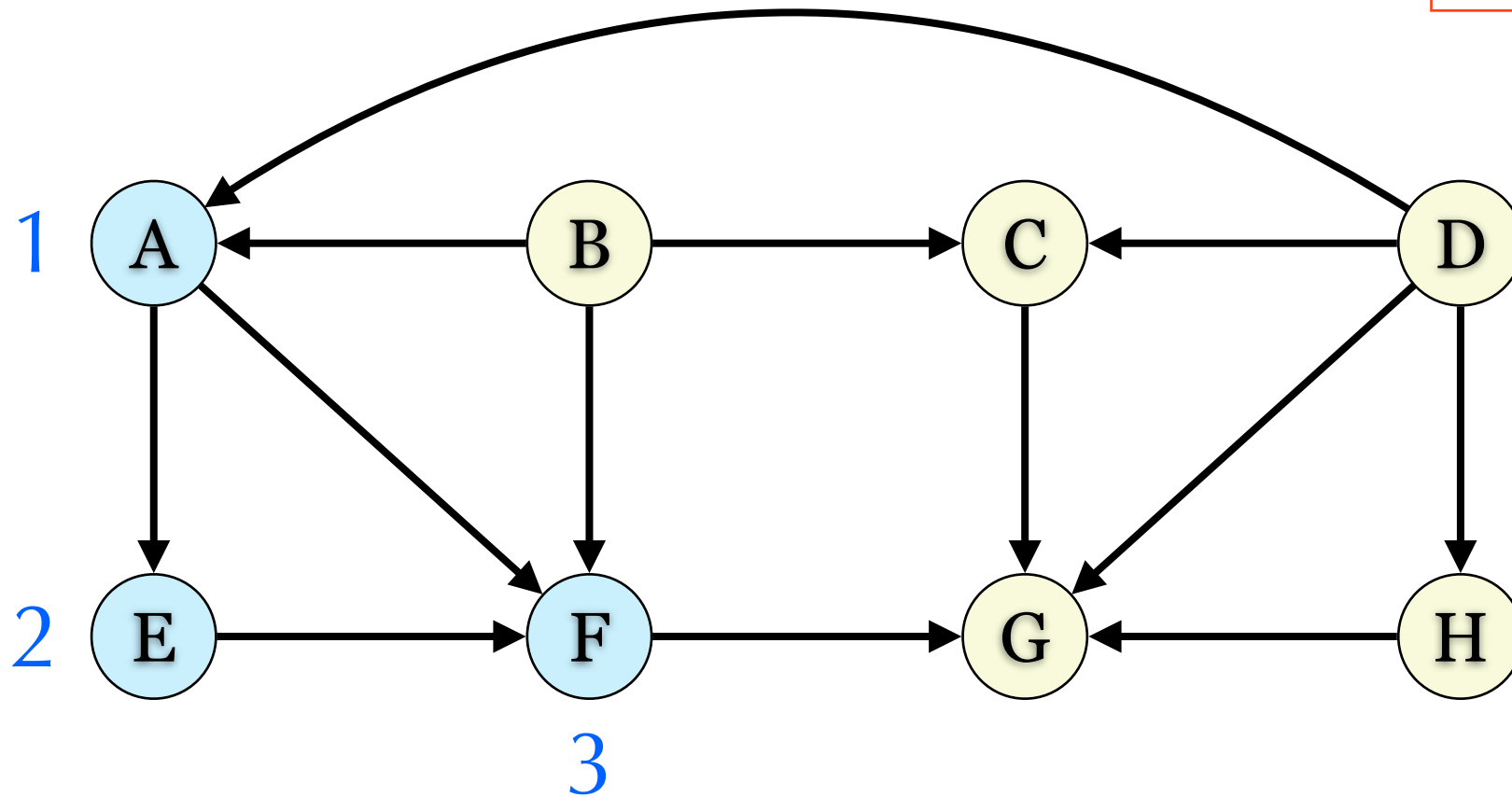
Queue | F | G | | | | | | |
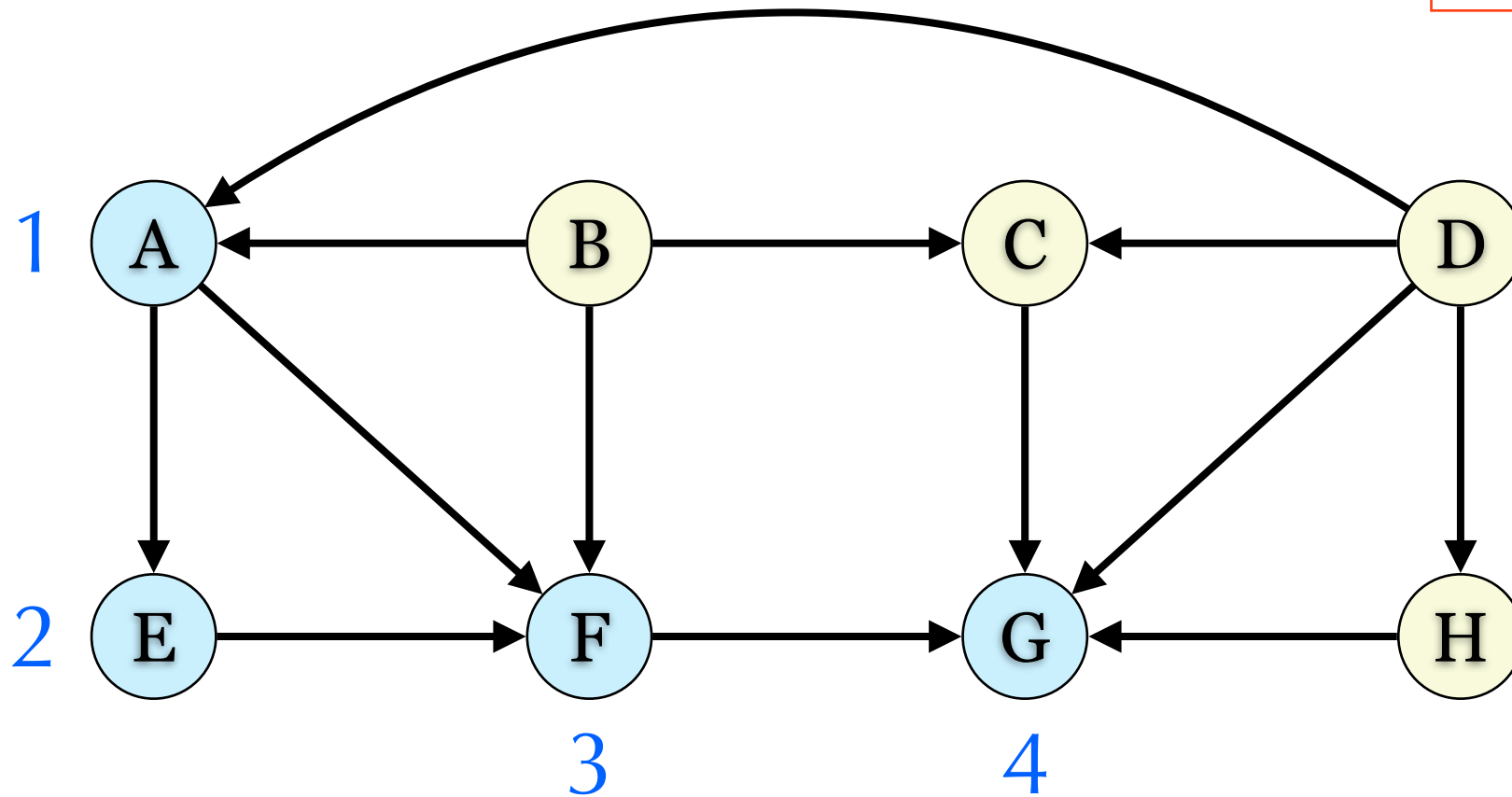
Example: BFS

Queue | G | | | | | | | |

Example: BFS
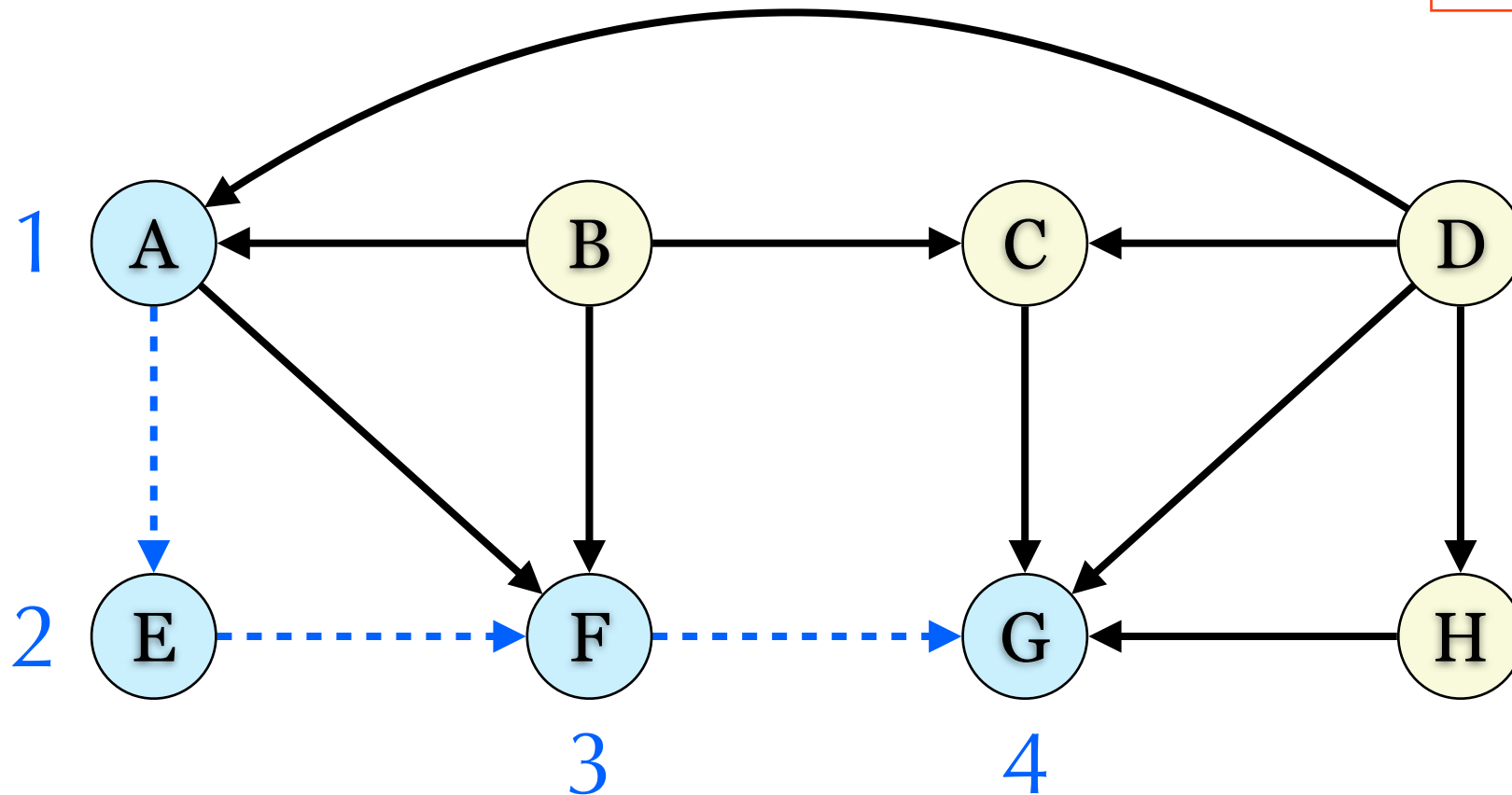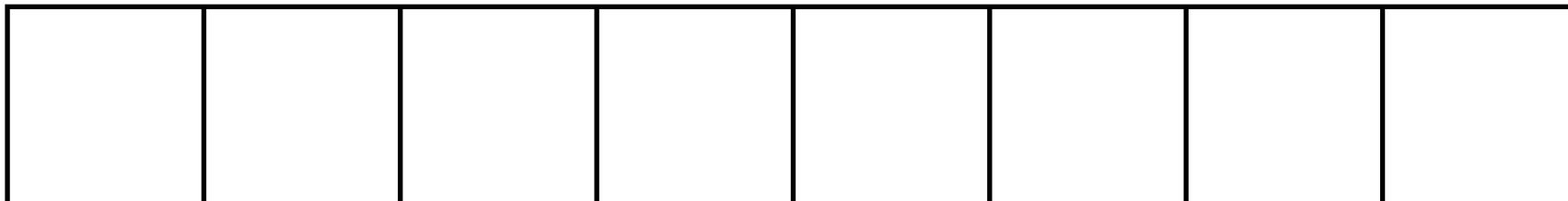
Queue

Done!

Example 2: BFS
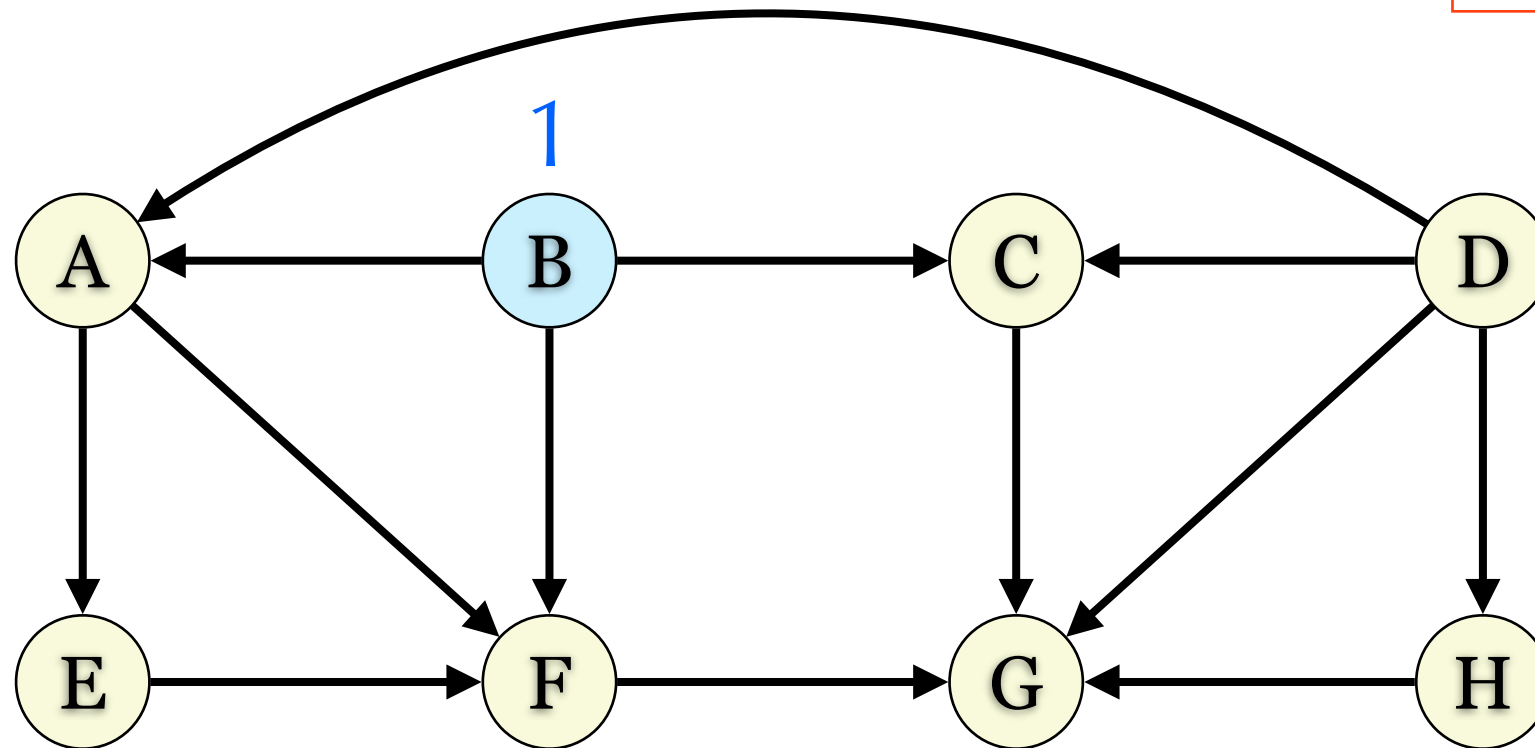
Queue: B

Example 2: BFS
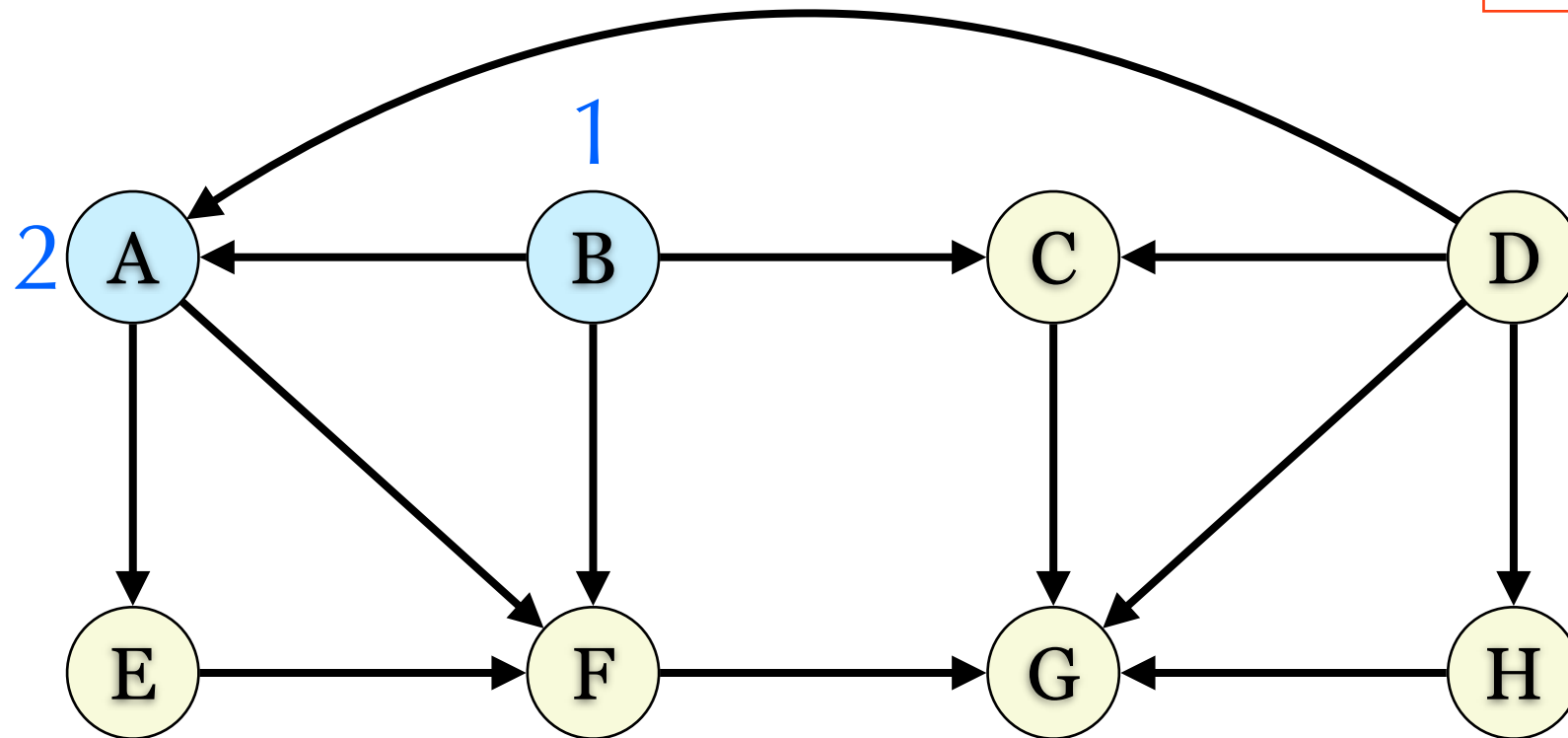
Queue | A | C | F | | | | | |

Example 2: BFS
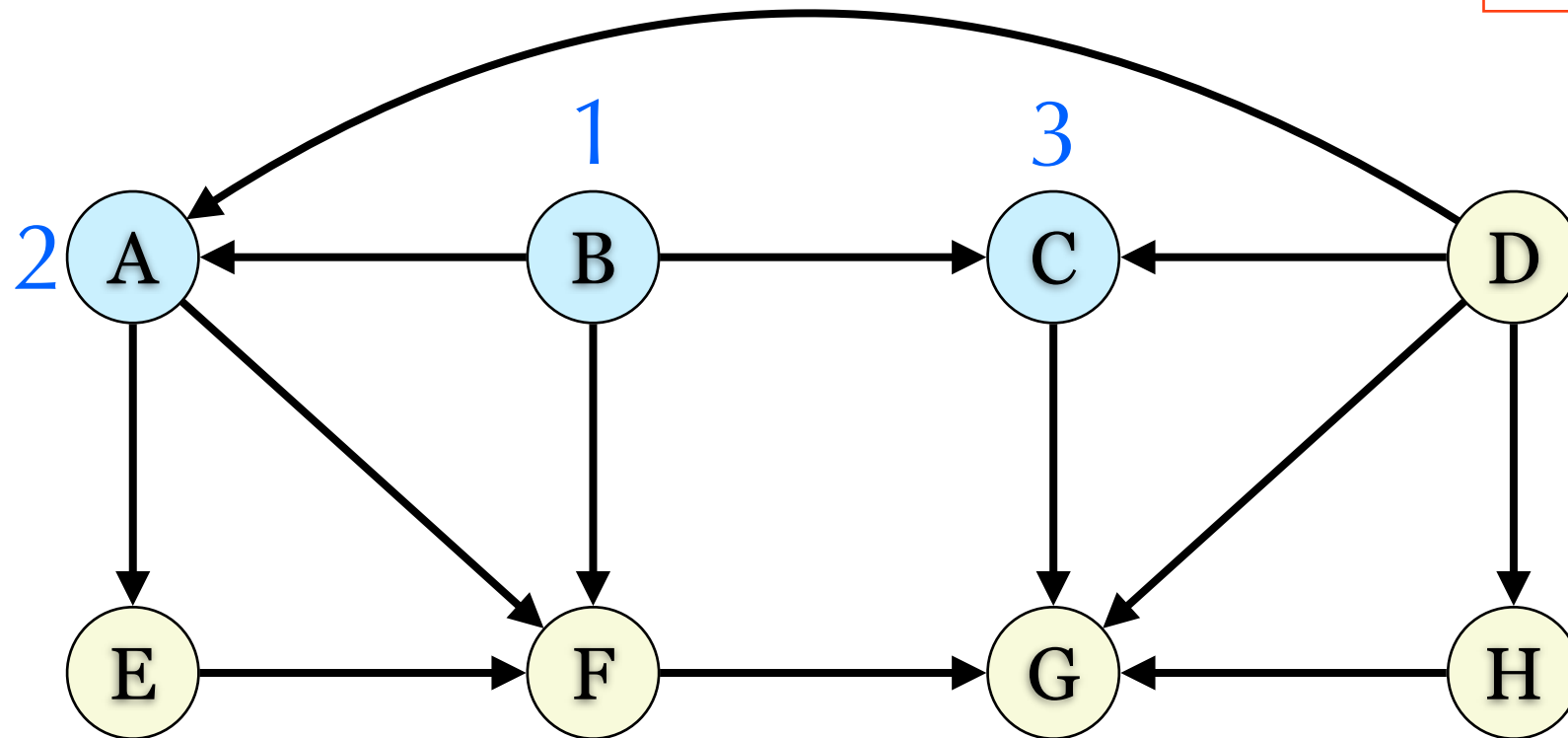
Queue

C	F	E					

Example 2: BFS
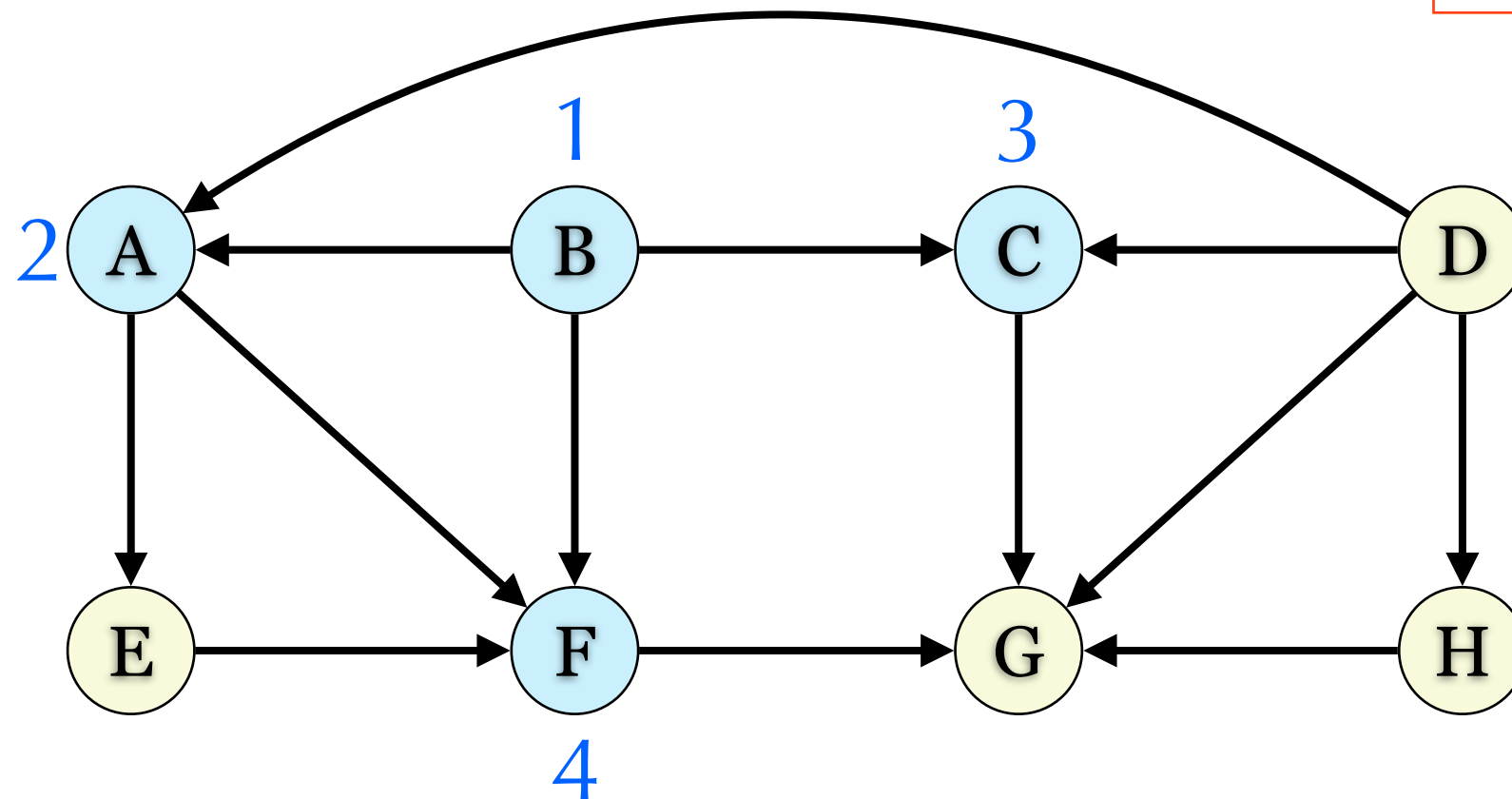
Queue

F	E	G					

Example 2: BFS

Queue

E	G						
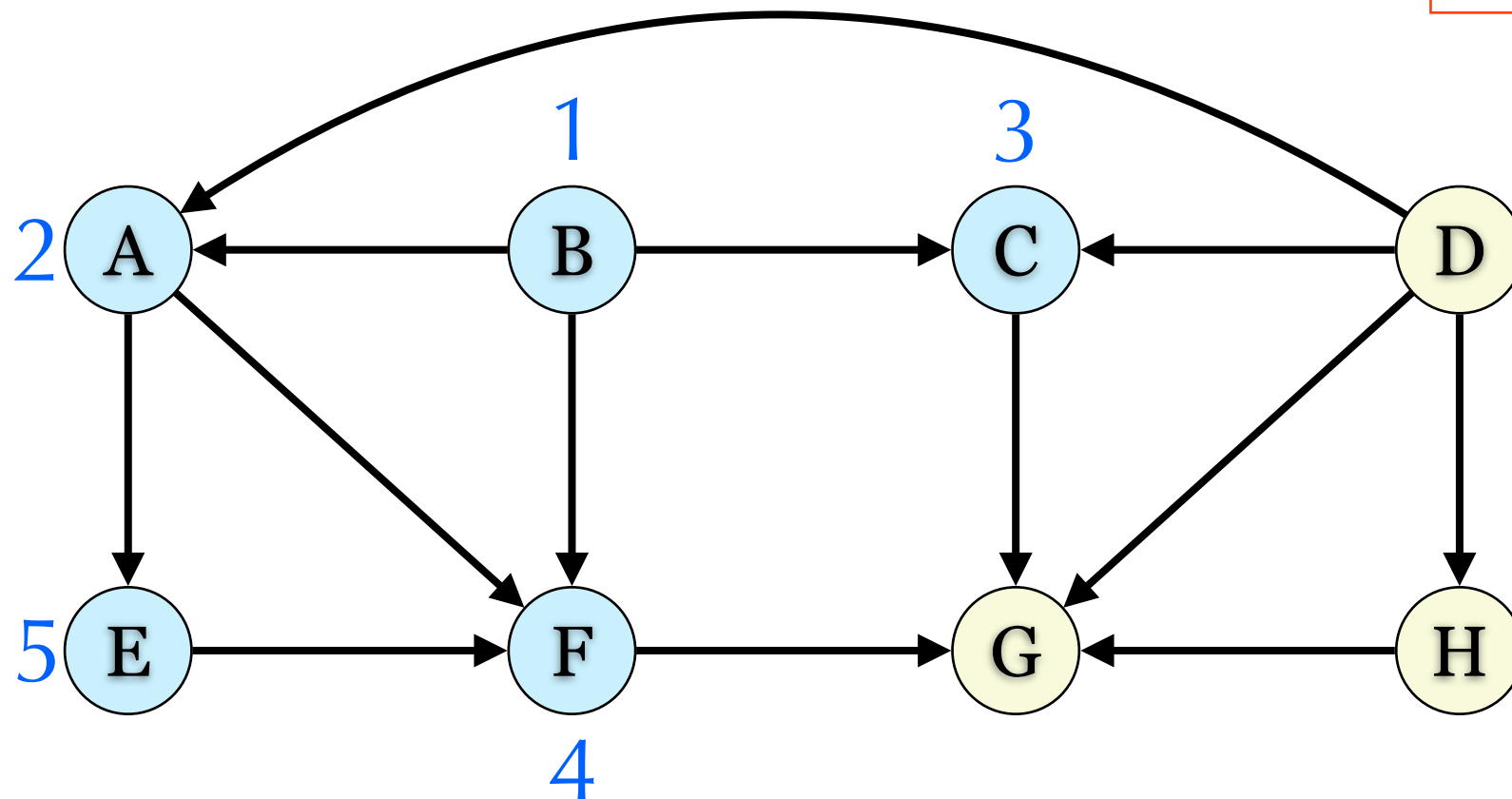
Example 2: BFS
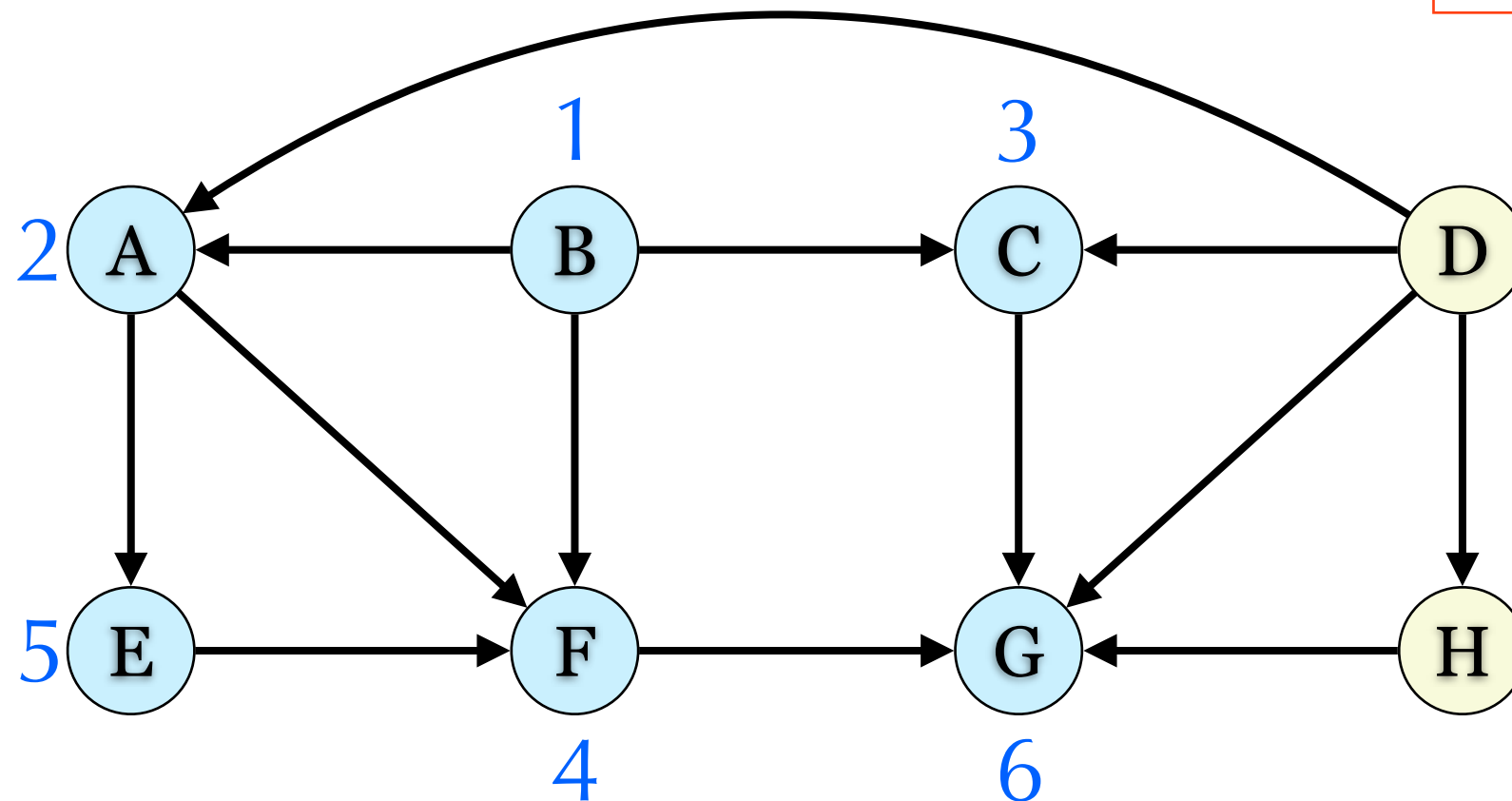
Queue | G | | | | | | | |

Example 2: BFS
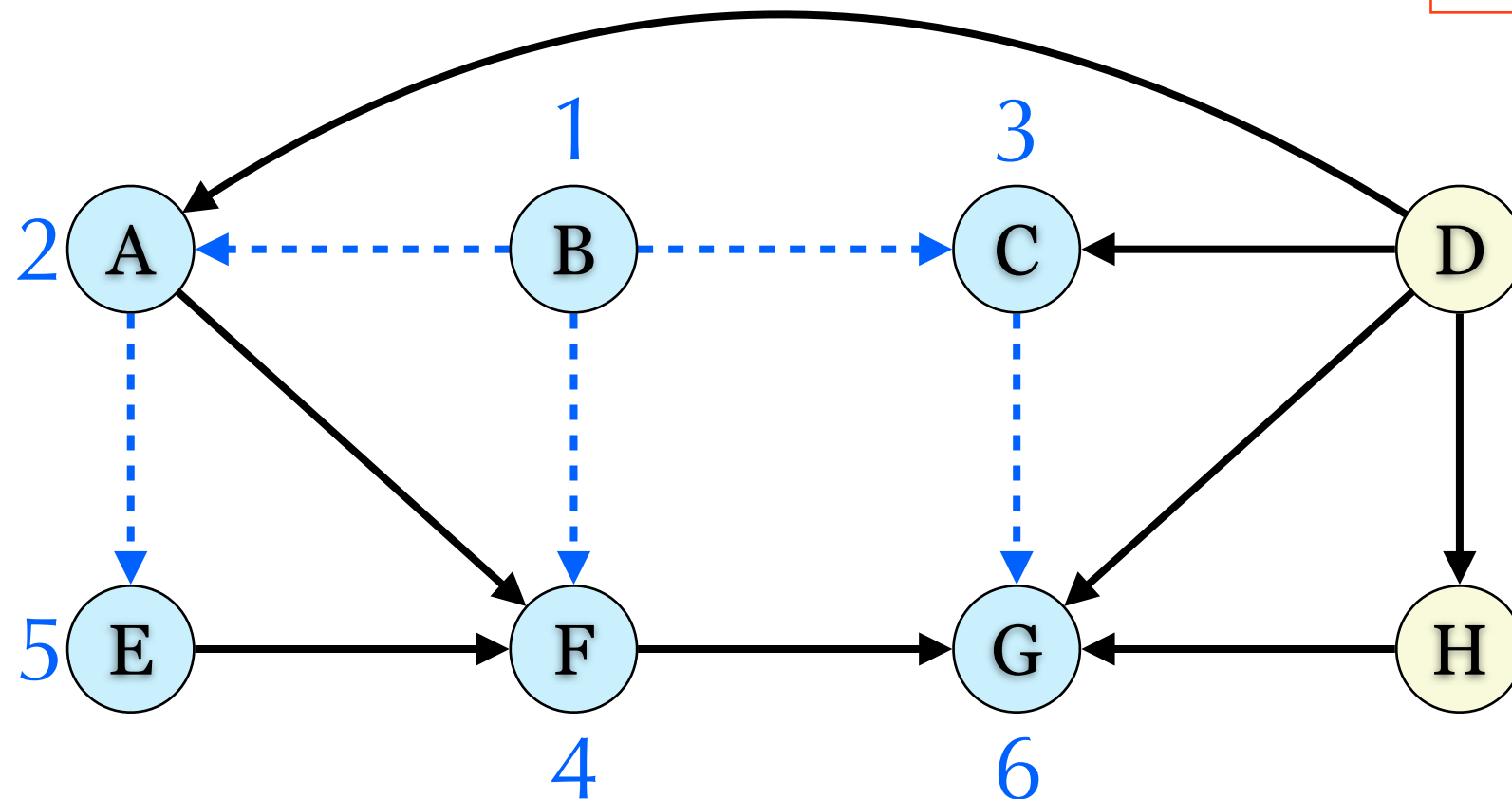
Queue

Done!

Depth First Search

▸ Initialization:
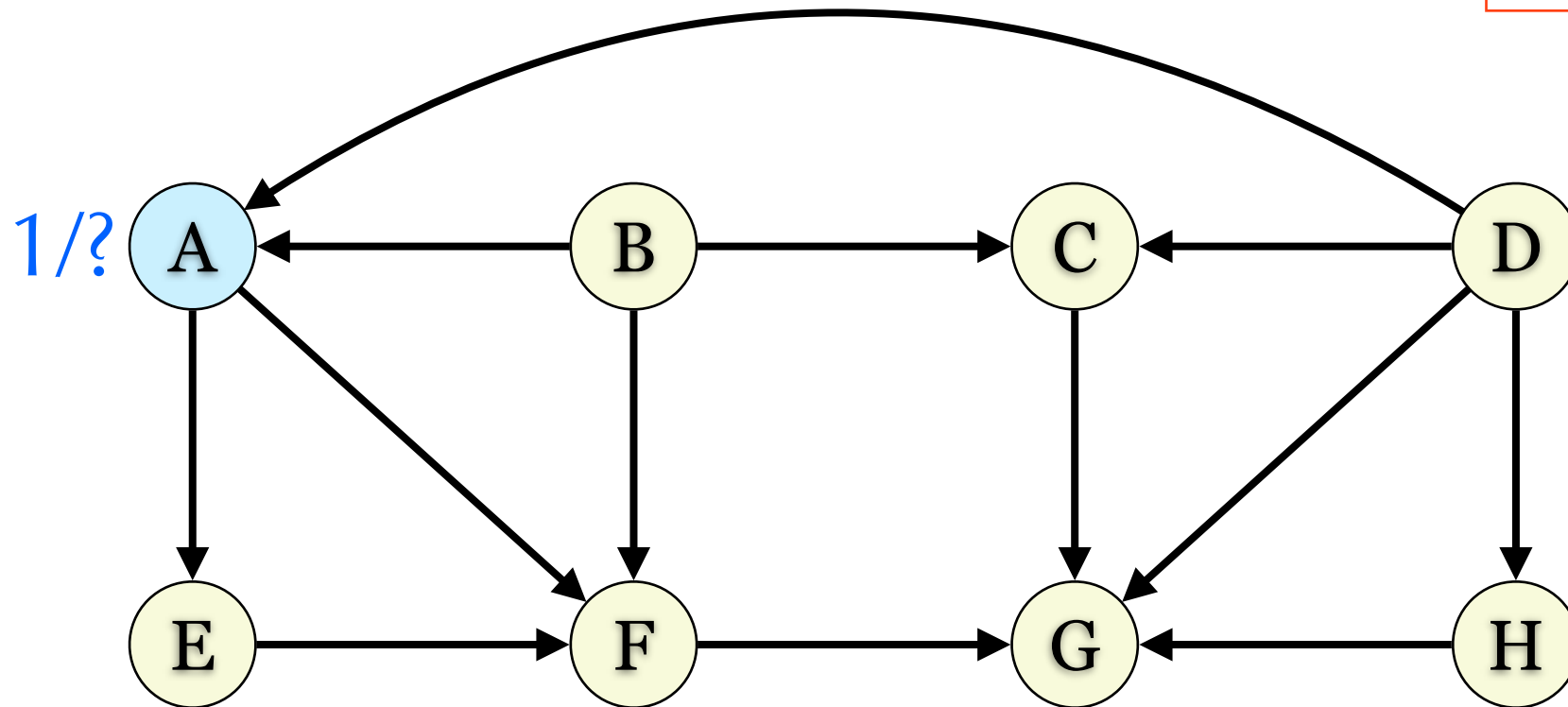
   ▸ for each v∈V: discovered[v]=visited[v]=false

▸ DFS-visit(u)

   ▸ discovered[u]=true
   for each v s.t. (u,v)∈E
       if discovered[v]==false
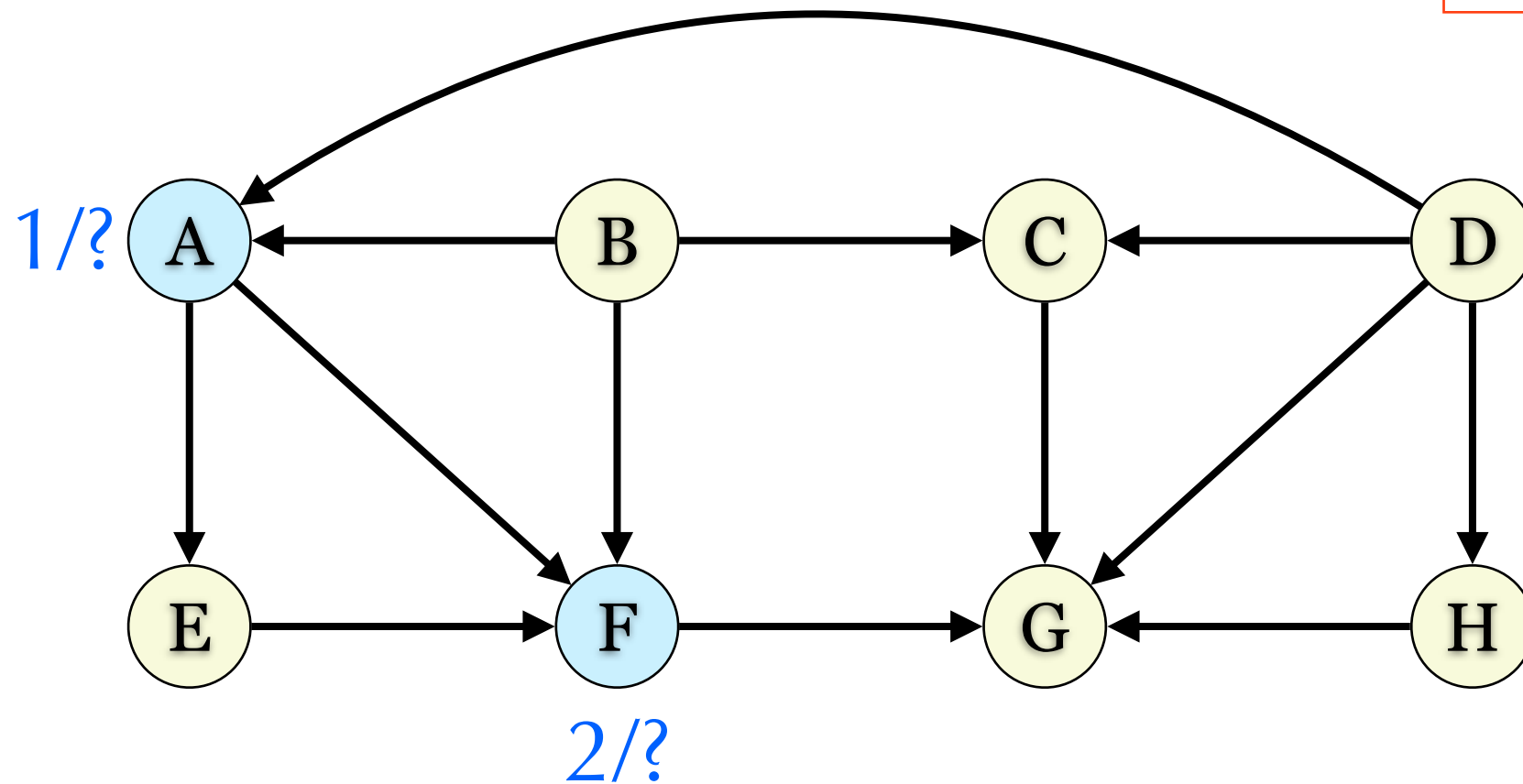   DFS-visit(v)
   visited[u]=true

Example: DFS

Stack

A/o							

Example: DFS

Stack

A/o	F/o						

Example: DFS
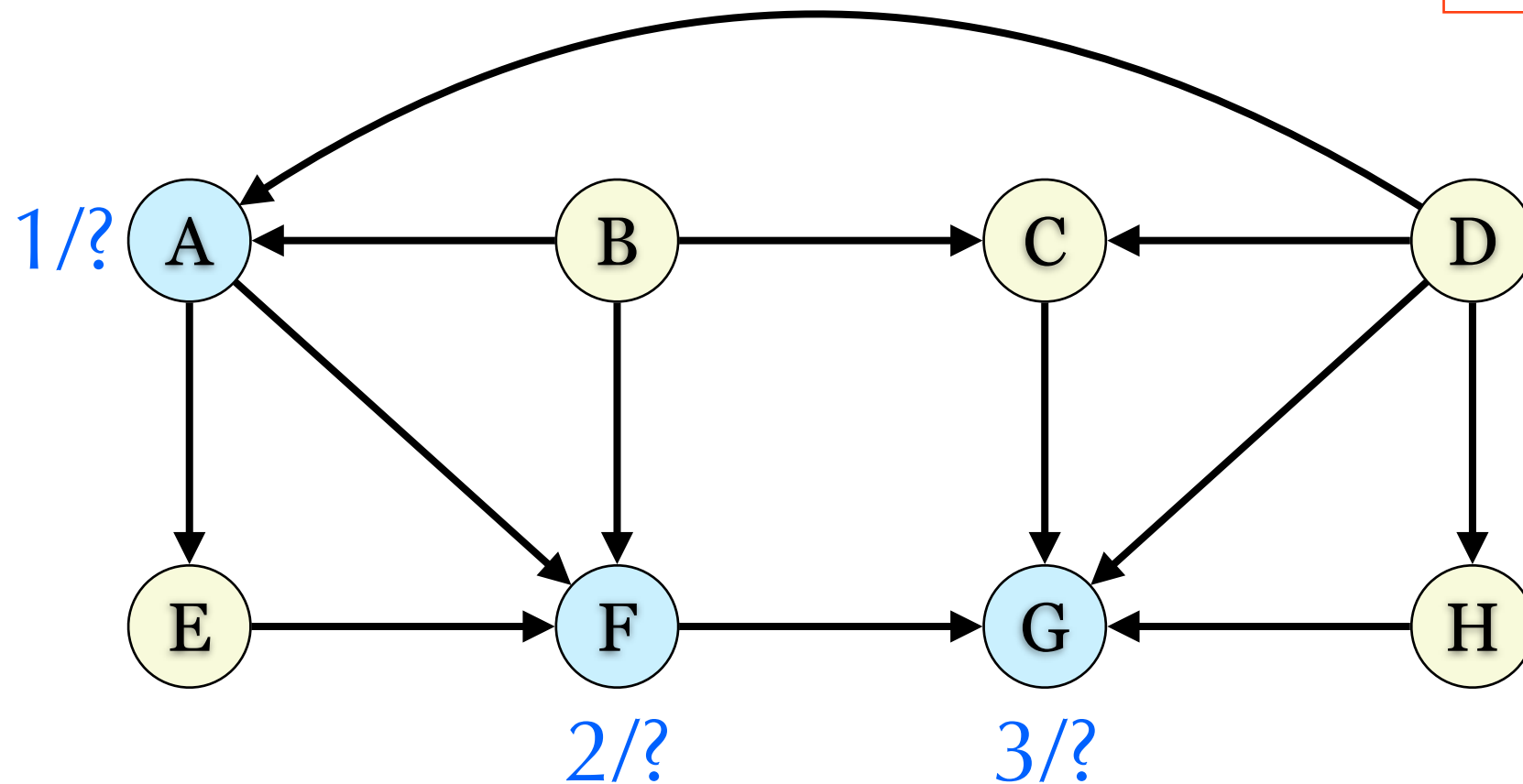
Stack

A/o	F/o	G/o					

Example: DFS

Stack
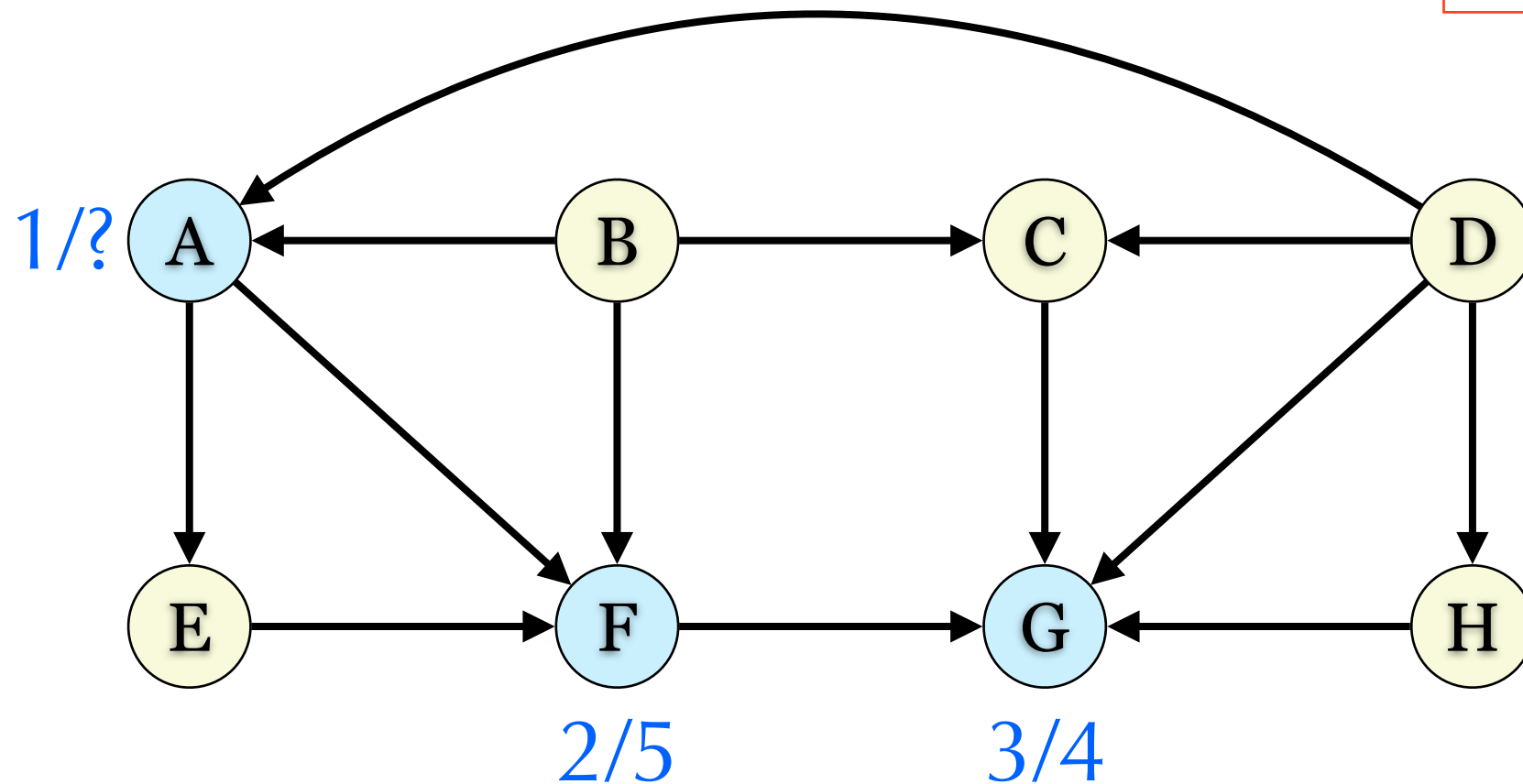
A/0	F/1						

Example: DFS

Stack

A/1							

Example: DFS

Stack

A/1	E/0						

Example: DFS

Stack

A/2							
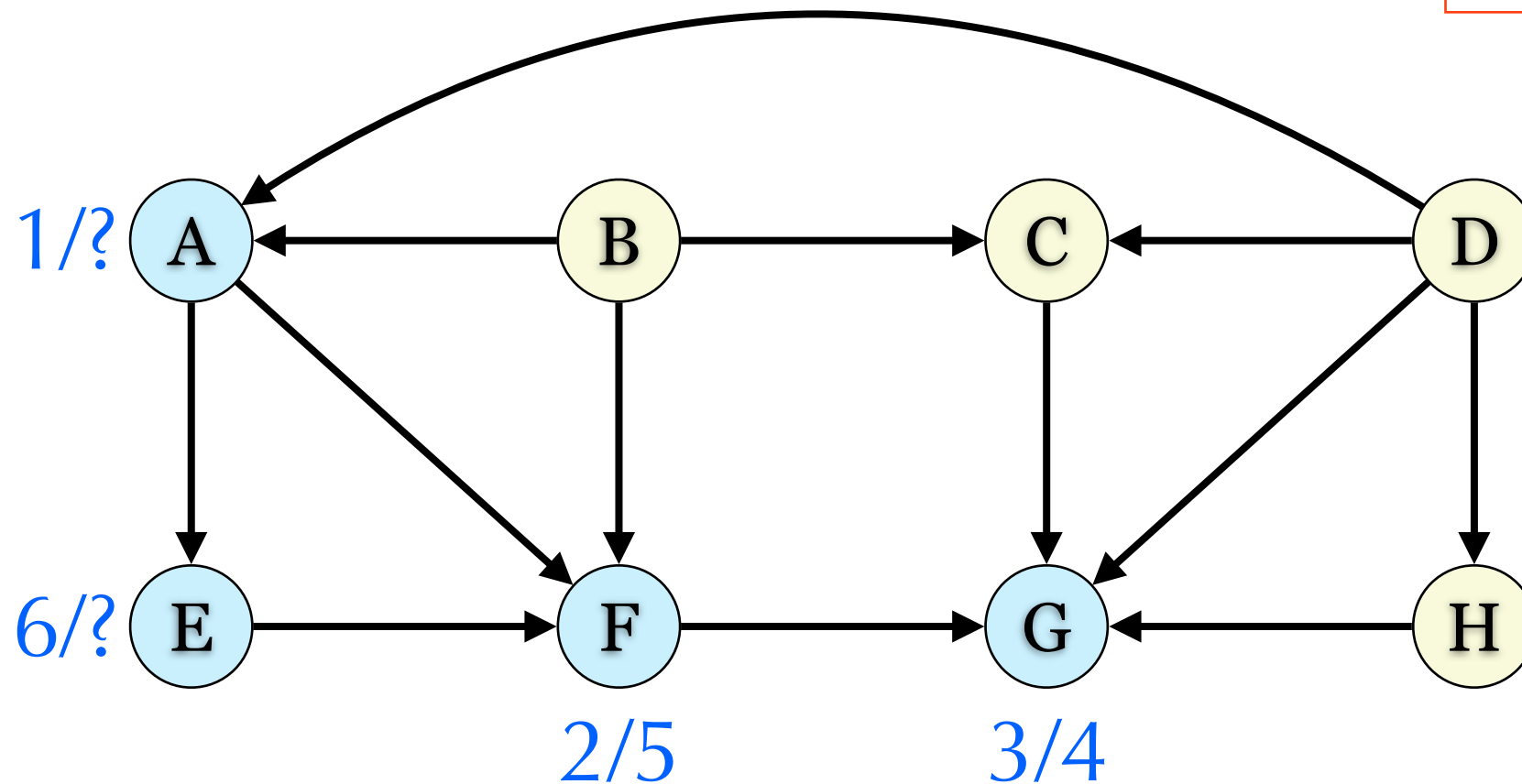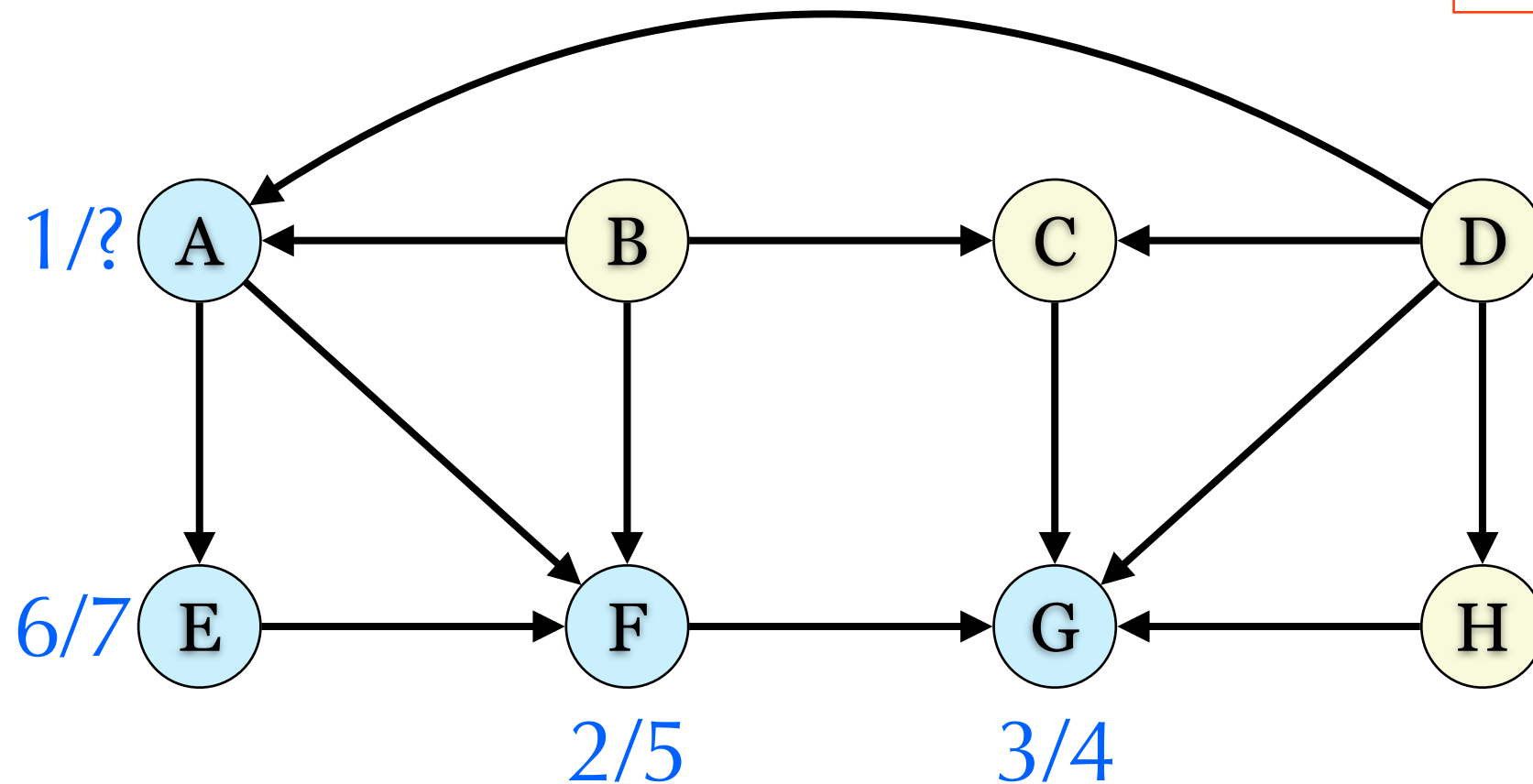
Example: DFS

Stack

Done!

Example 2: DFS

1/?

Stack | A/o

Example 2: DFS

1/?   A

2/?   E
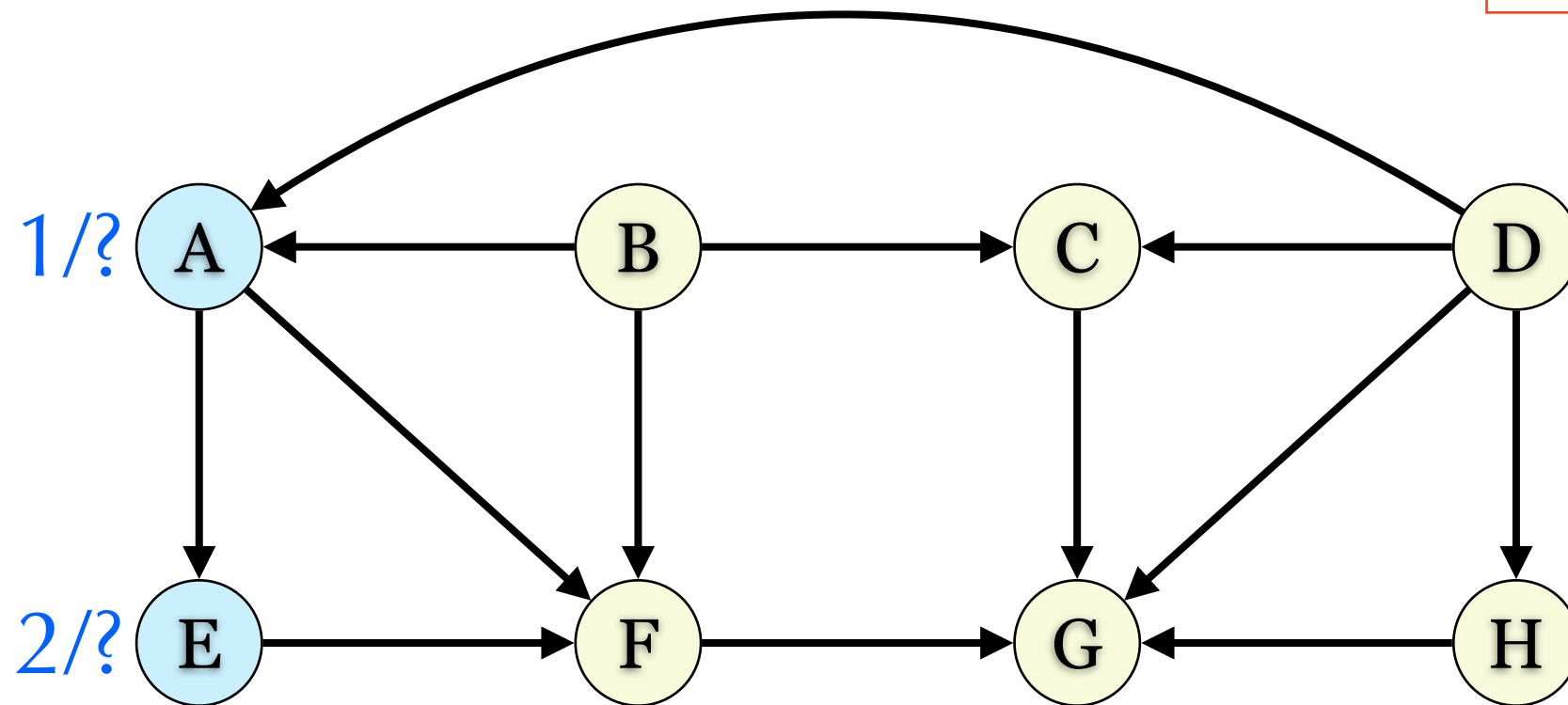
Stack

A/o	E/o						

Example 2: DFS

1/? A ← B → C ← D

2/? E → F → G   H

3/?
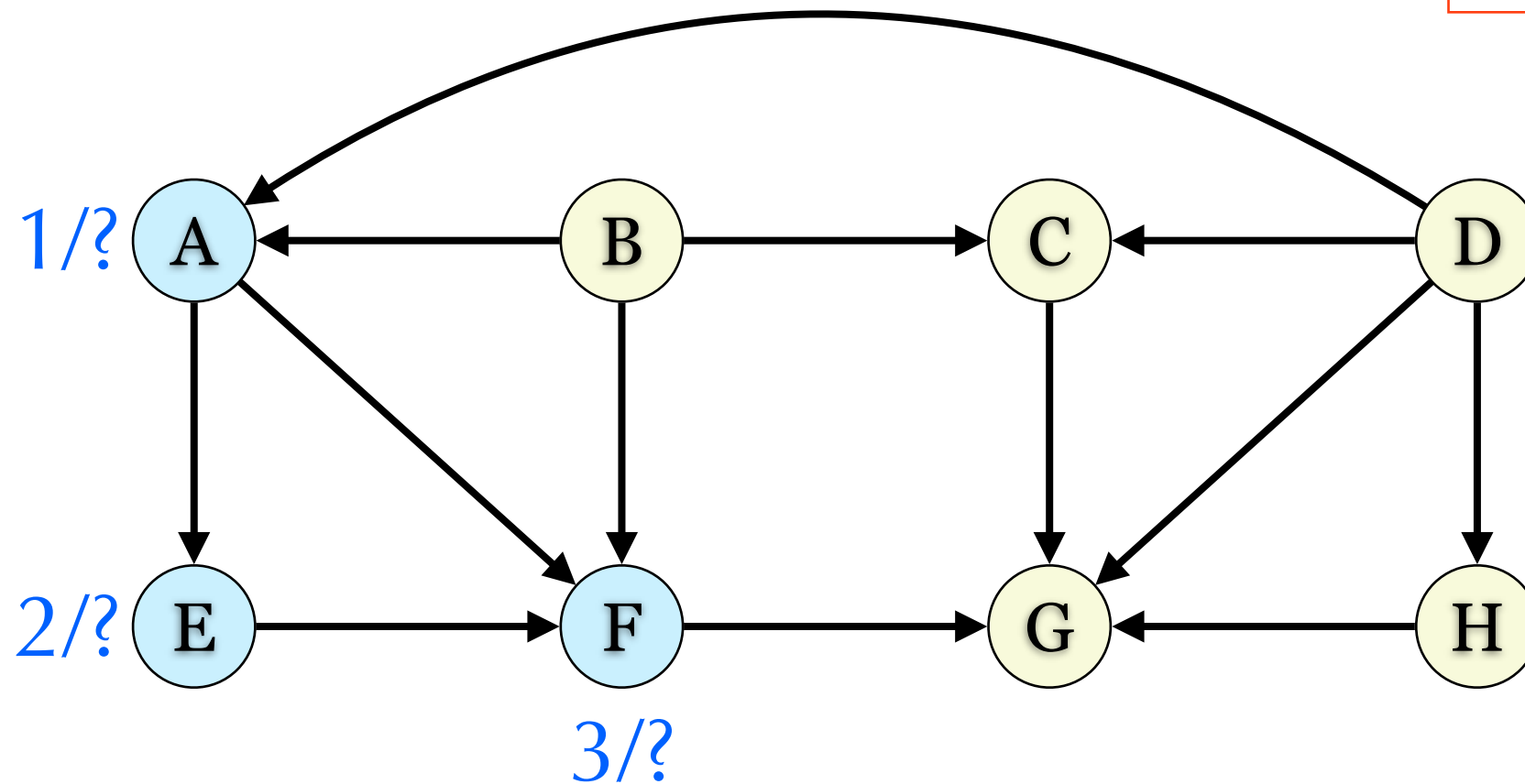
Stack | A/o | E/o | F/o | | | | | |

Example 2: DFS
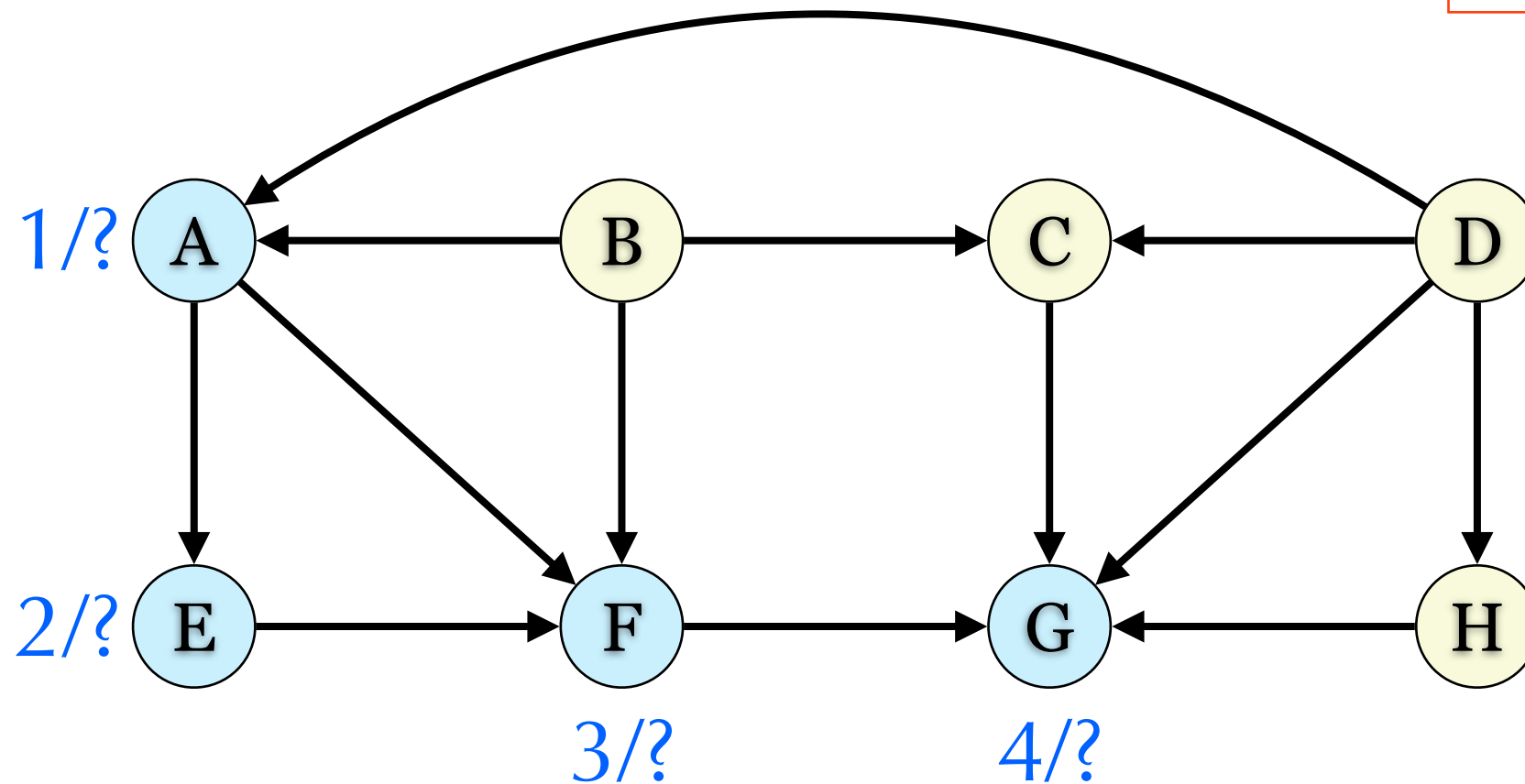
Stack

A/o	E/o	F/o	G/o				

Example 2: DFS

Stack

A/0	E/0	F/1					

Example 2: DFS

1/? A   B   C   D

2/? E   F   G   H

3/6   4/5
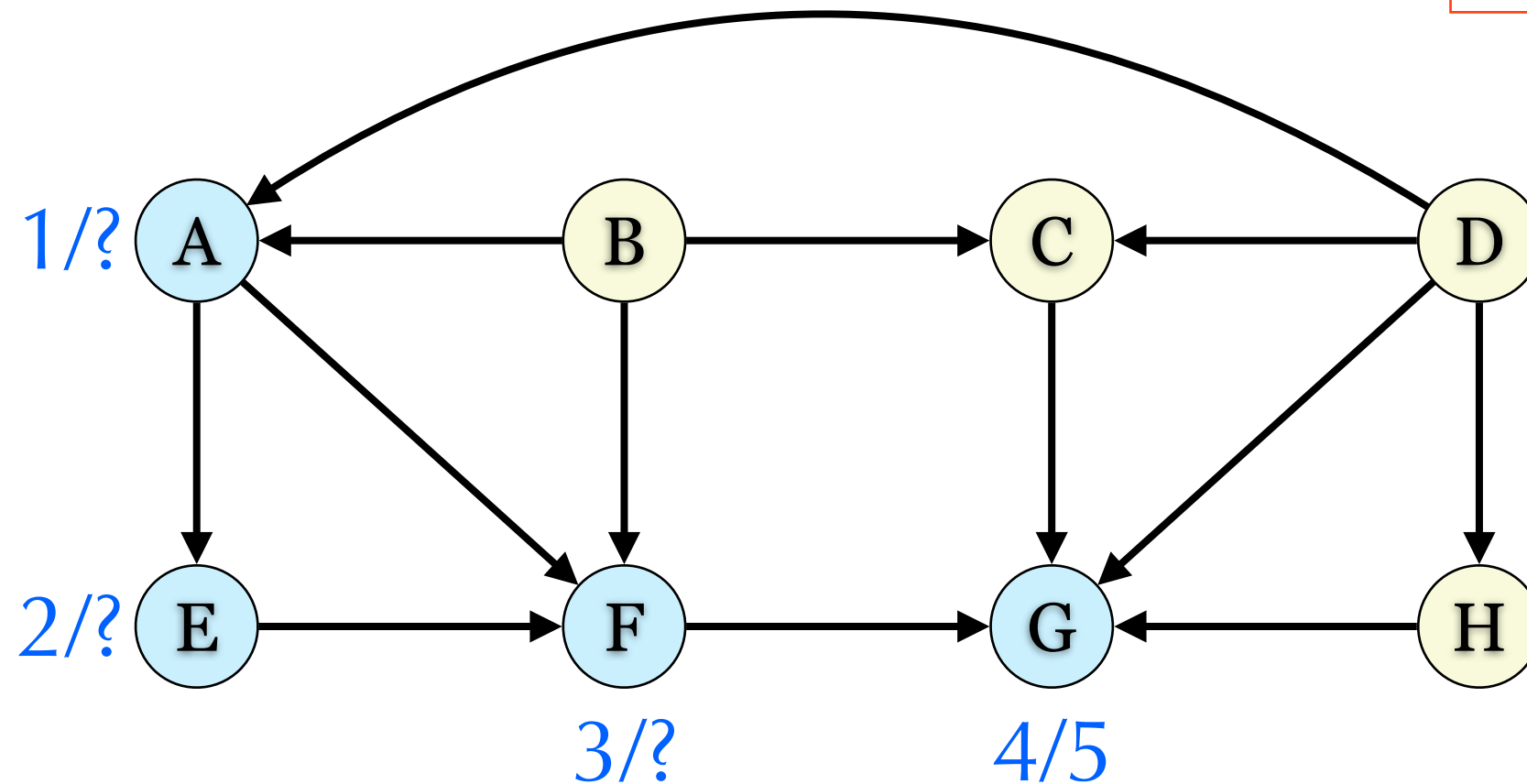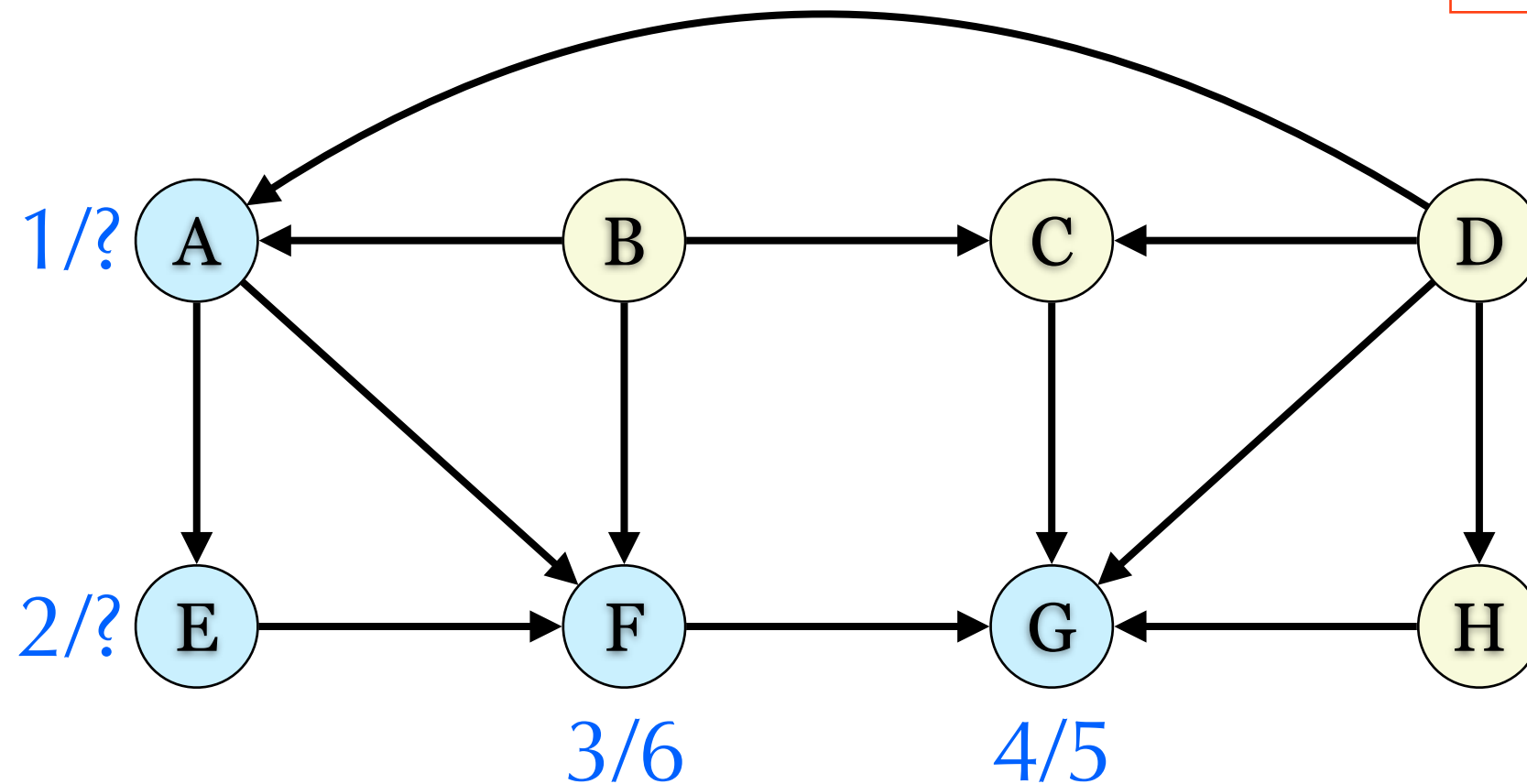
Stack | A/0 | E/1 | | | | | | |

Example 2: DFS
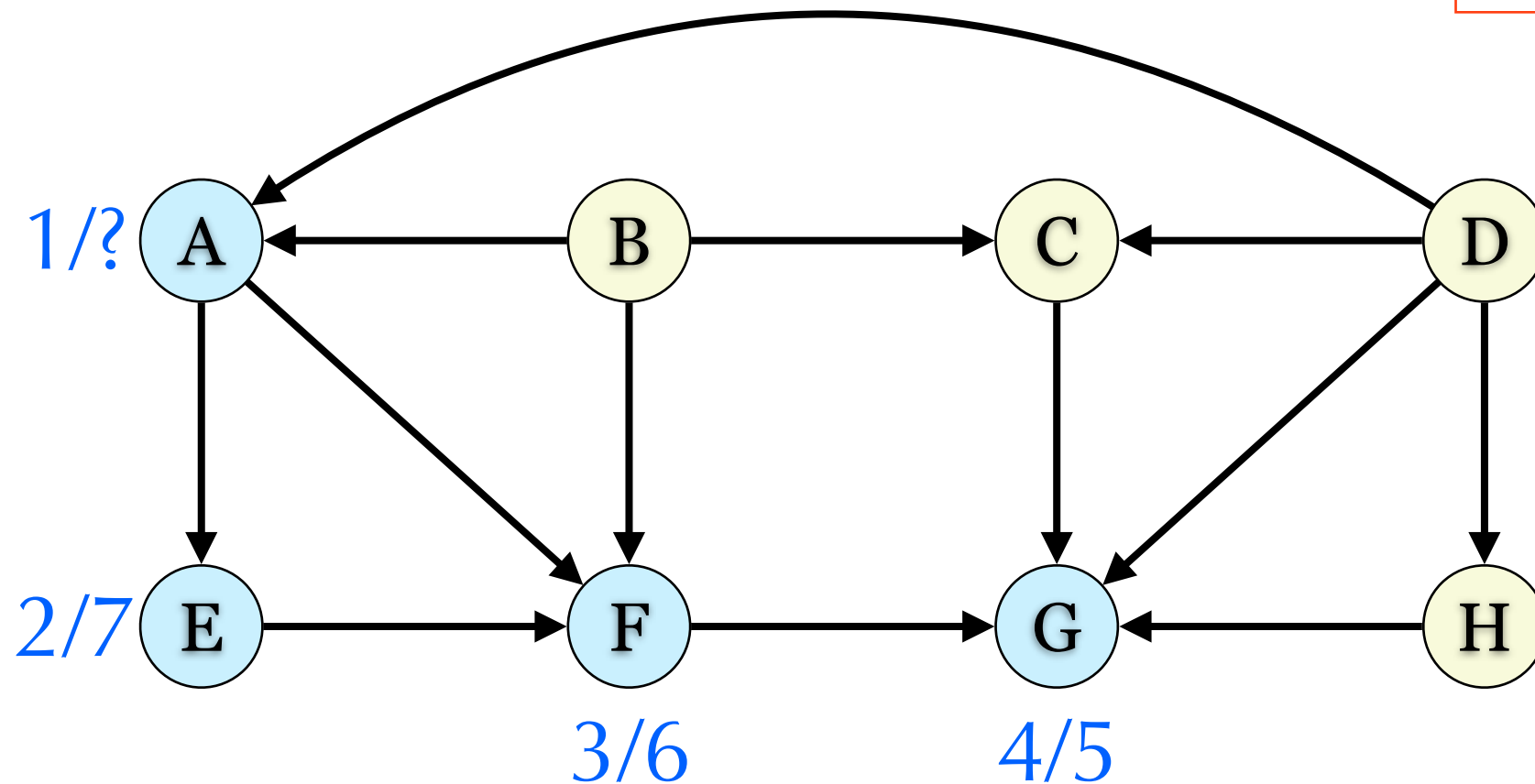
Stack

A/1							

Example 2: DFS

1/8 A   B   C   D

2/7 E   F   G   H

3/6   4/5
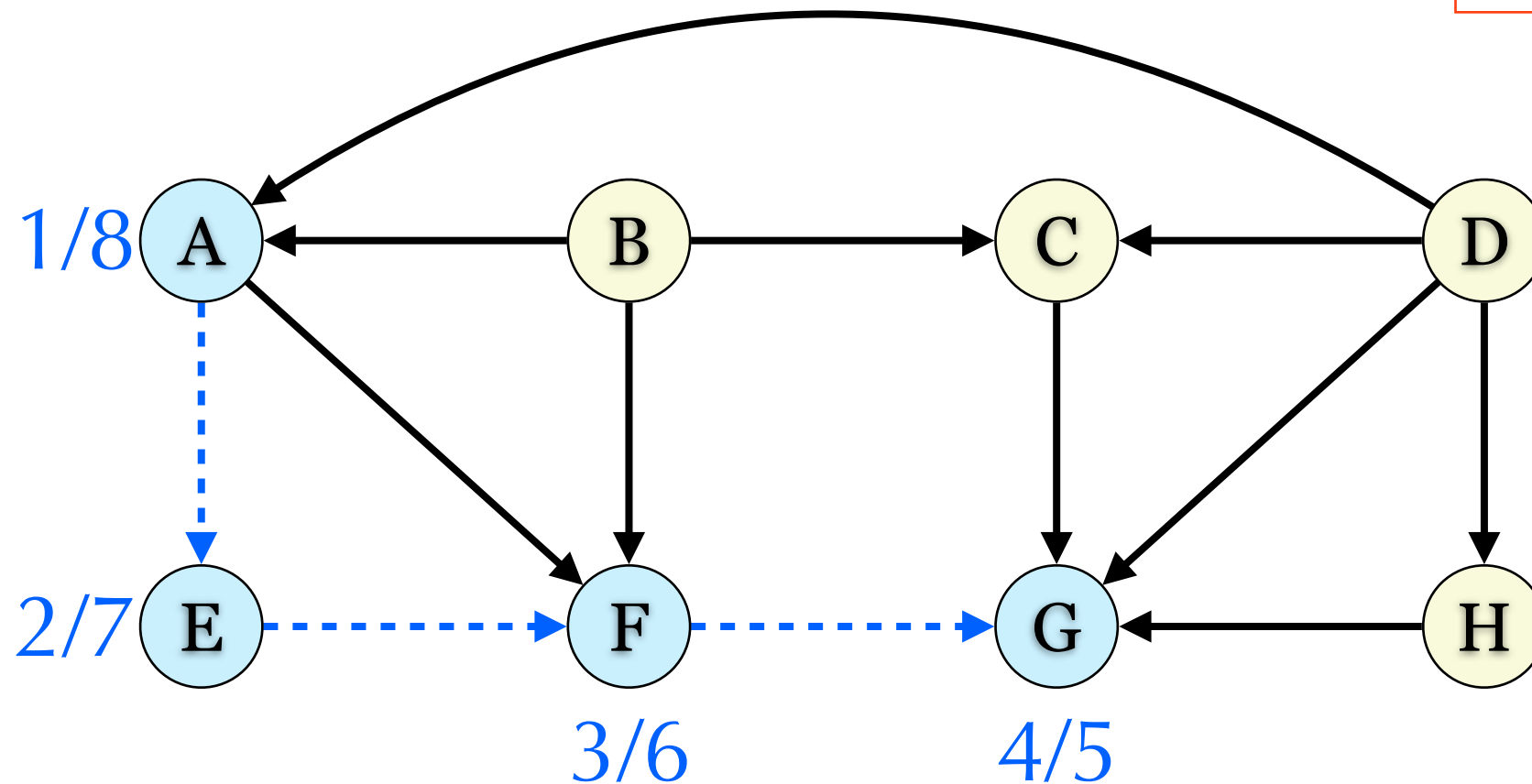
Stack

Done!

Example 3: DFS
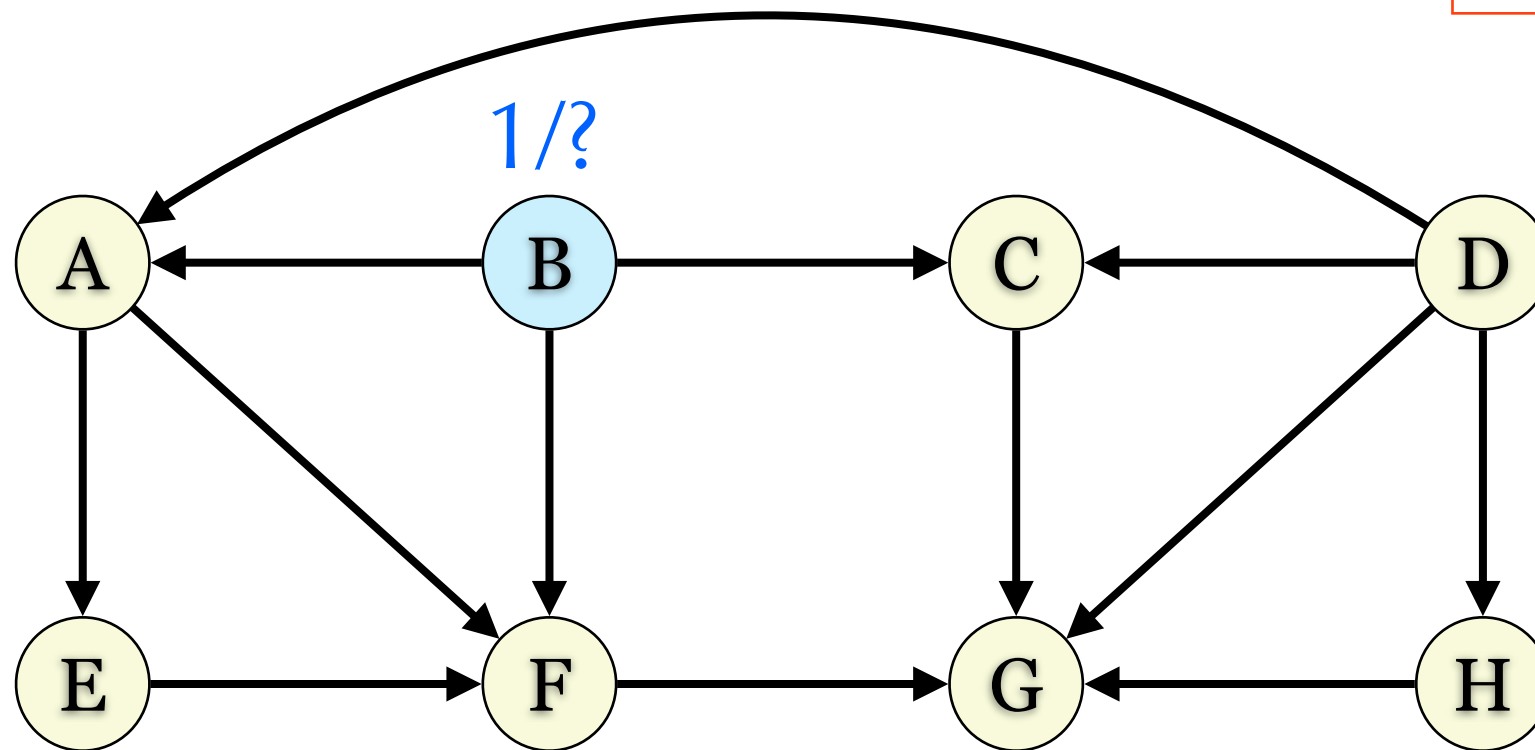
Stack | B/o | | | | | | | |

Example 3: DFS
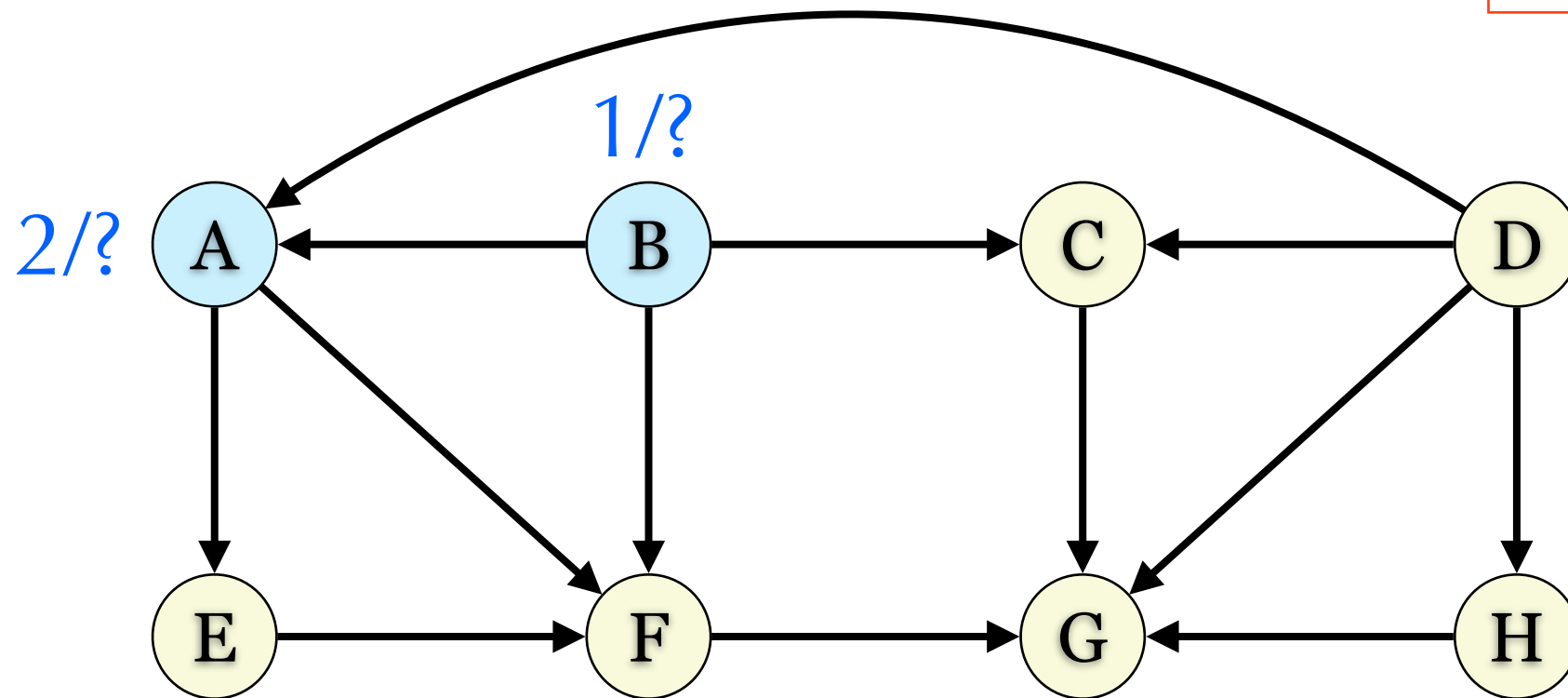
Stack | B/o | A/o | | | | | | |

Example 3: DFS
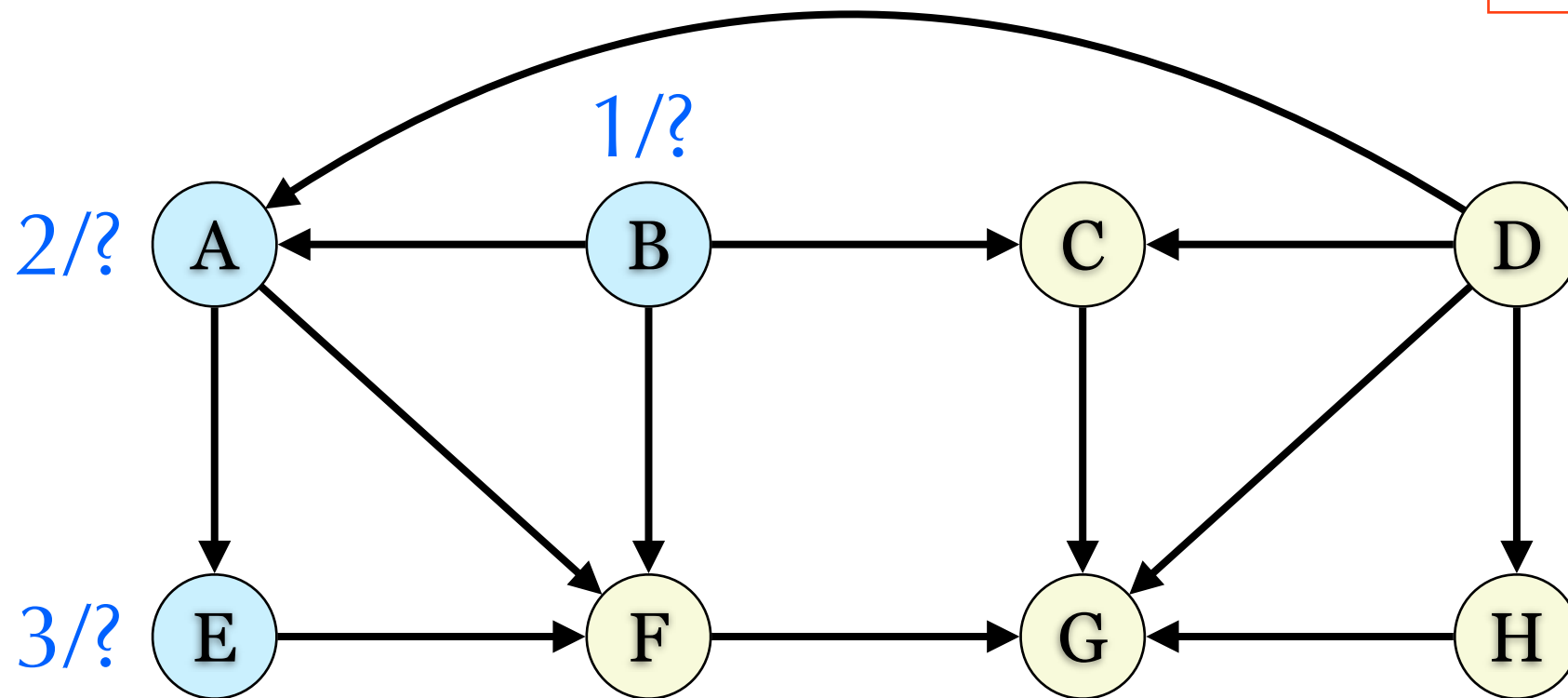
Stack

B/o	A/o	E/o					

Example 3: DFS
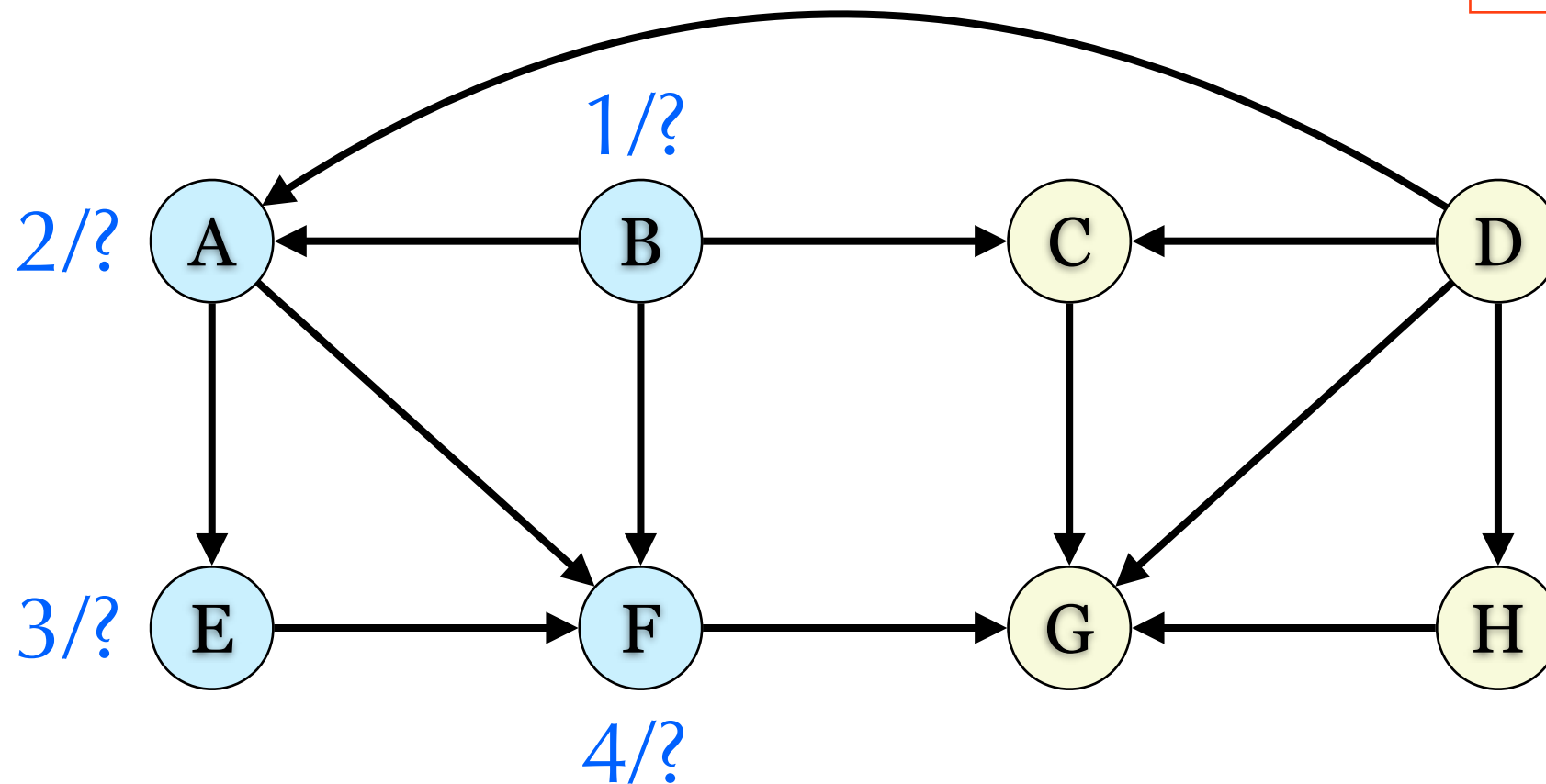
Stack

B/o	A/o	E/o	F/o				

Example 3: DFS
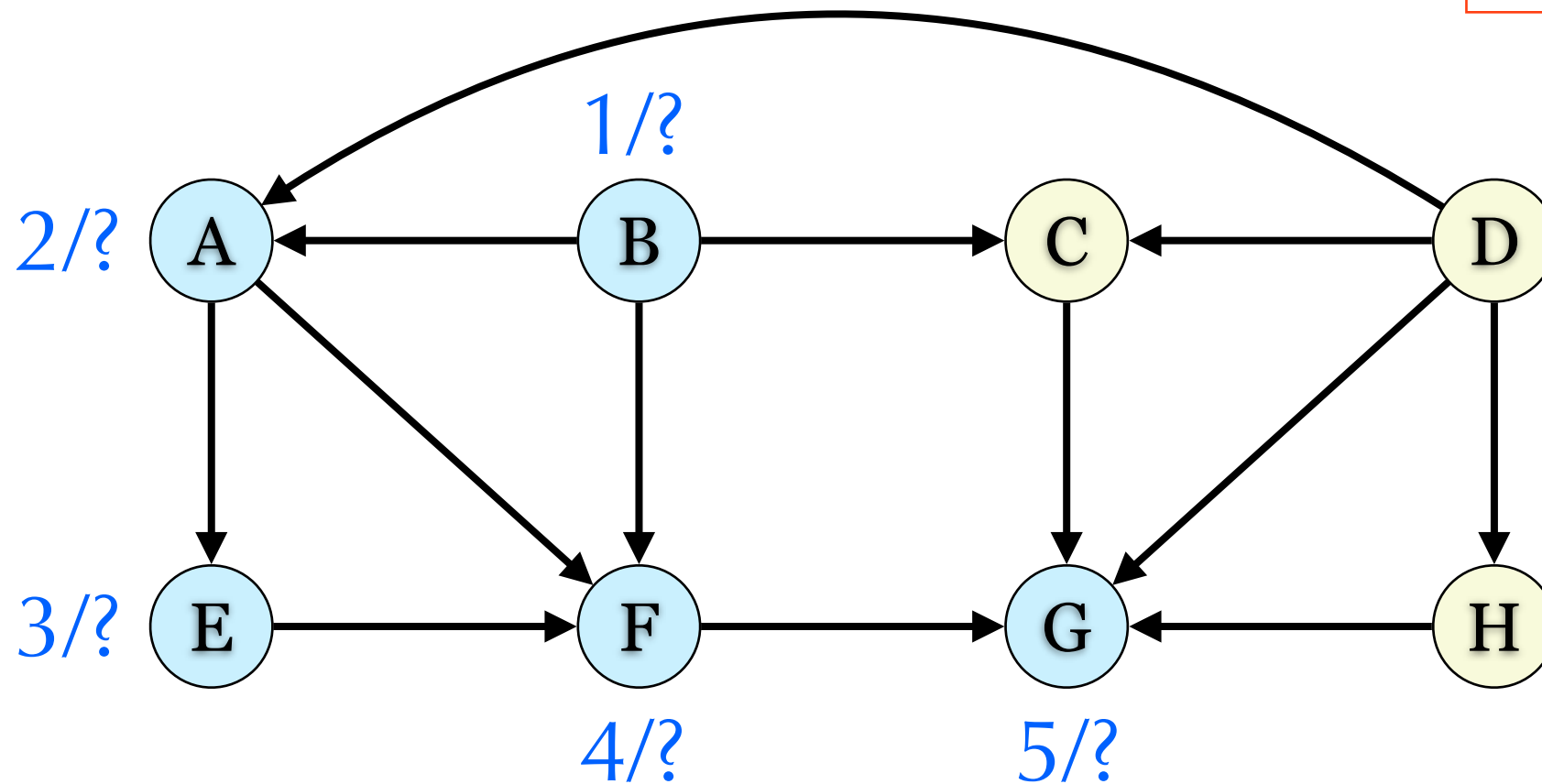
Stack	B/o	A/o	E/o	F/o	G/o			

Example 3: DFS

Stack
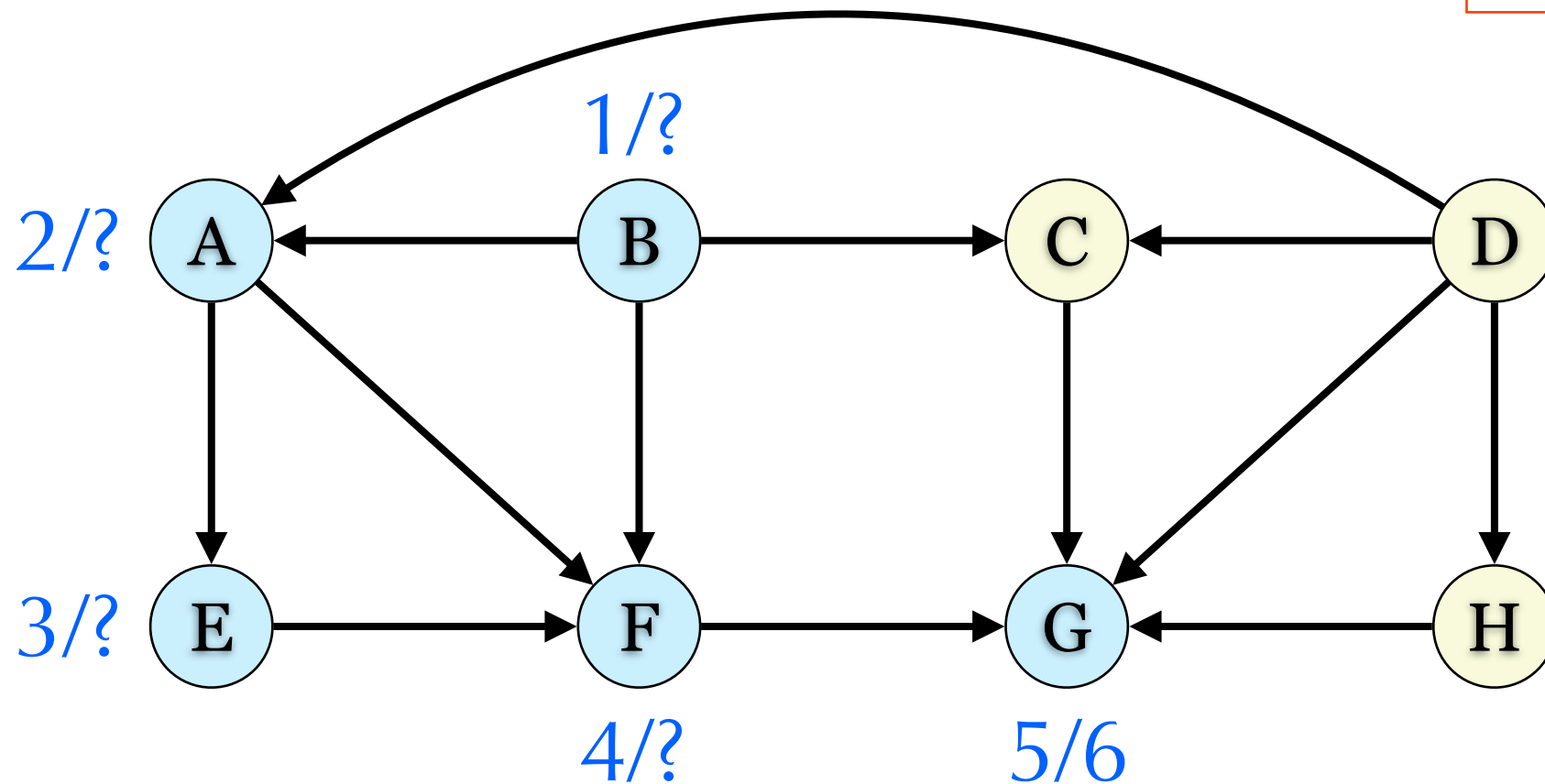
B/0	A/0	E/0	F/1				

Example 3: DFS

Stack

B/0	A/0	E/1					
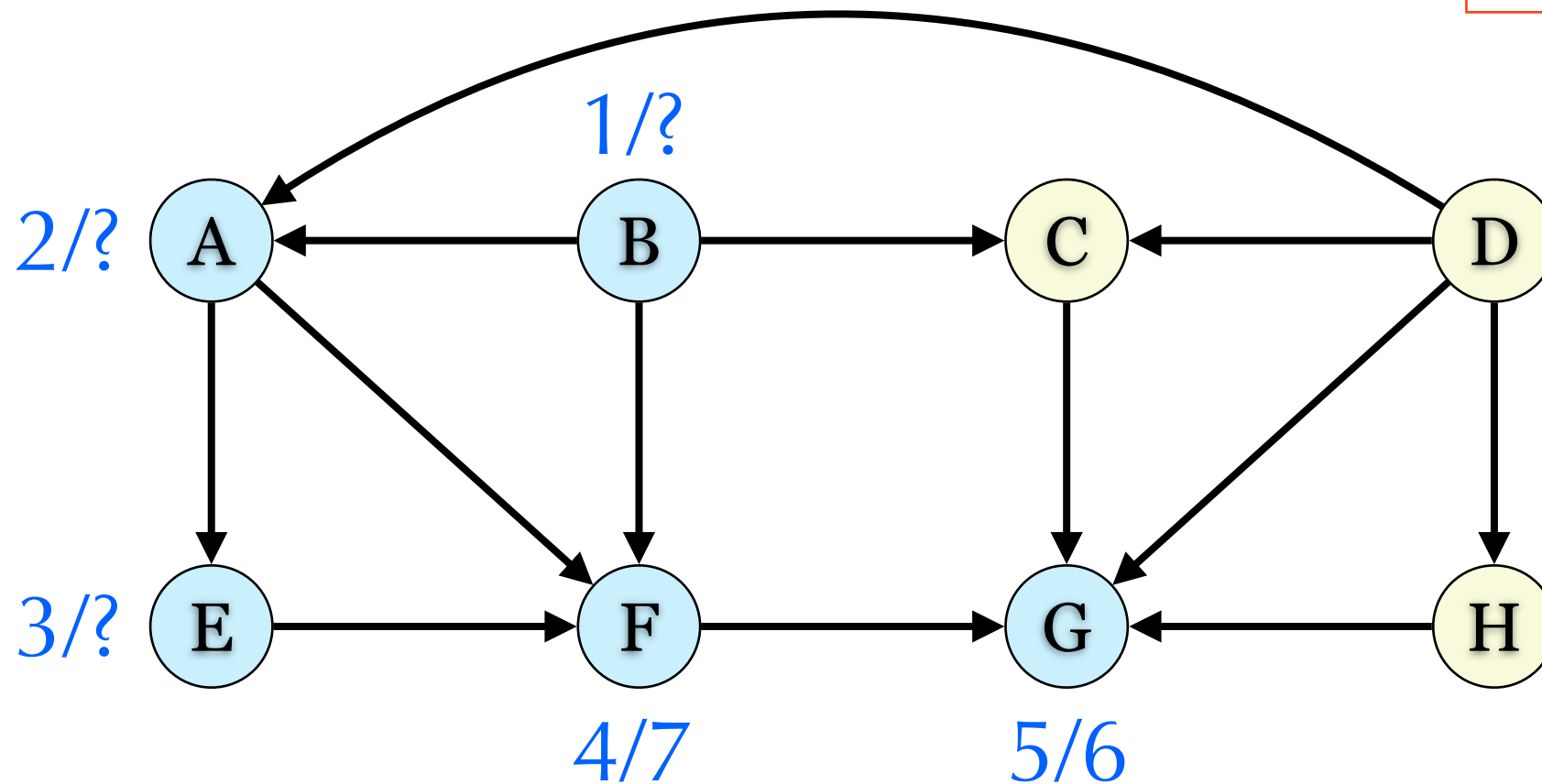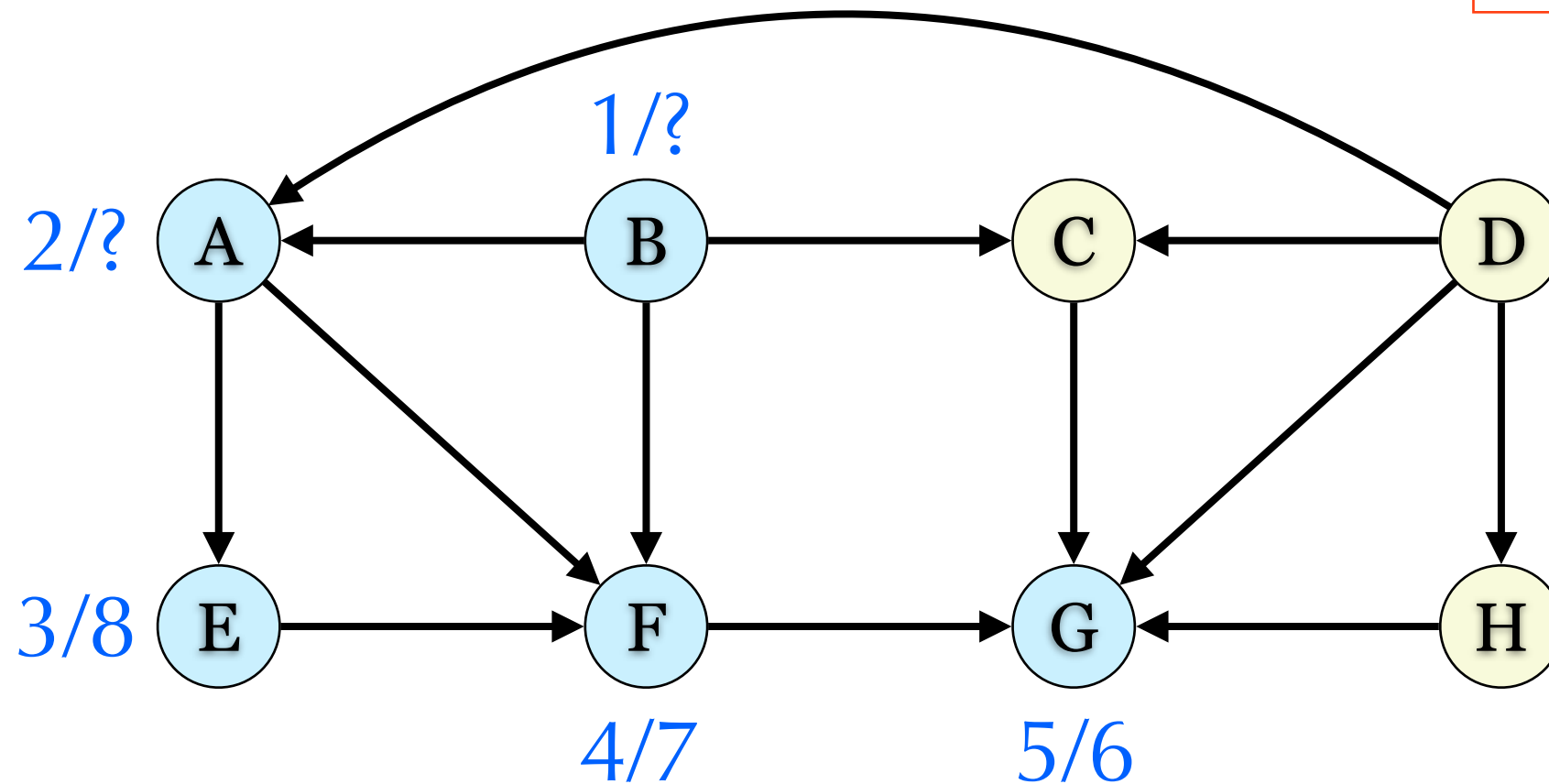
Example 3: DFS

Stack

B/0	A/1						
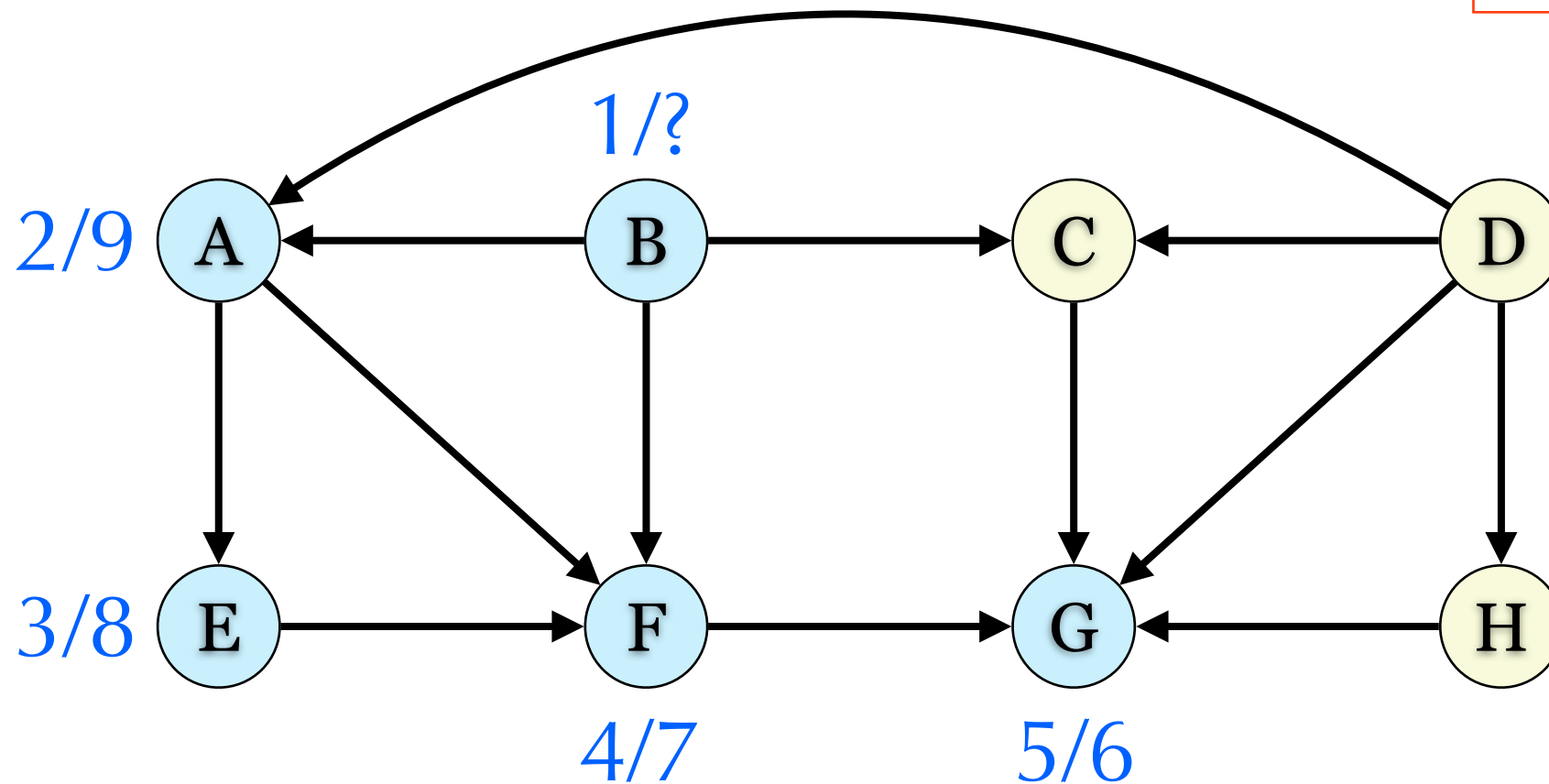
Example 3: DFS

Stack

B/1							

Example 3: DFS

Stack

B/1	C/0						

Example 3: DFS

Stack | B/2 | | | | | | |

Example 3: DFS

1/12   10/11

2/9 A   B   C   D

3/8 E   F   G   H

4/7   5/6

Stack

Done!