Lecture 2: Uninformed Search

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS410/index.html

Depth-First Search : Algorithm

DFS(G)

for each $u \in V$ **do**

state[u]$\leftarrow$ *un-visited*

put the start node u* into a stack S

While S is !empty

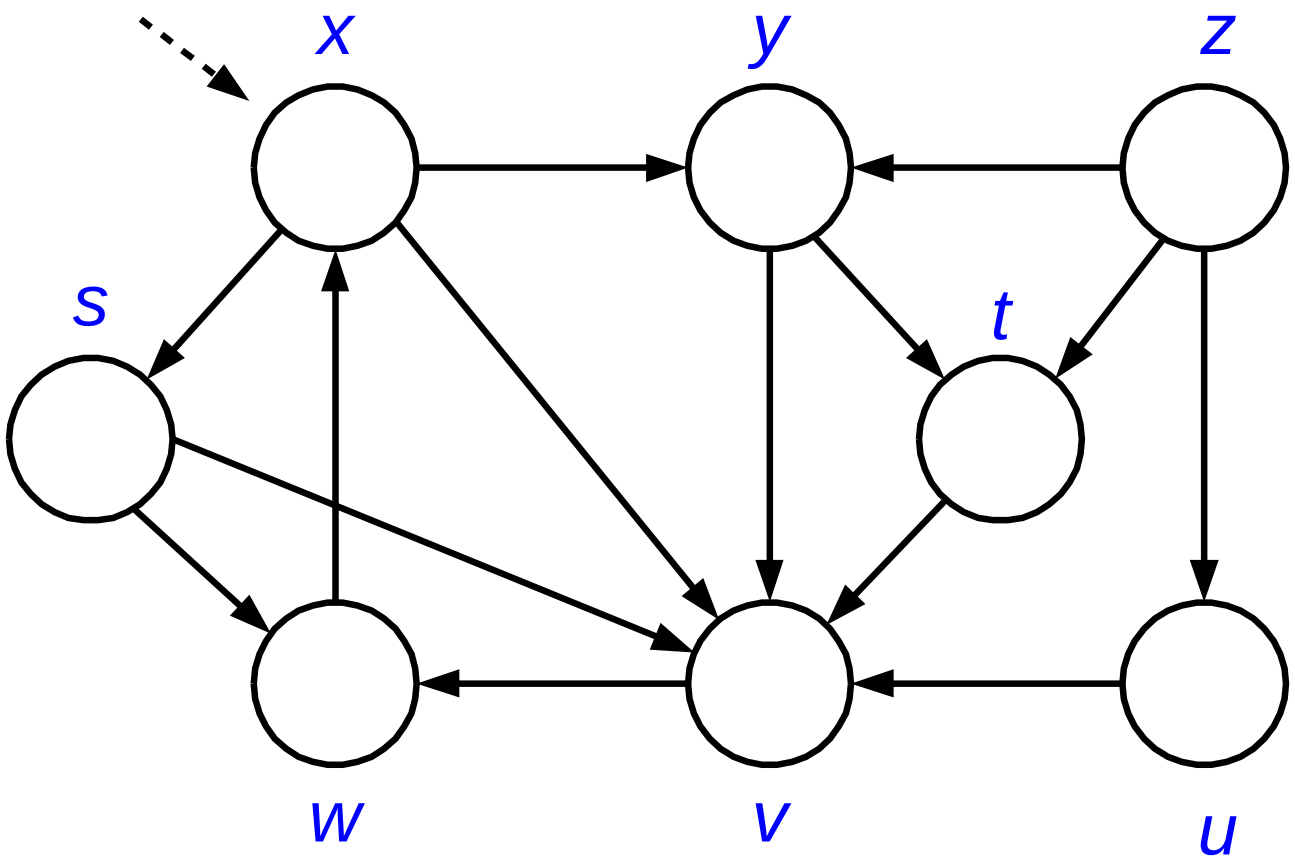
n = S.pop()

if state[n] == un-visited

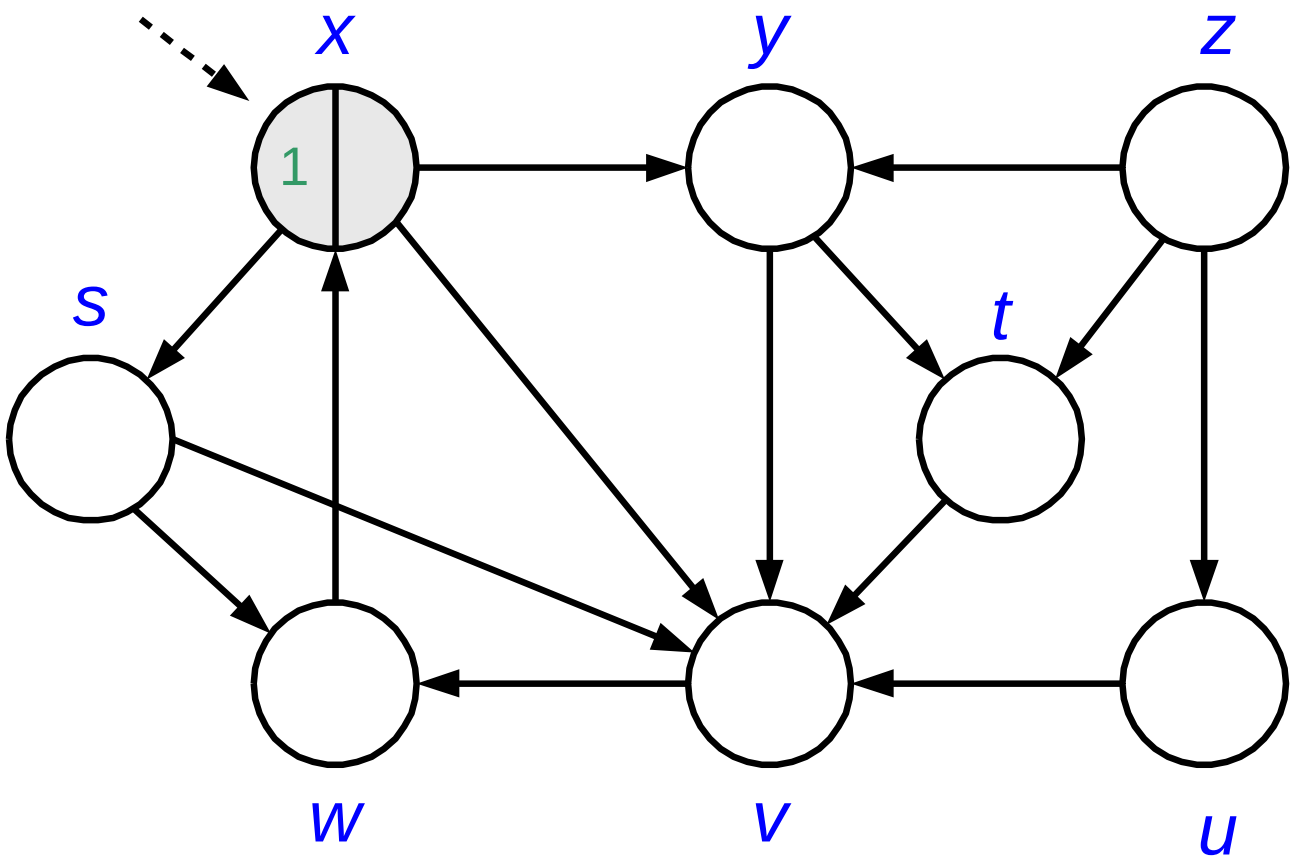
S.push(n.Adj)

state[n] = visited

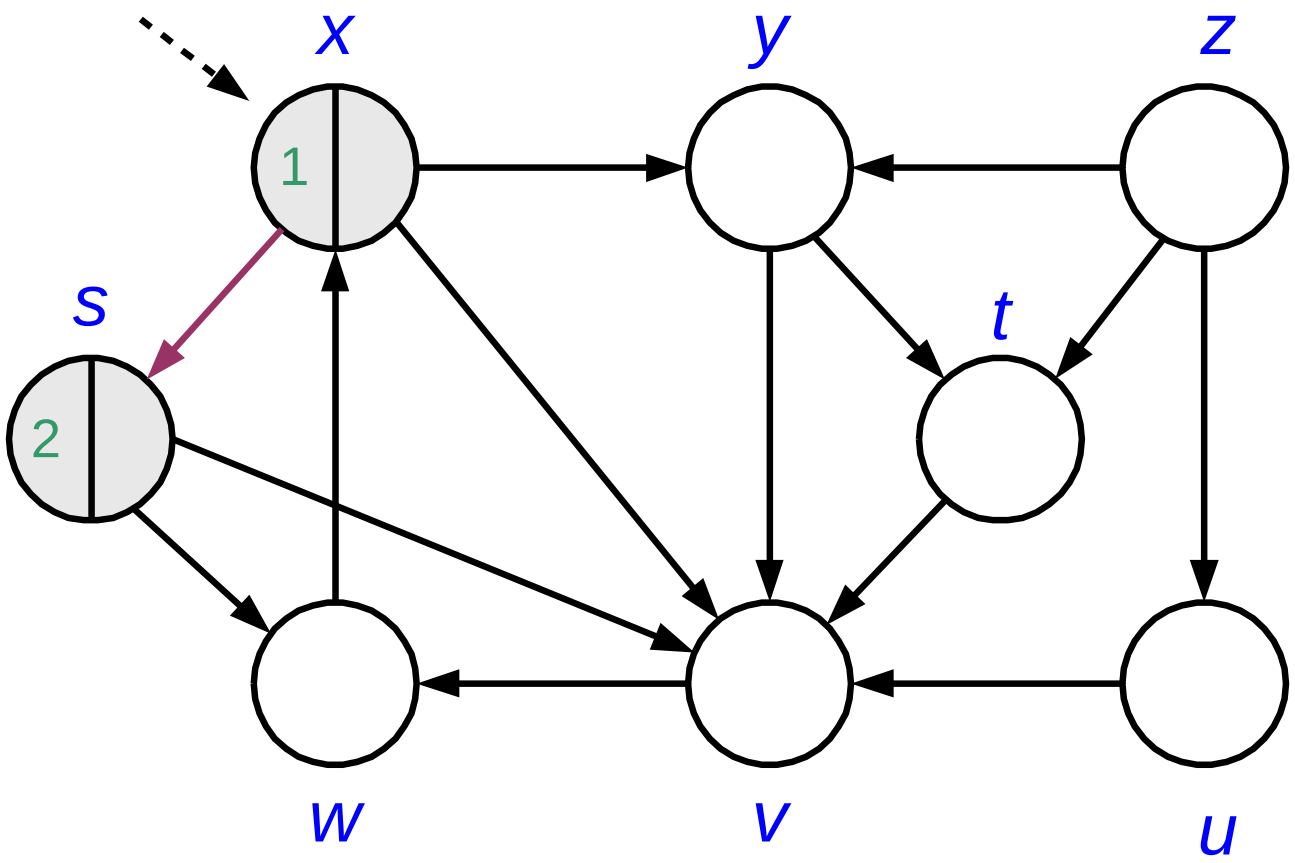
Depth-First Search : Example



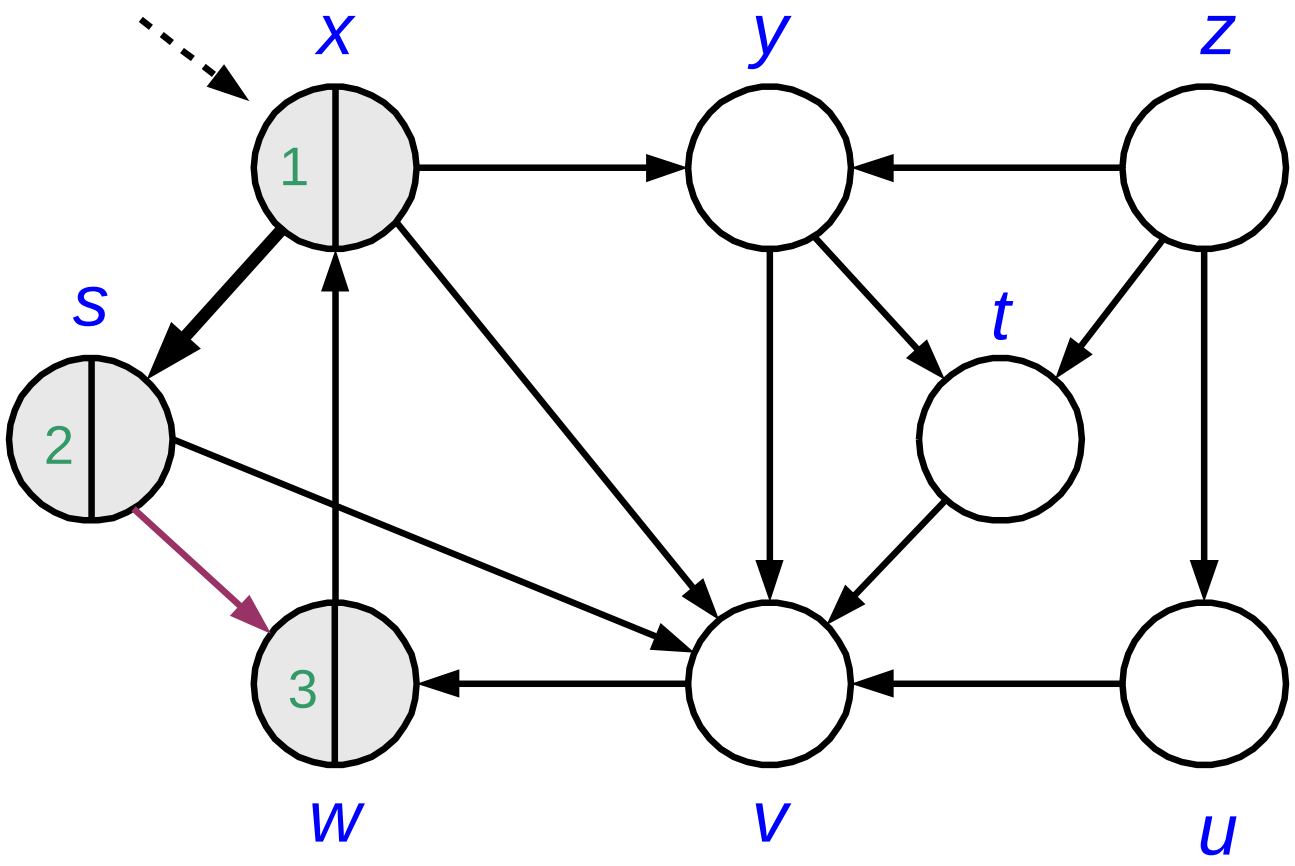
Depth-First Search : Example



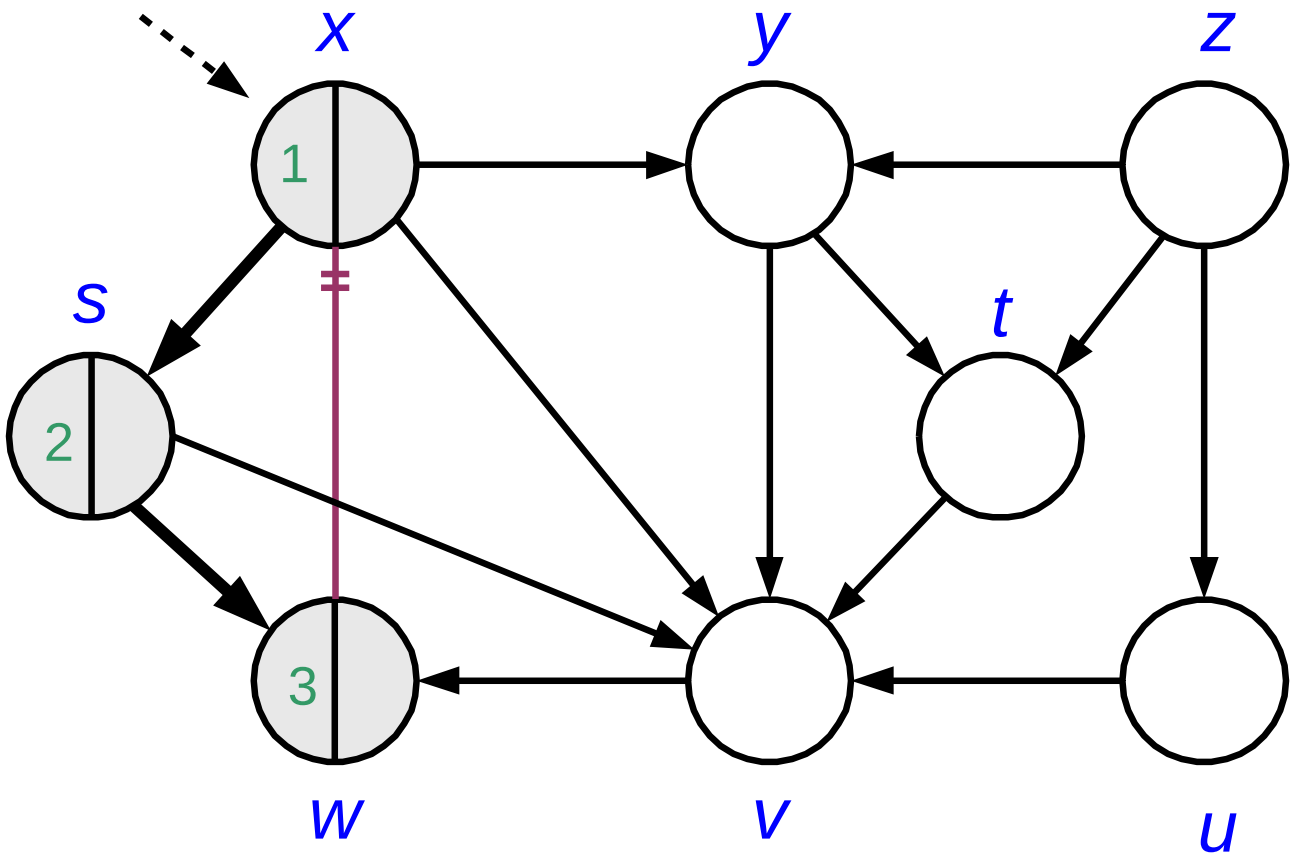
Depth-First Search : Example



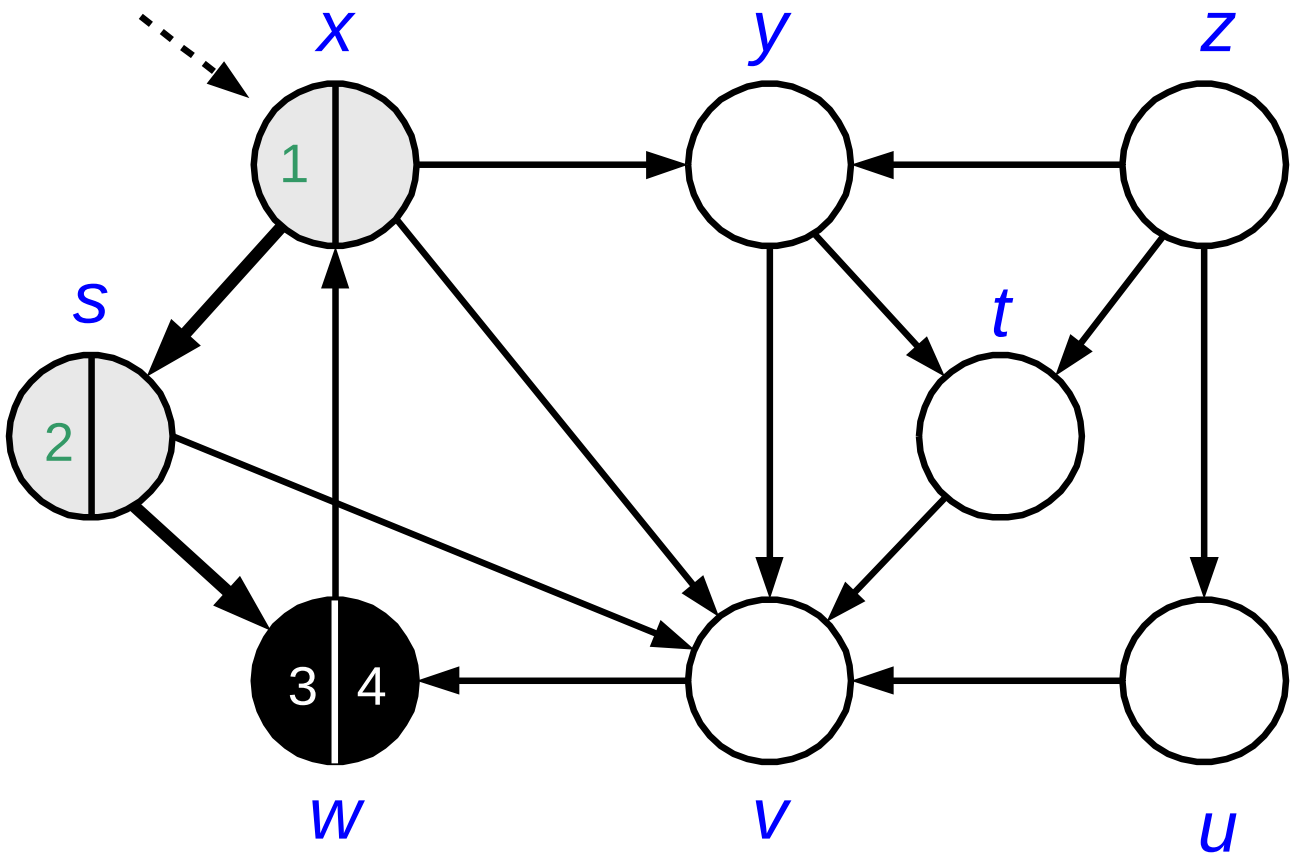
Depth-First Search : Example



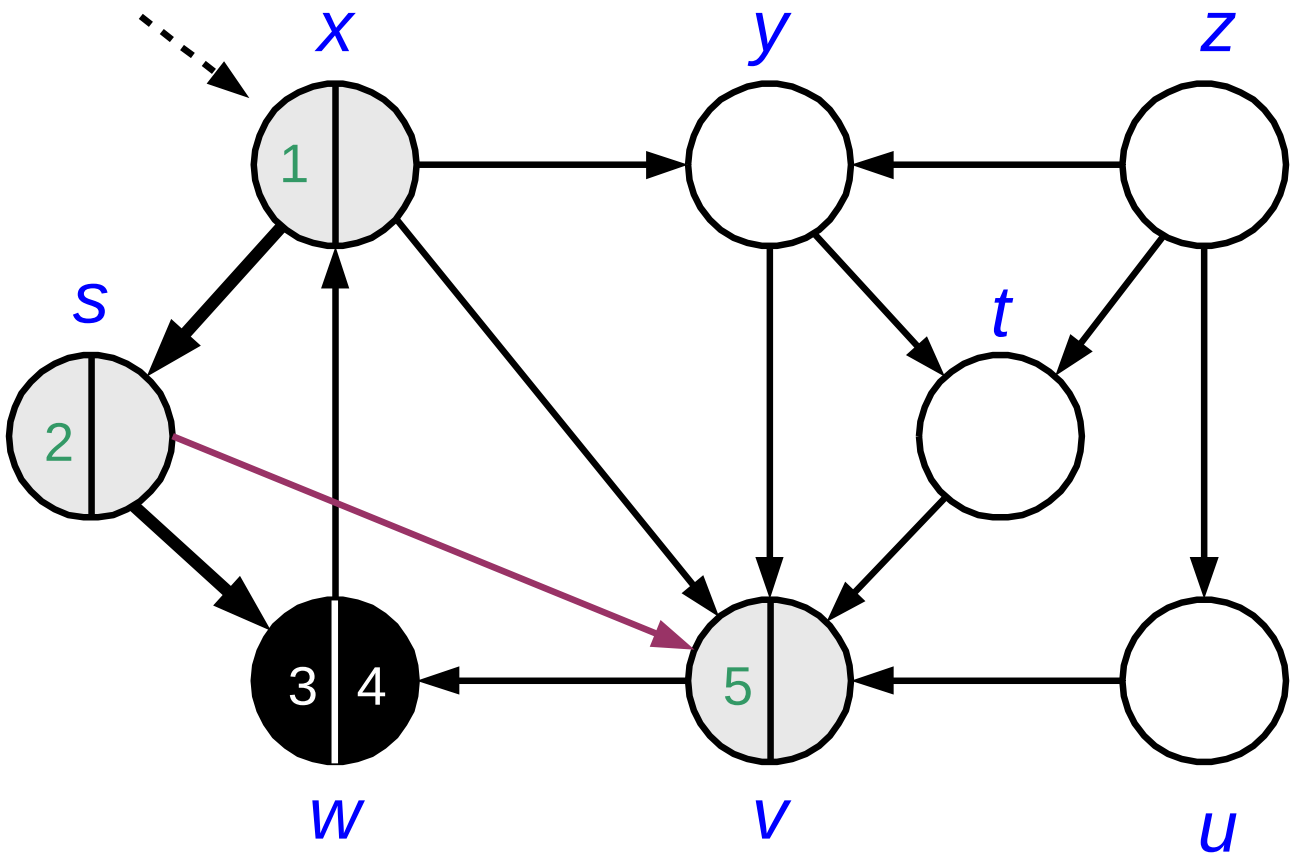
Depth-First Search : Example



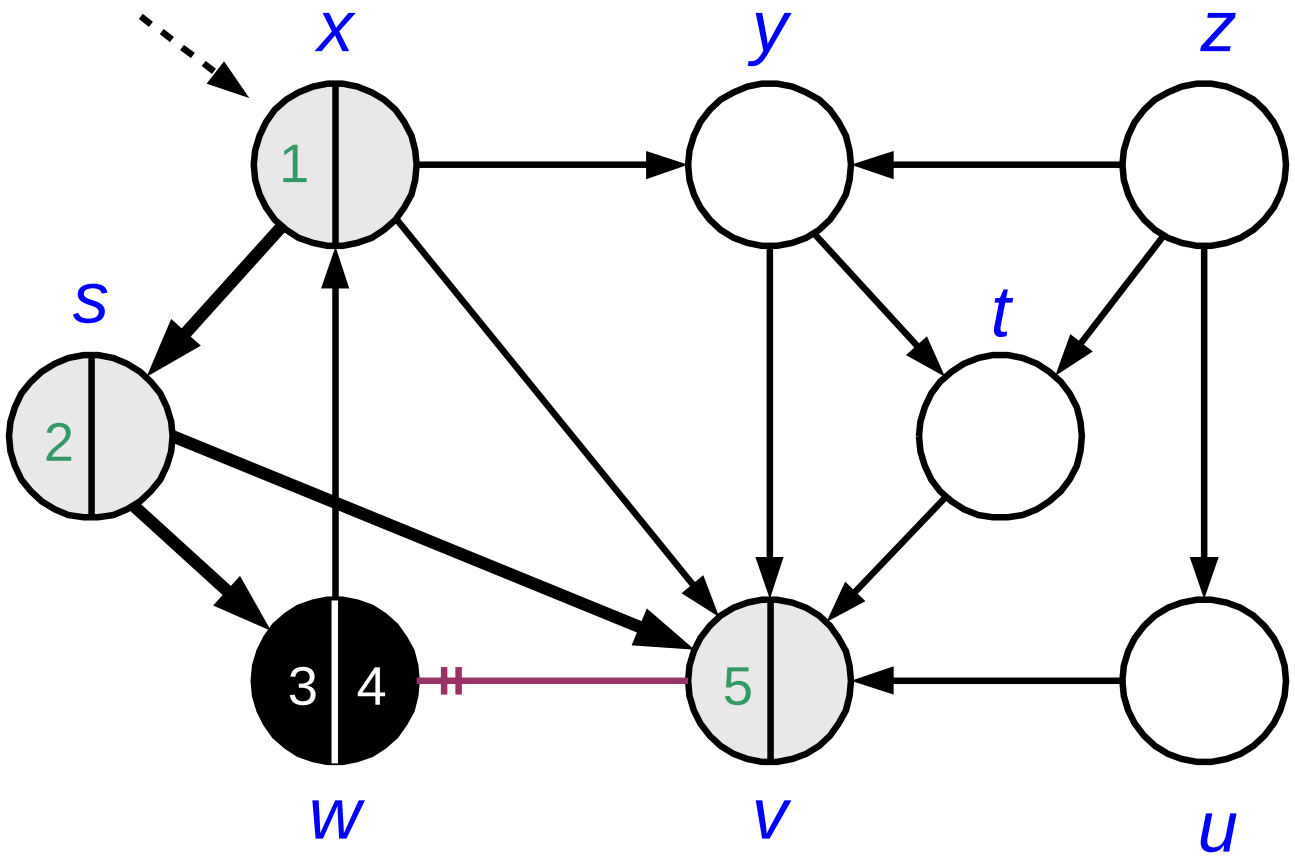
Depth-First Search : Example



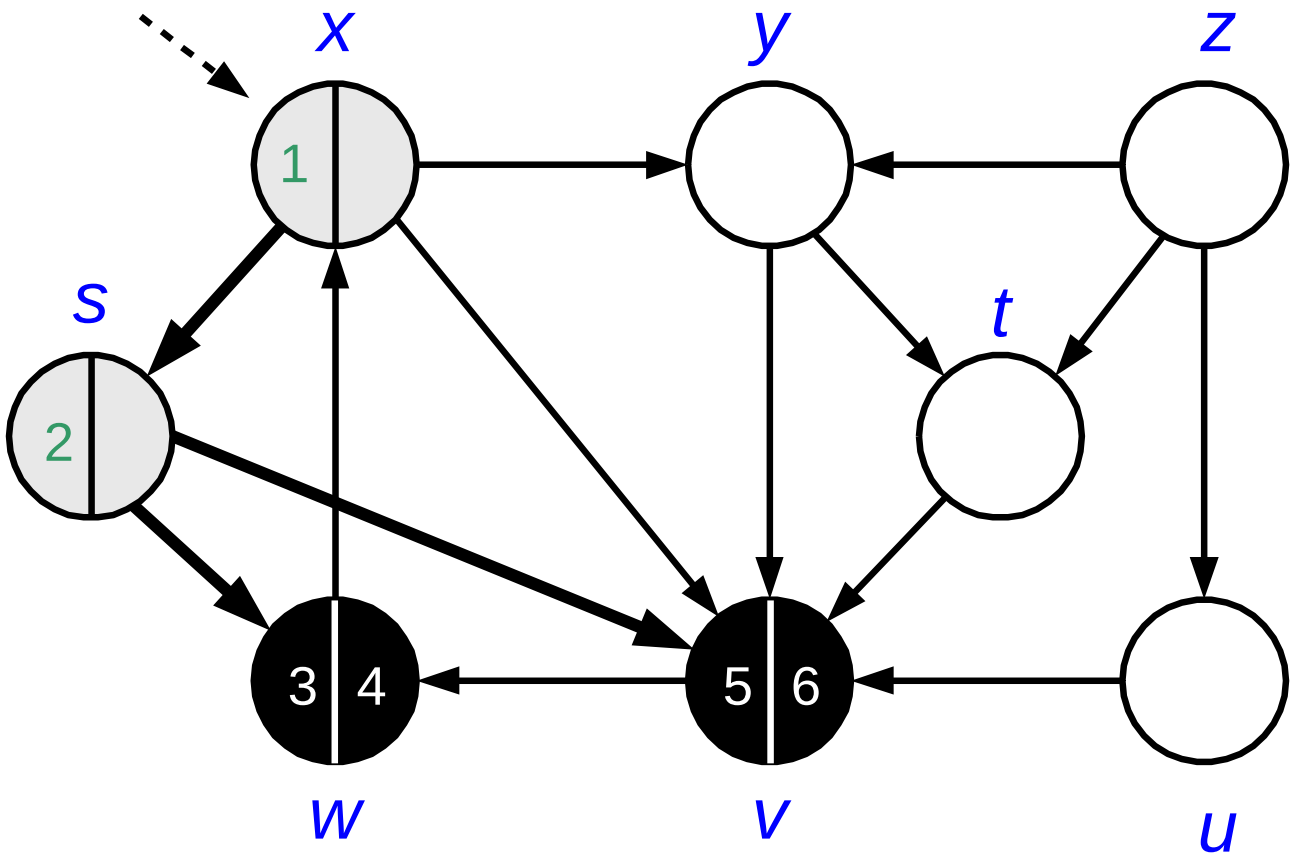
Depth-First Search : Example



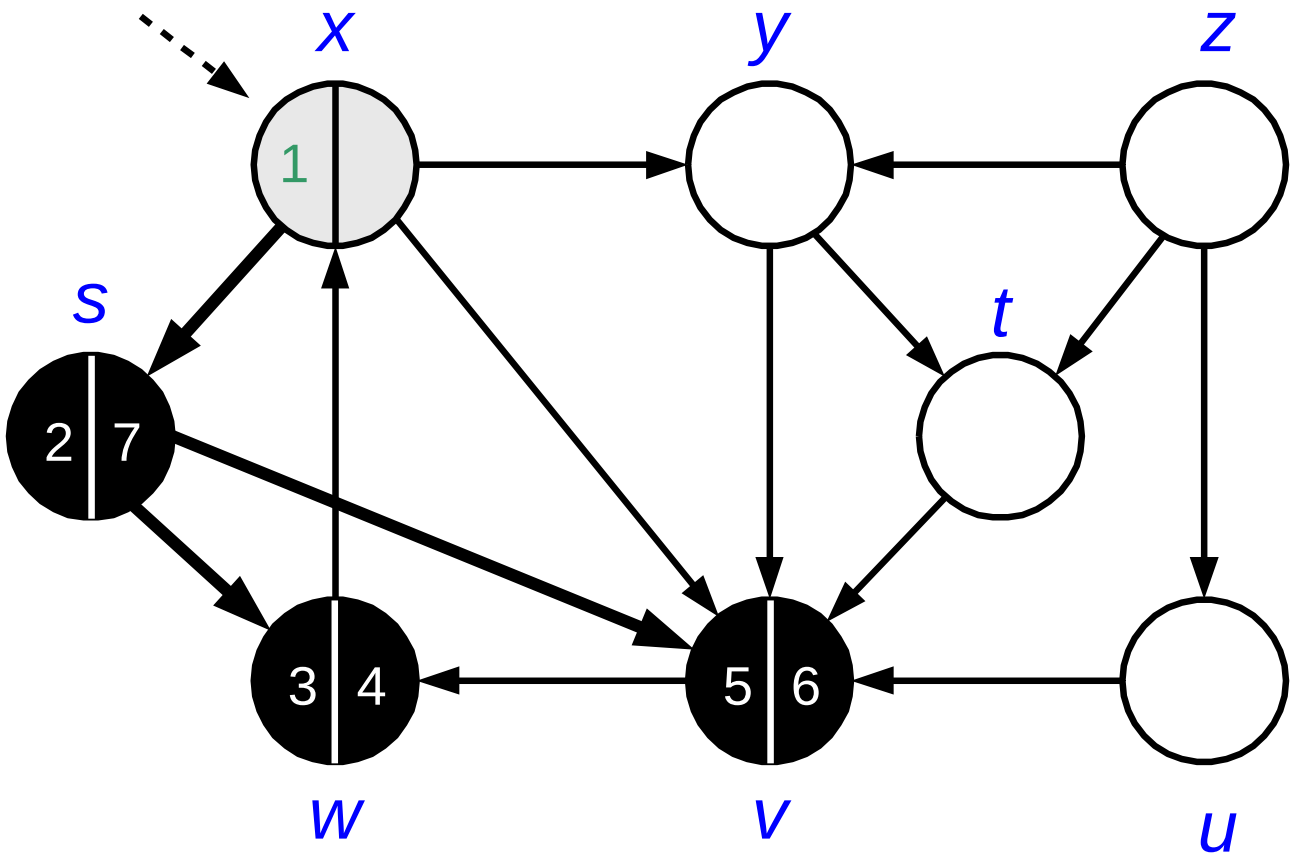
Depth-First Search : Example



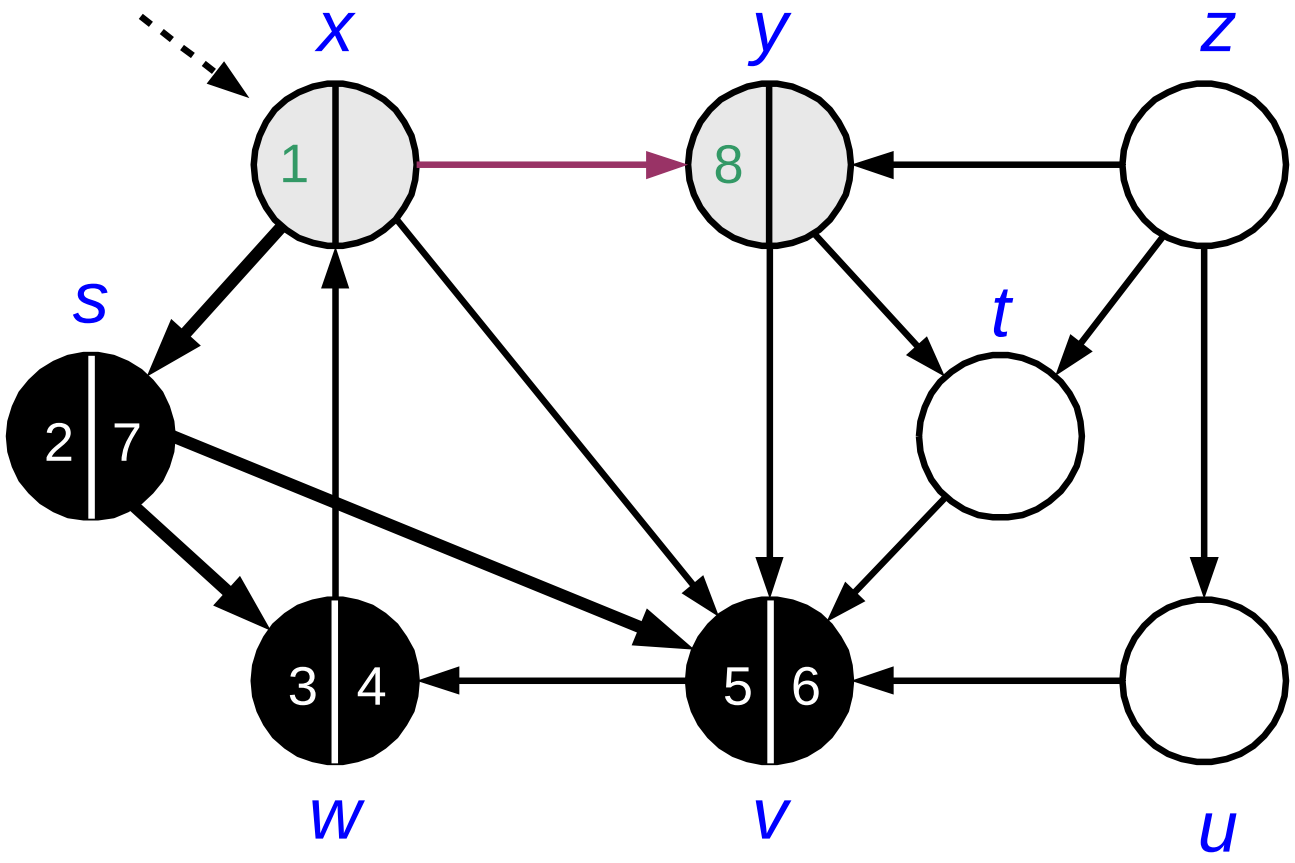
Depth-First Search : Example



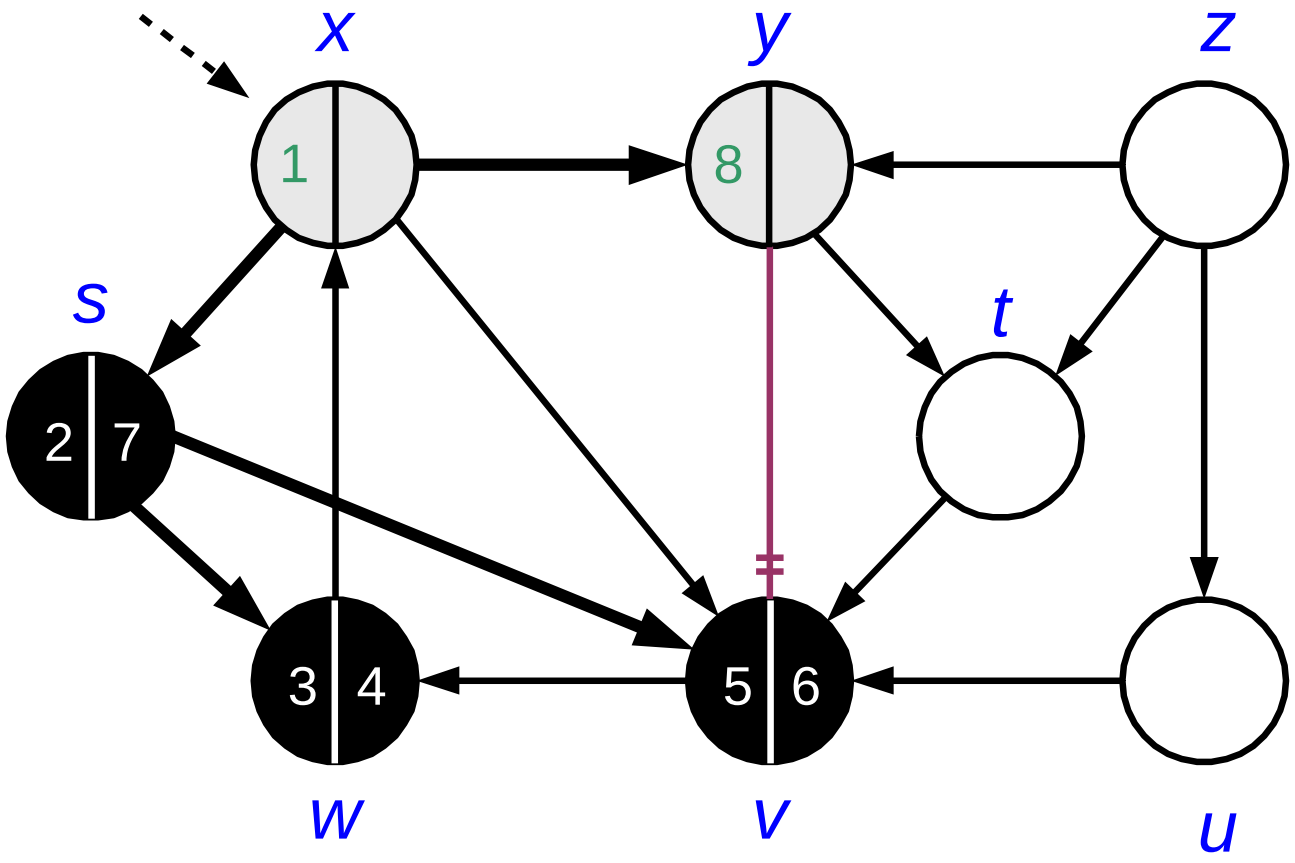
Depth-First Search : Example



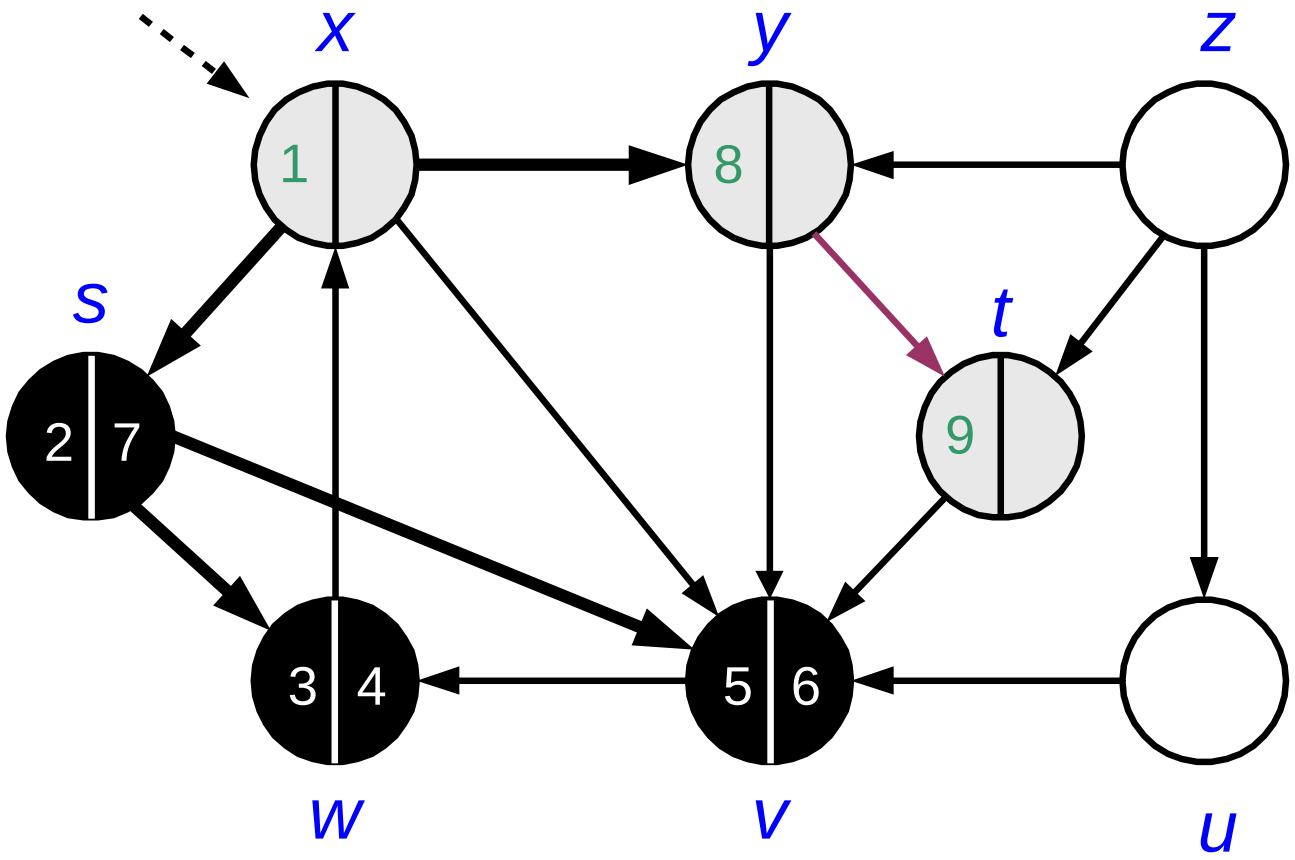
Depth-First Search : Example



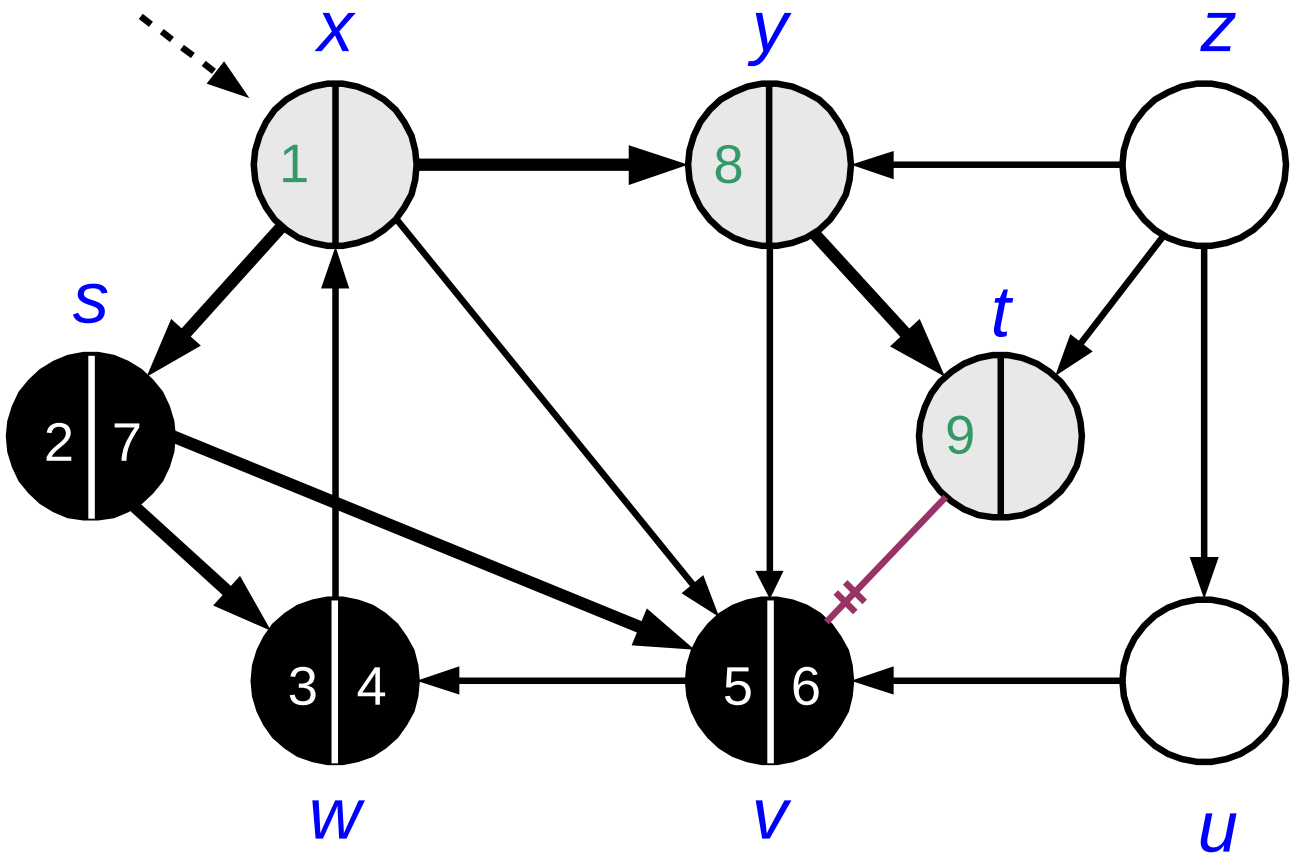
Depth-First Search : Example



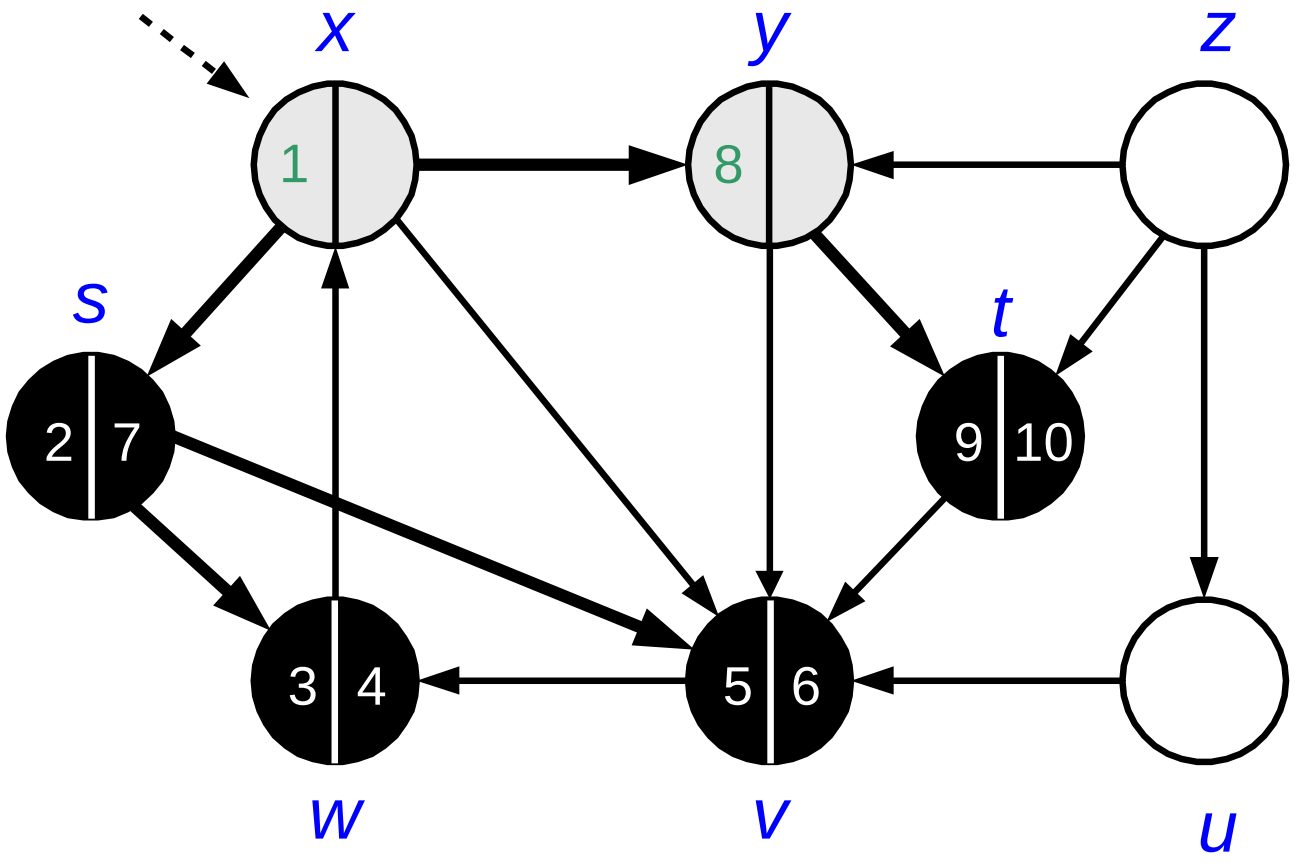
Depth-First Search : Example



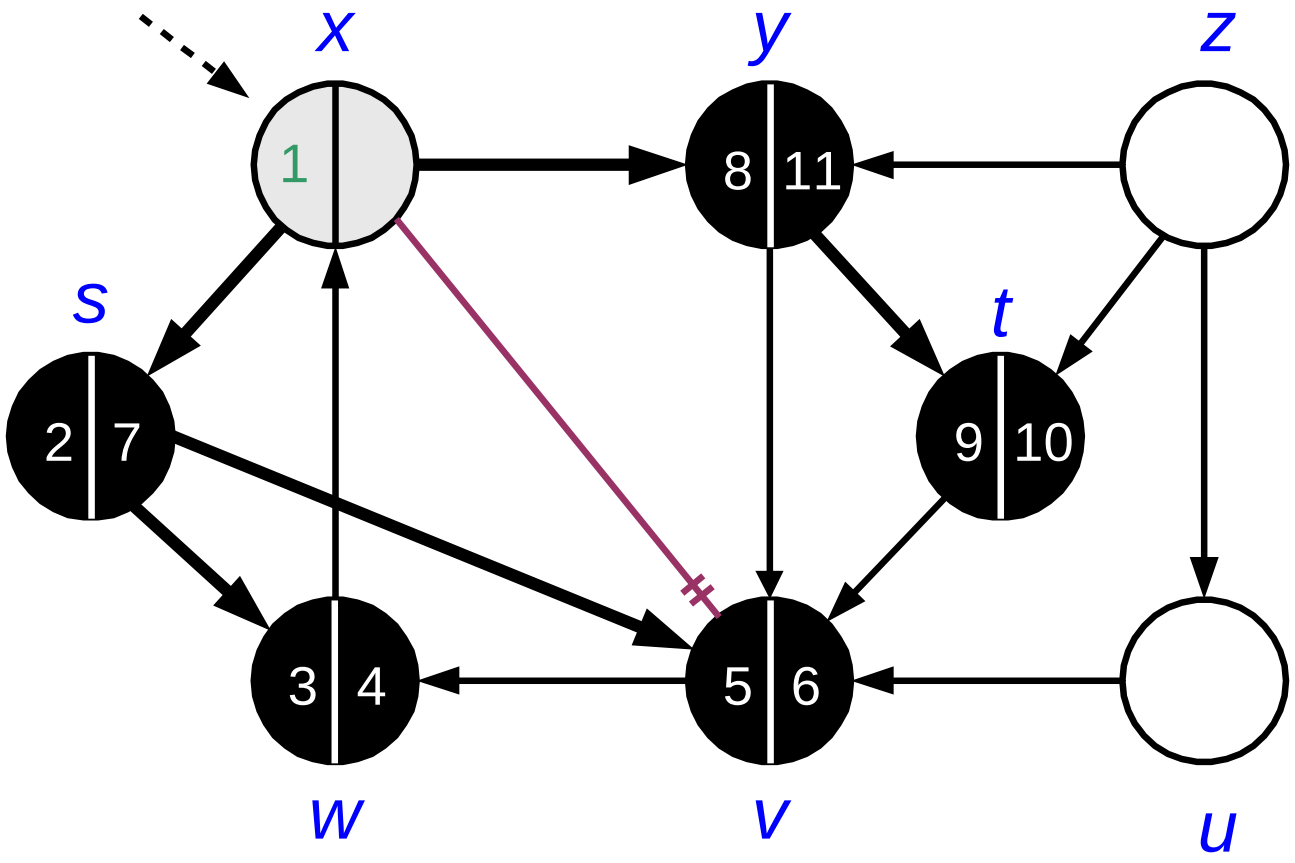
Depth-First Search : Example



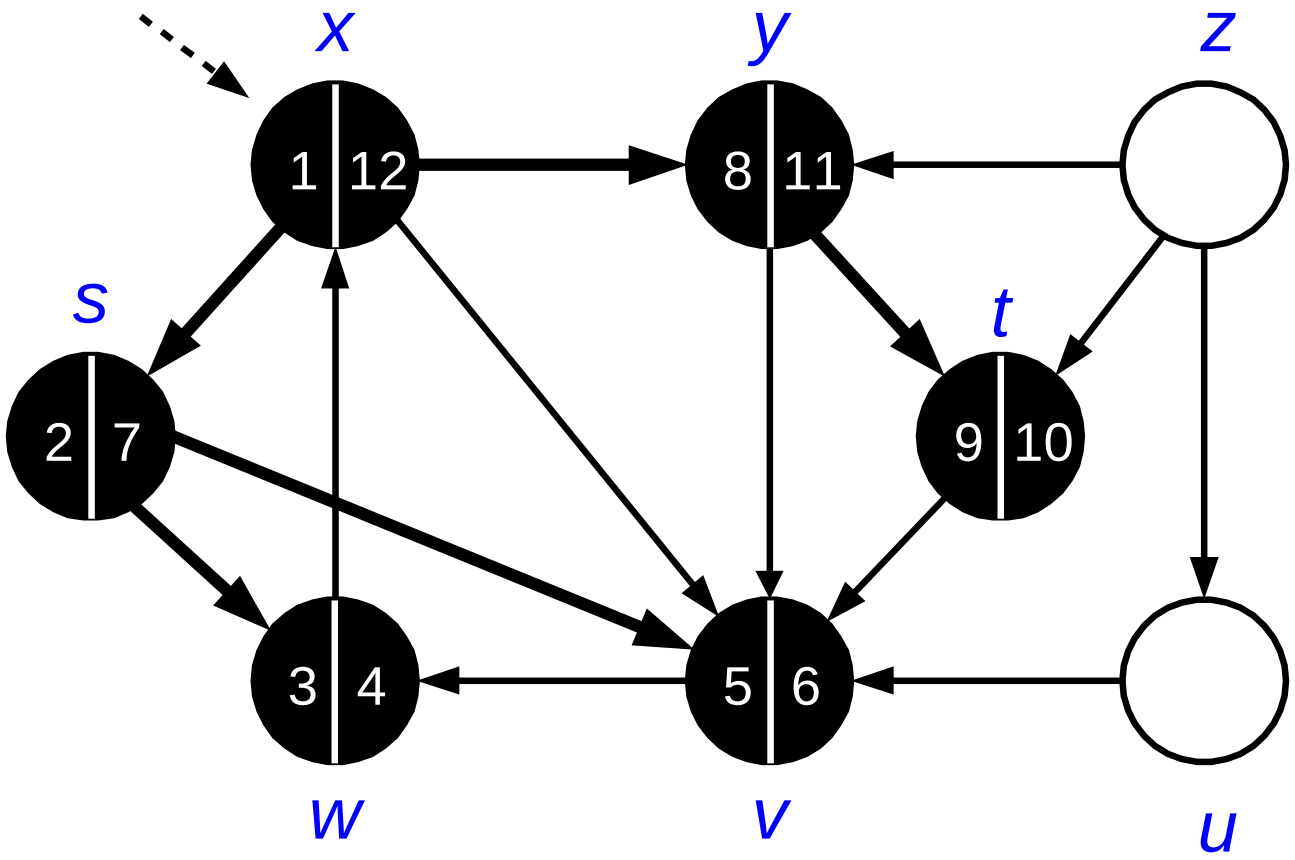
Depth-First Search : Example



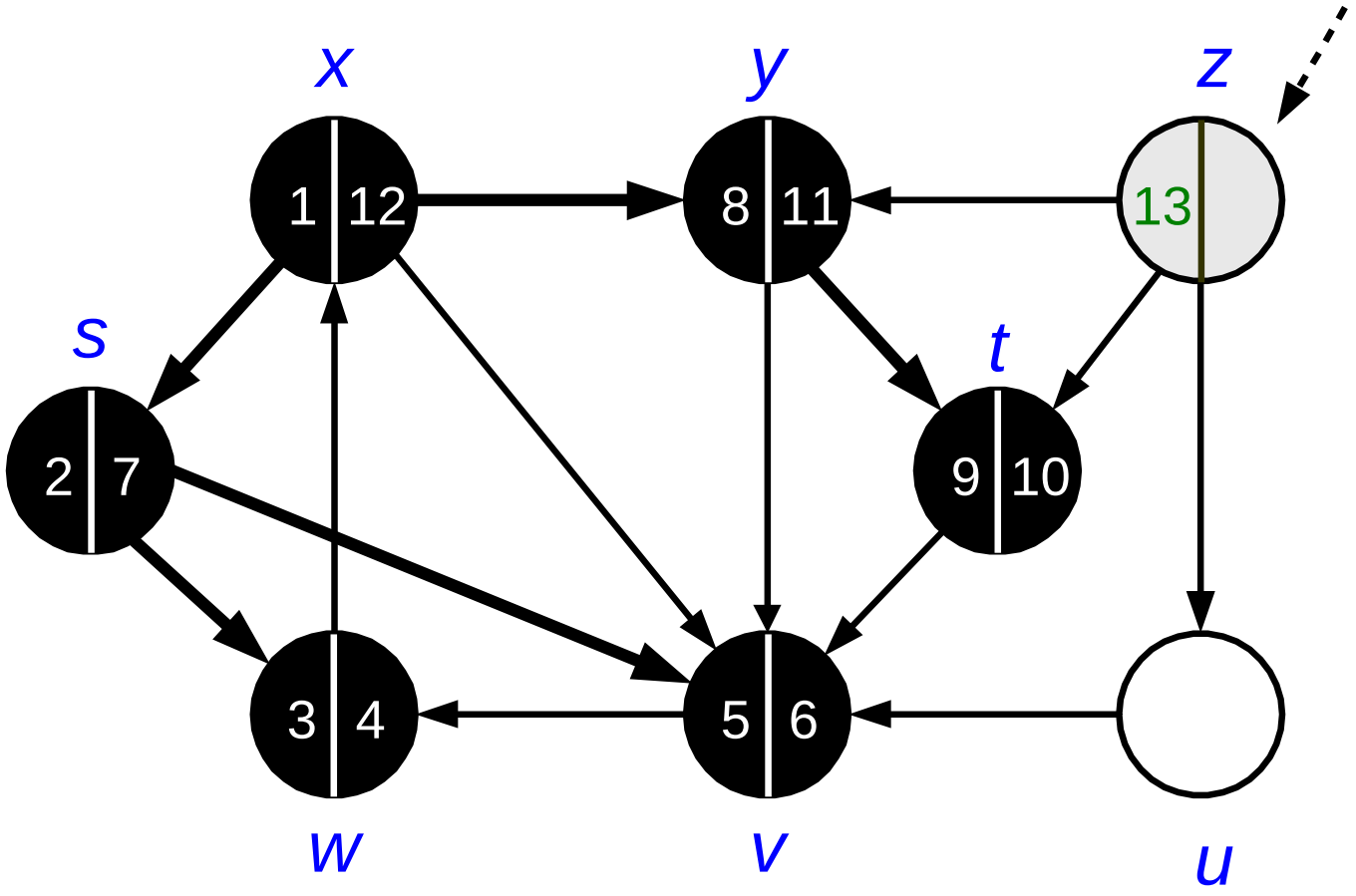
Depth-First Search : Example



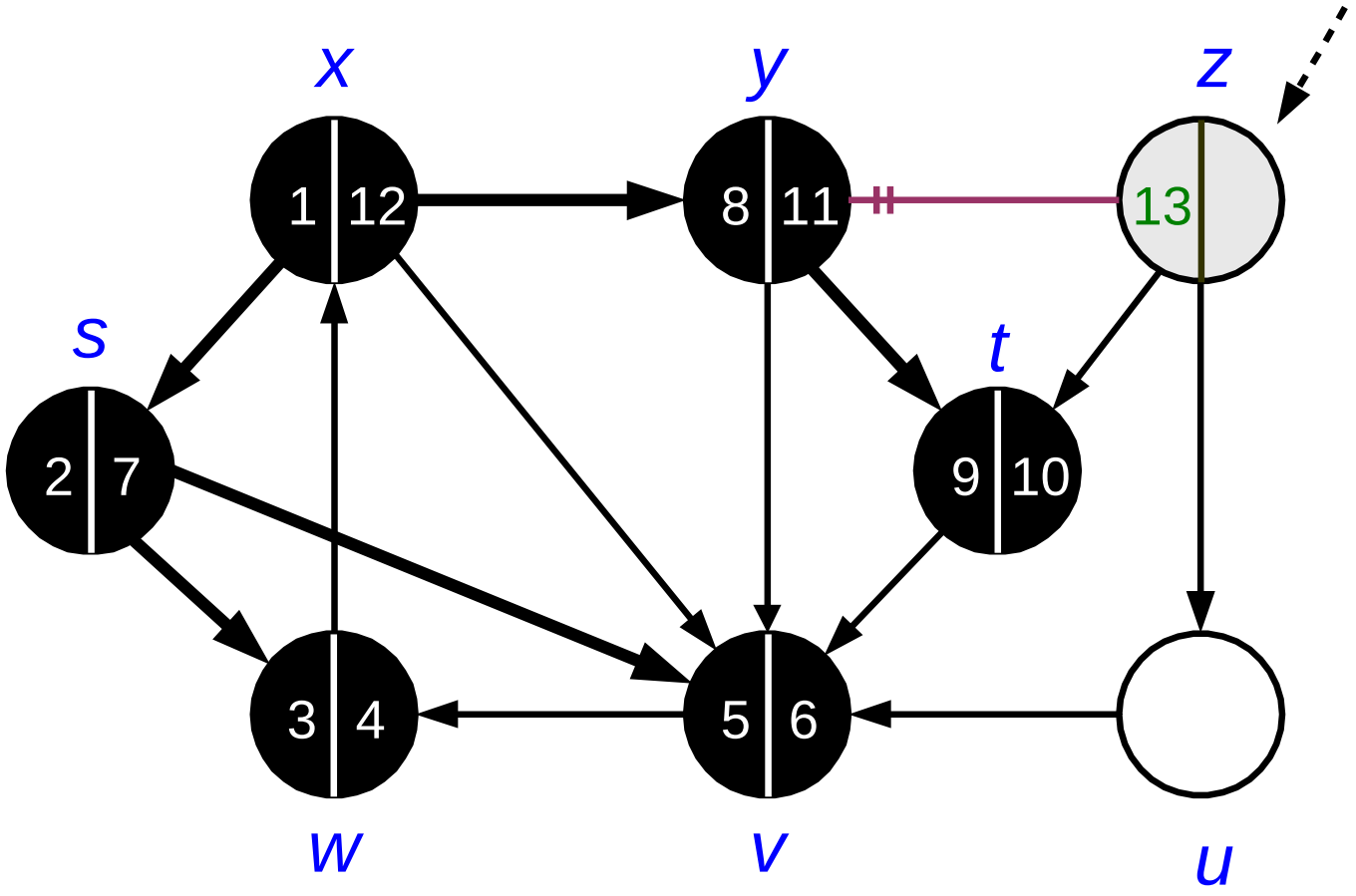
Depth-First Search : Example



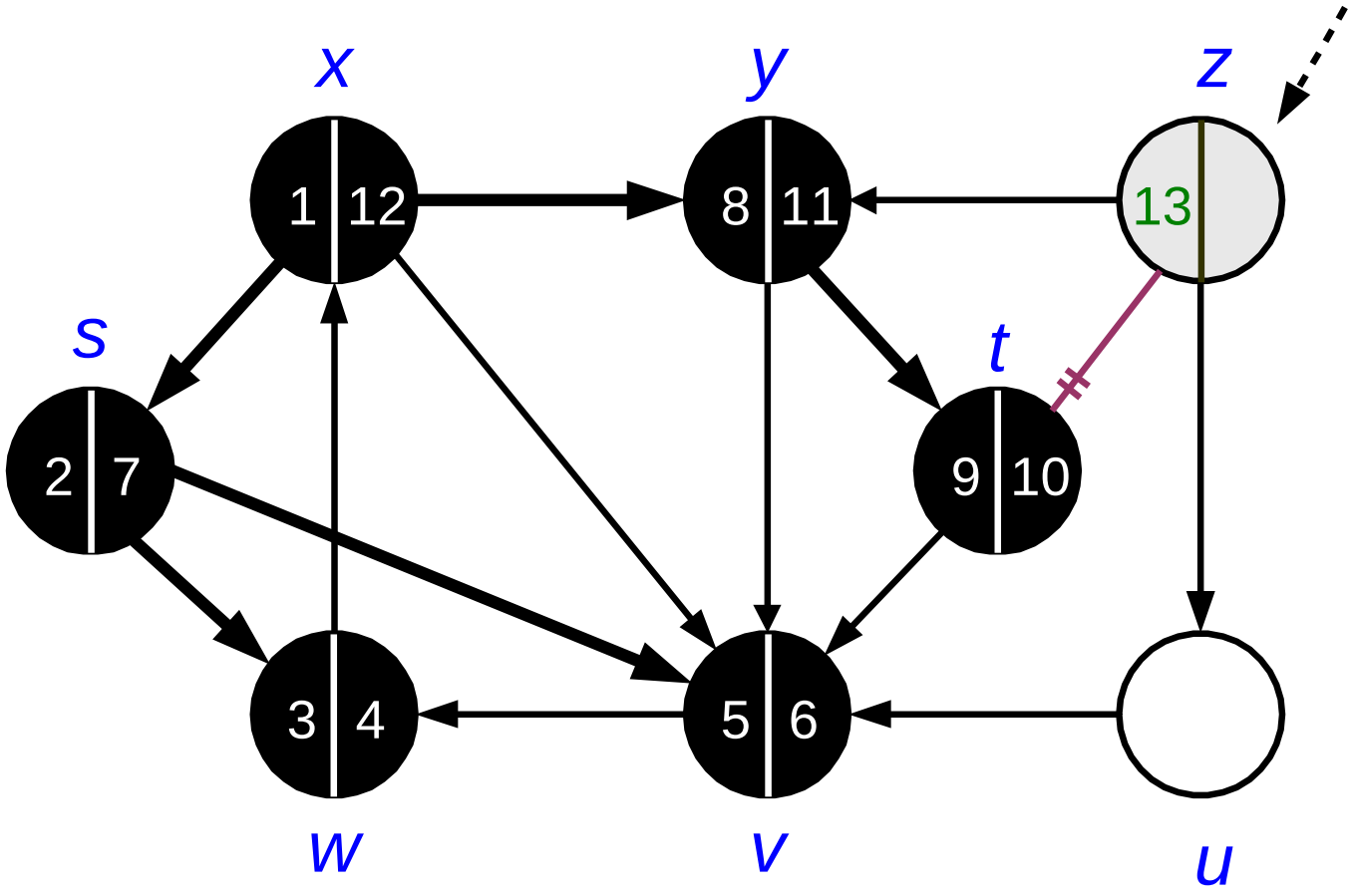
Depth-First Search : Example



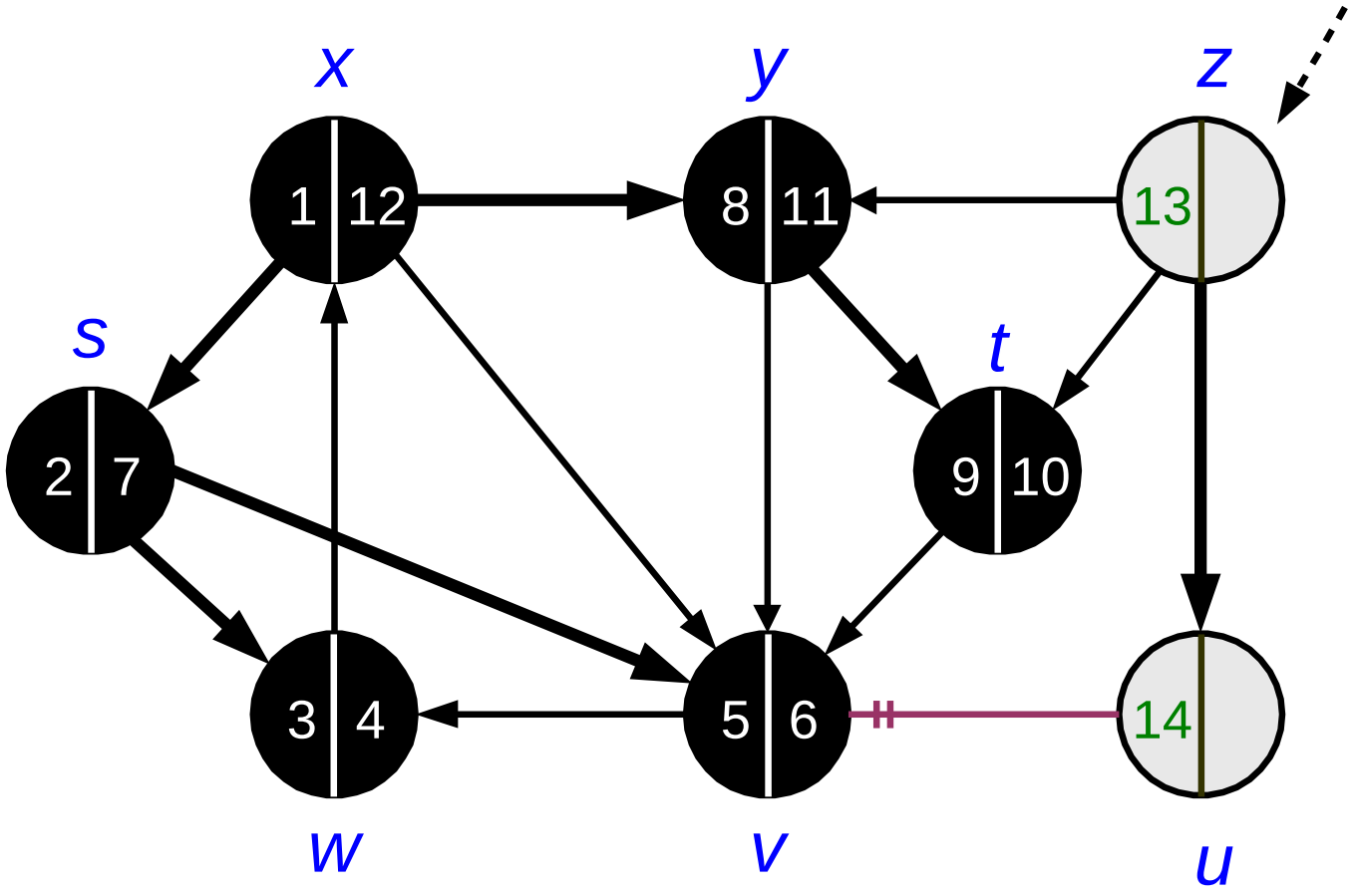
Depth-First Search : Example



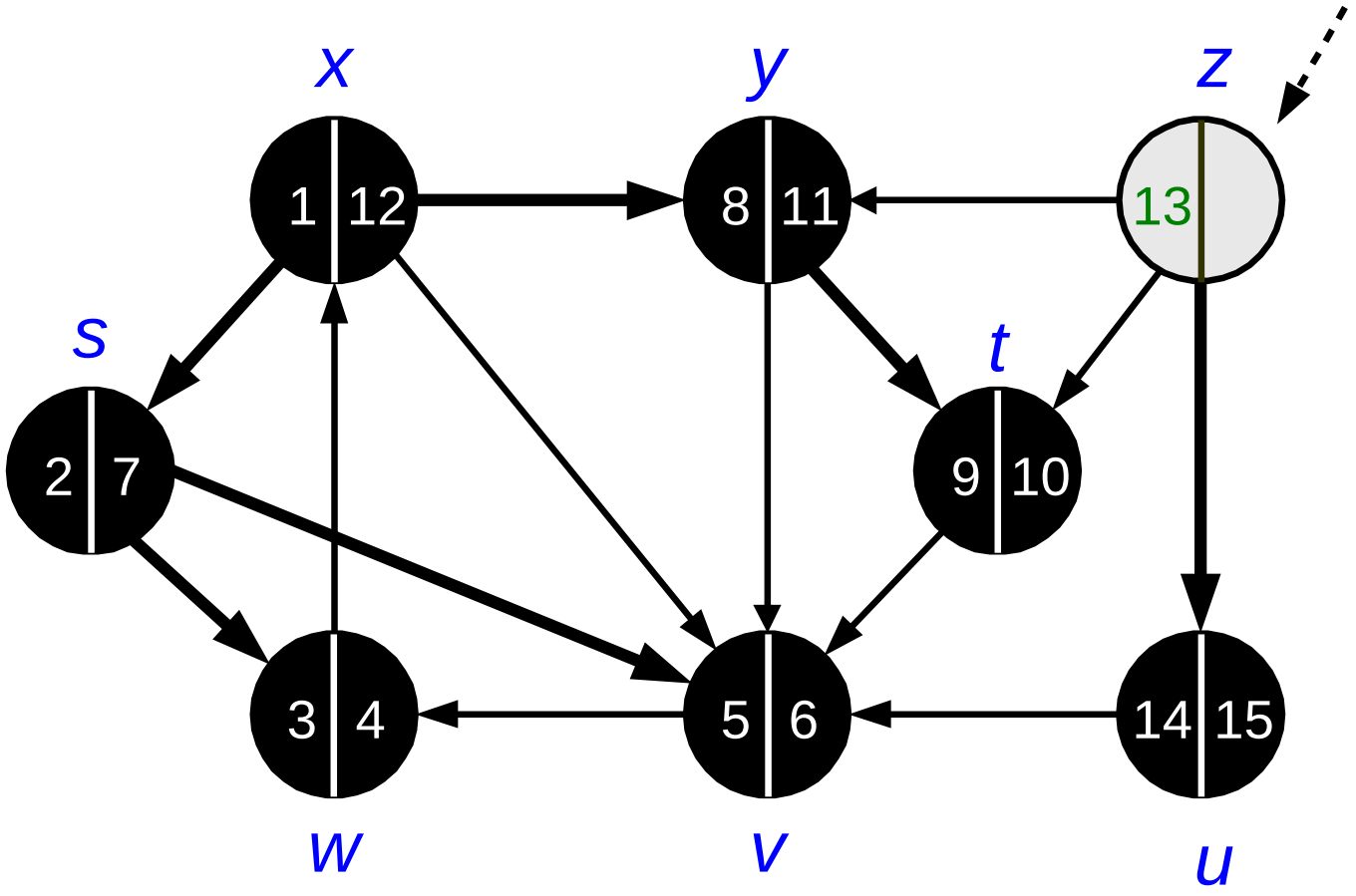
Depth-First Search : Example



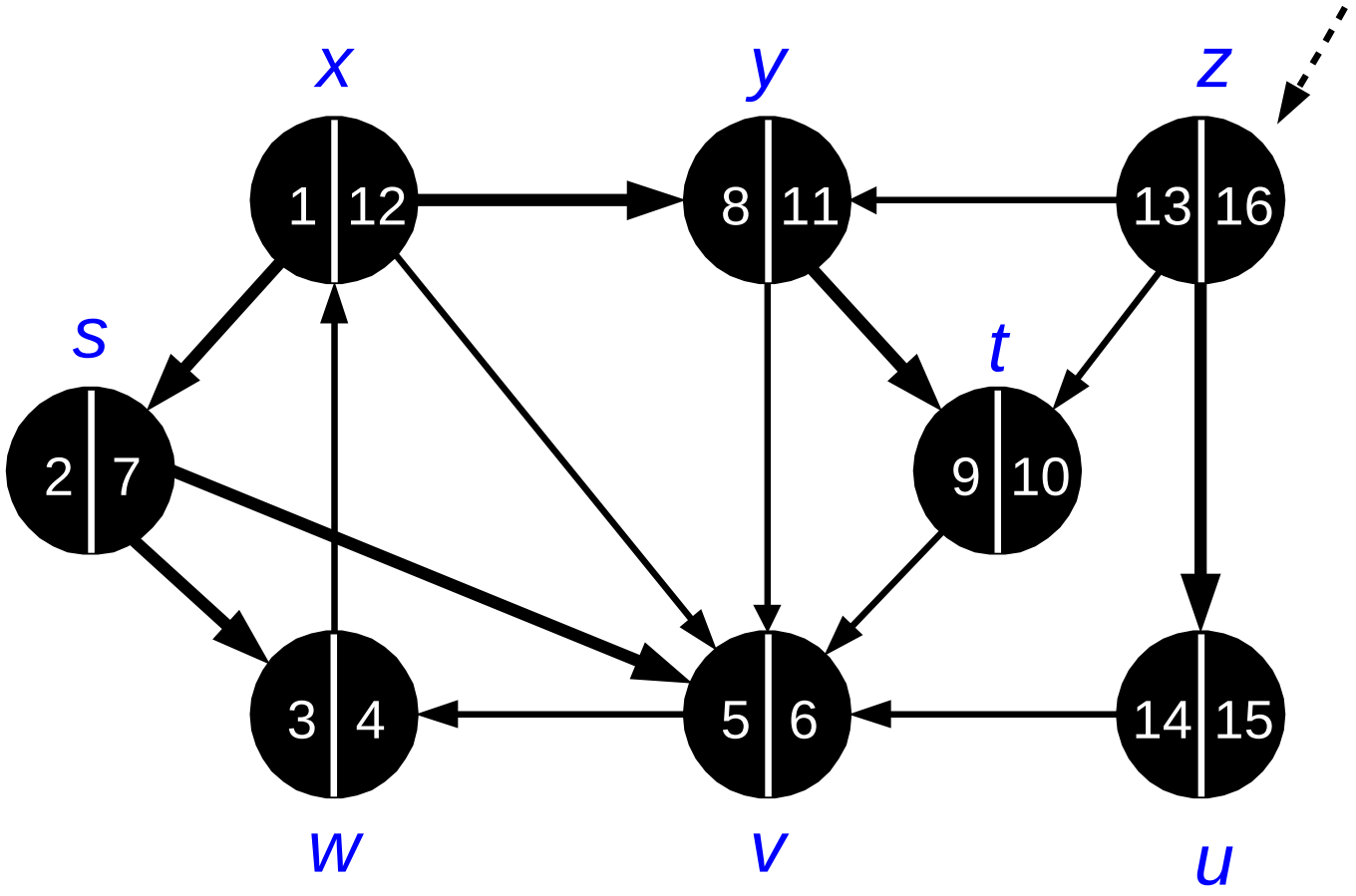
Depth-First Search : Example



Depth-First Search : Example

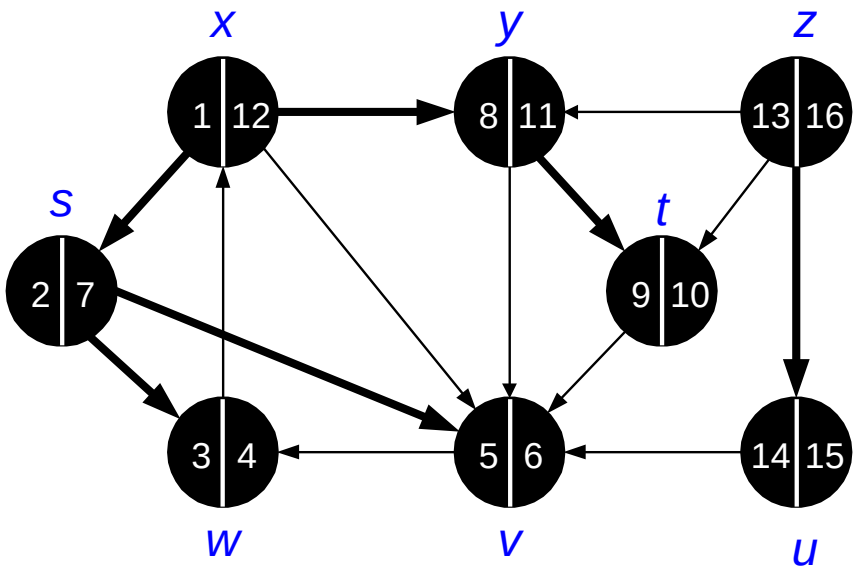


Depth-First Search : Example

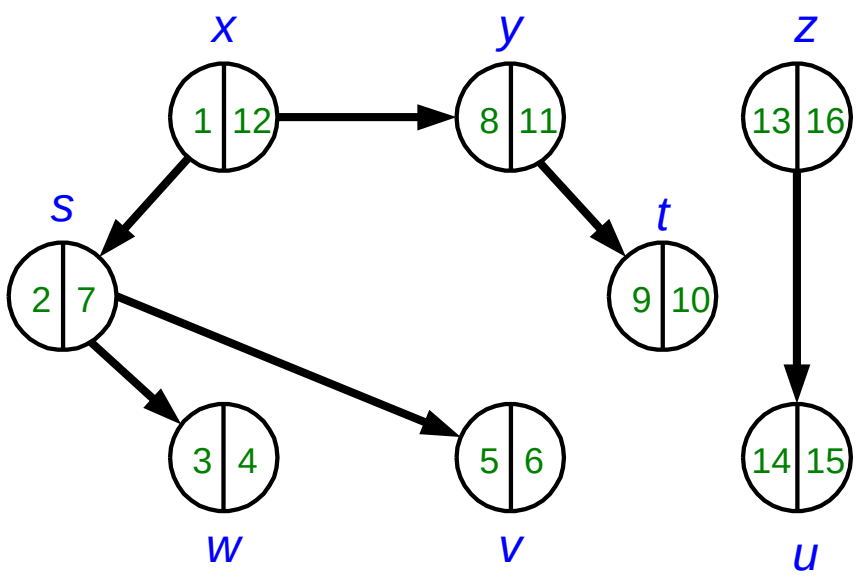


Depth-First Search : Example

Terminated DFS



Depth First Forest DFF



Breadth-First Search : Algorithm

BFS(G)

for each $u \in V$ **do**

state[u]$\leftarrow$ *un-visited*

put the start node u* into a queue Q

While Q is !empty

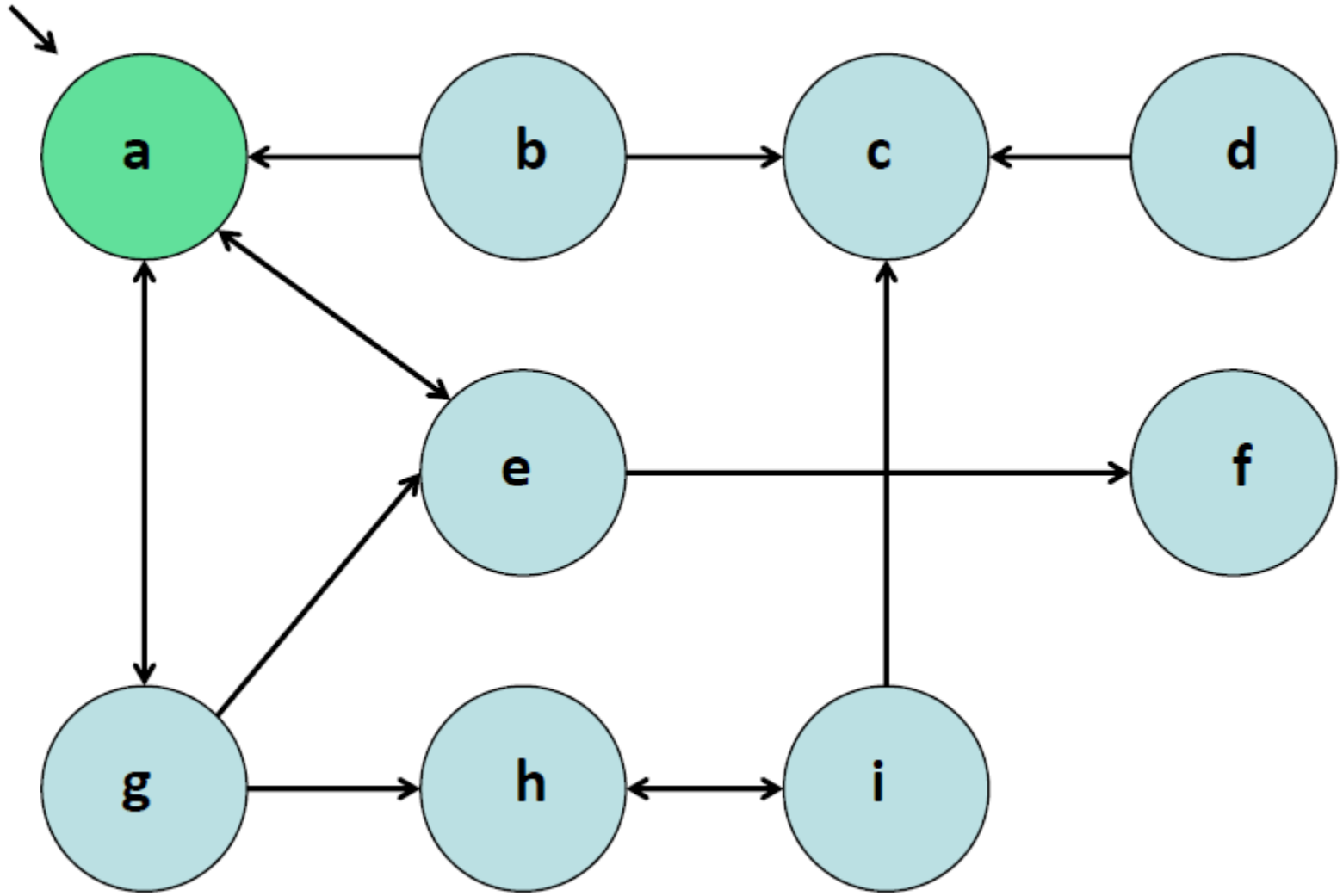
n = S.pop()

if state[n] == un-visited

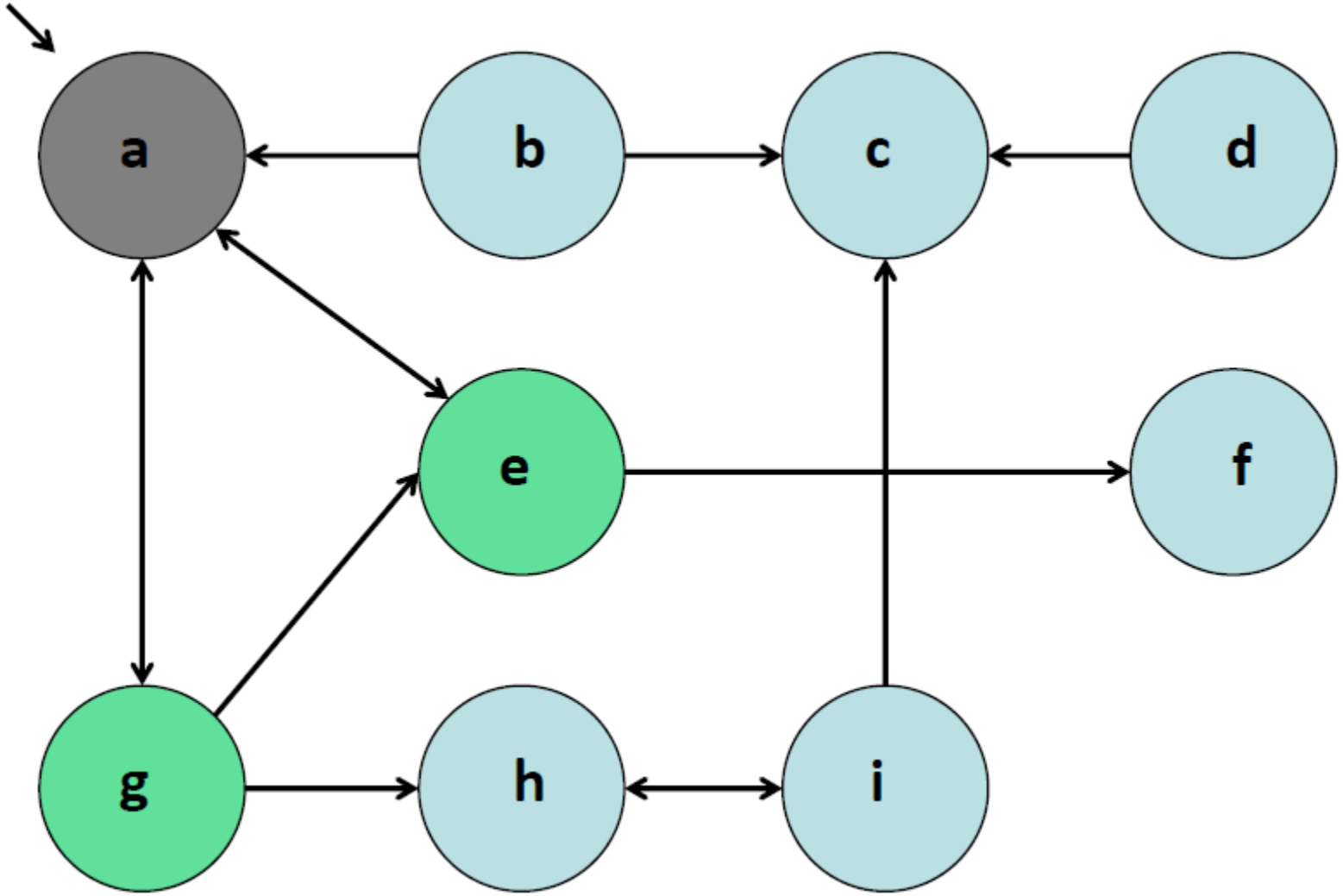
S.push(n.Adj)

state[n] = visited

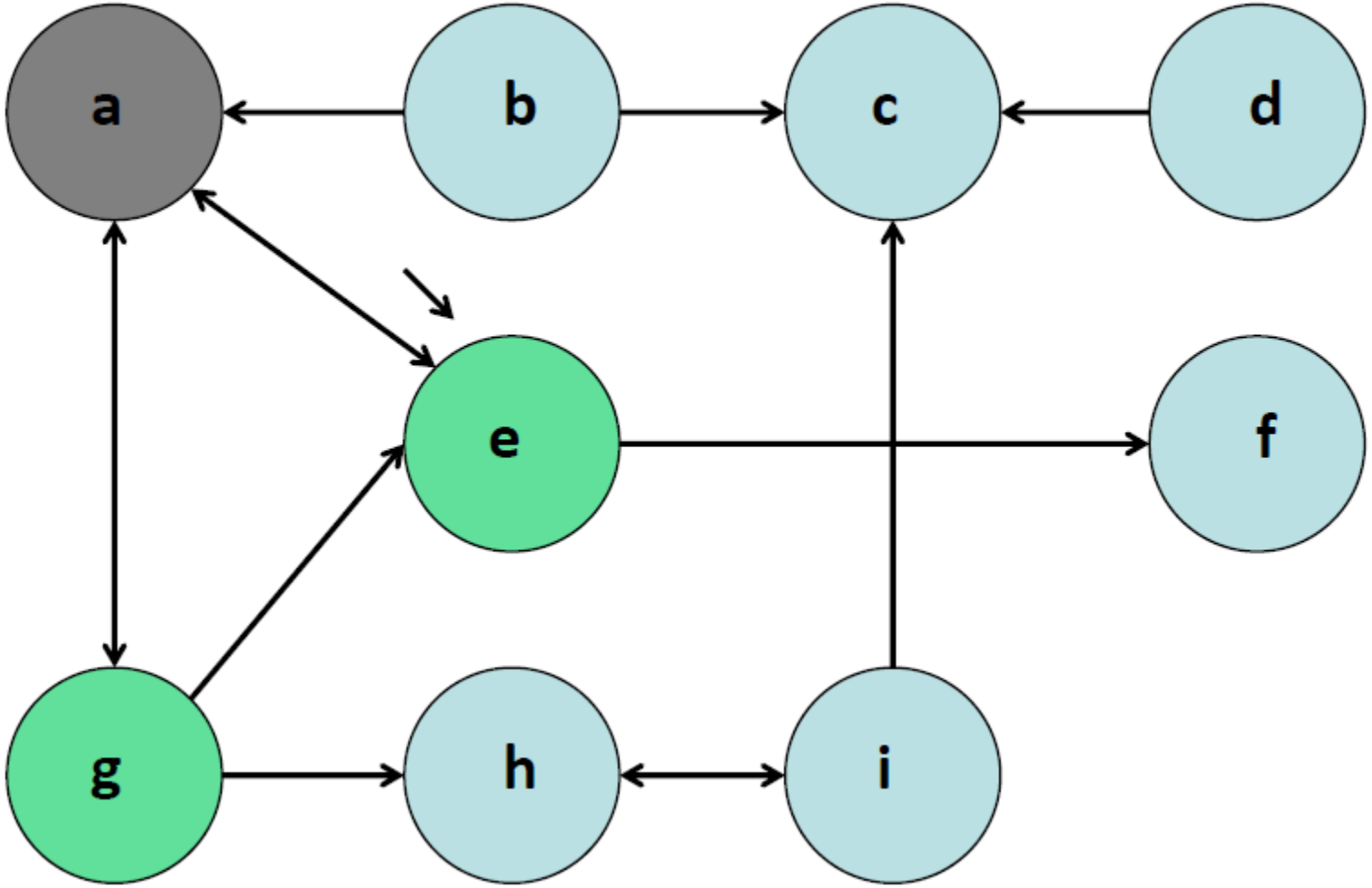
Breadth-First Search : Example



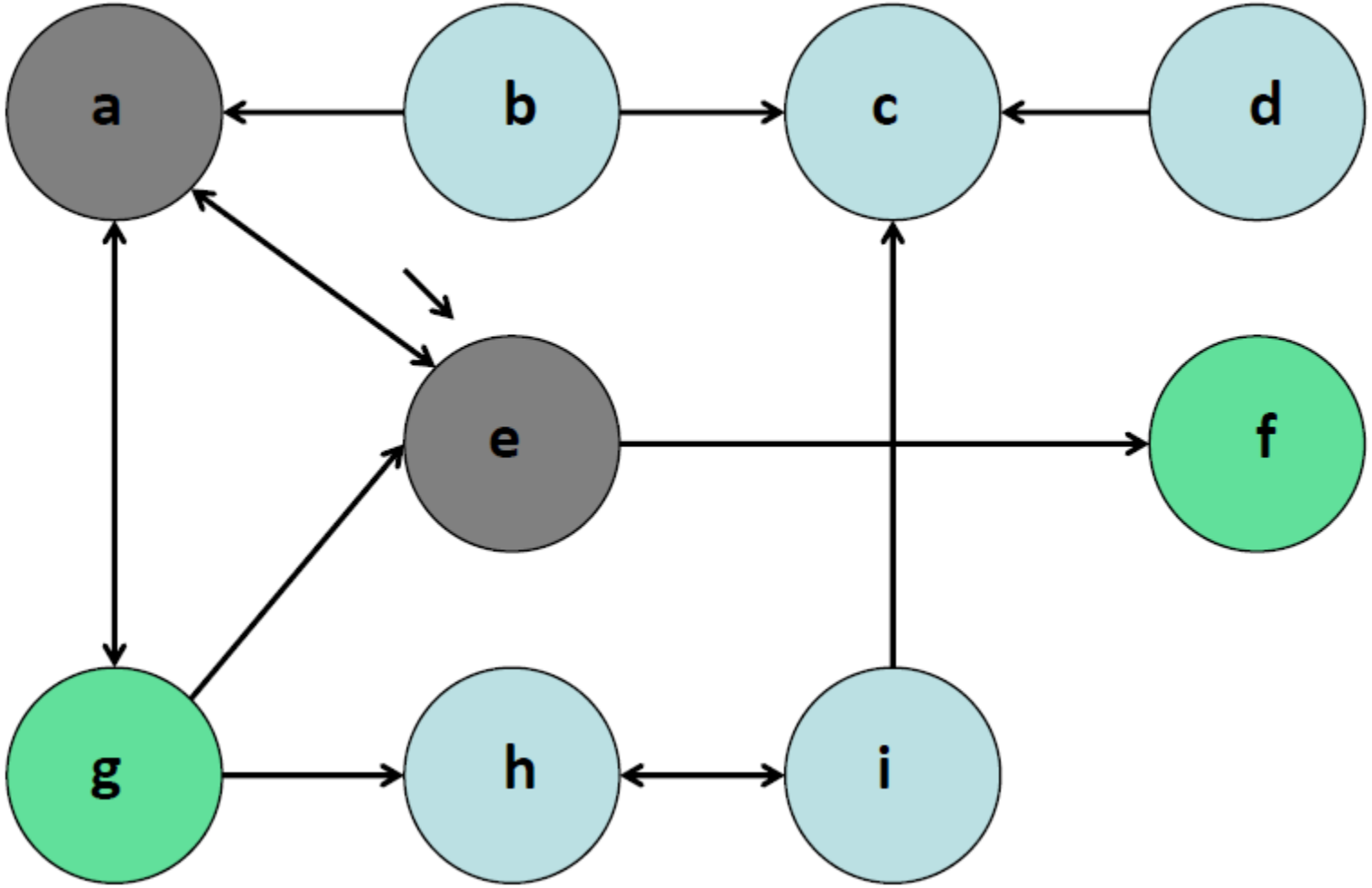
Breadth-First Search : Example



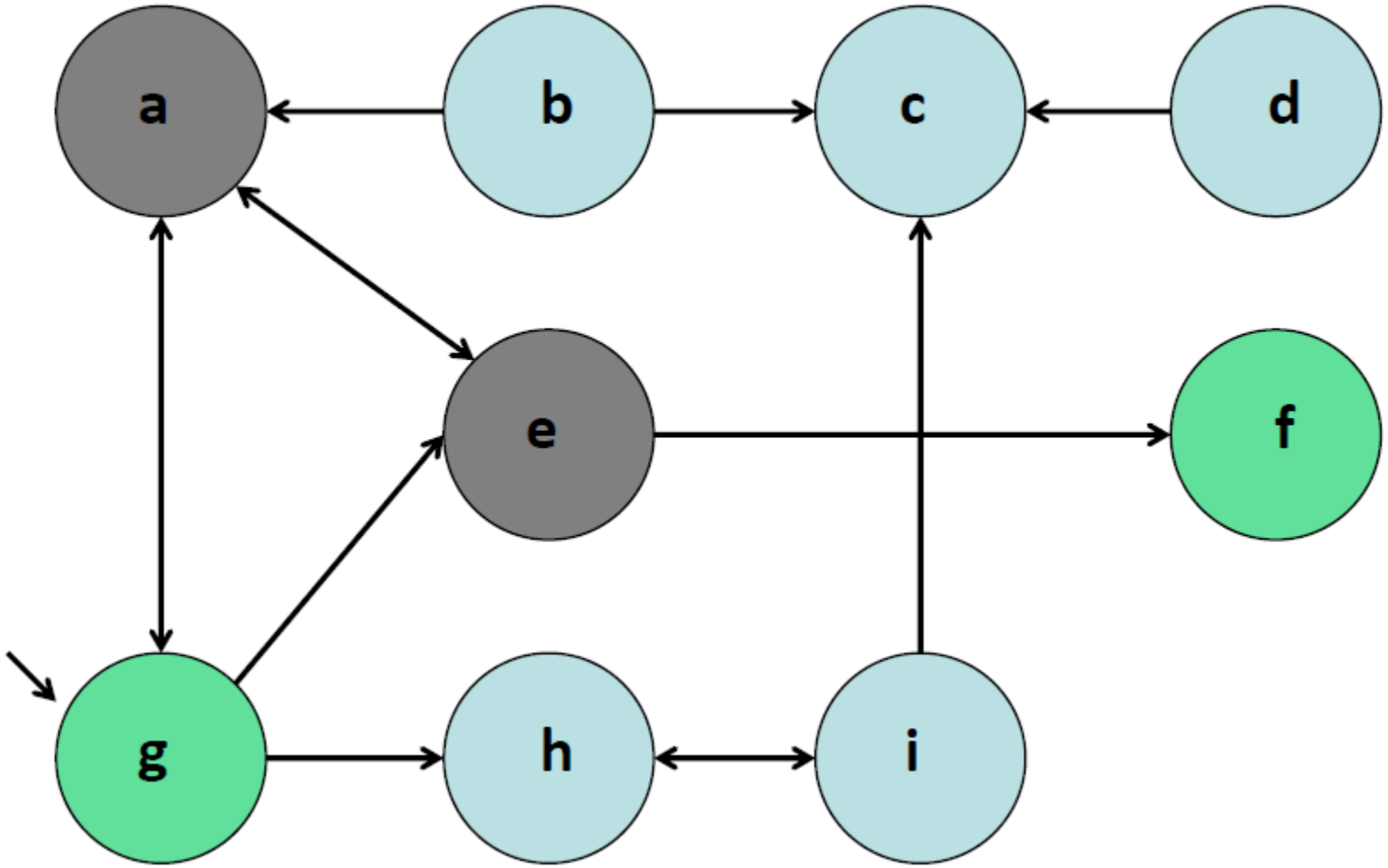
Breadth-First Search : Example



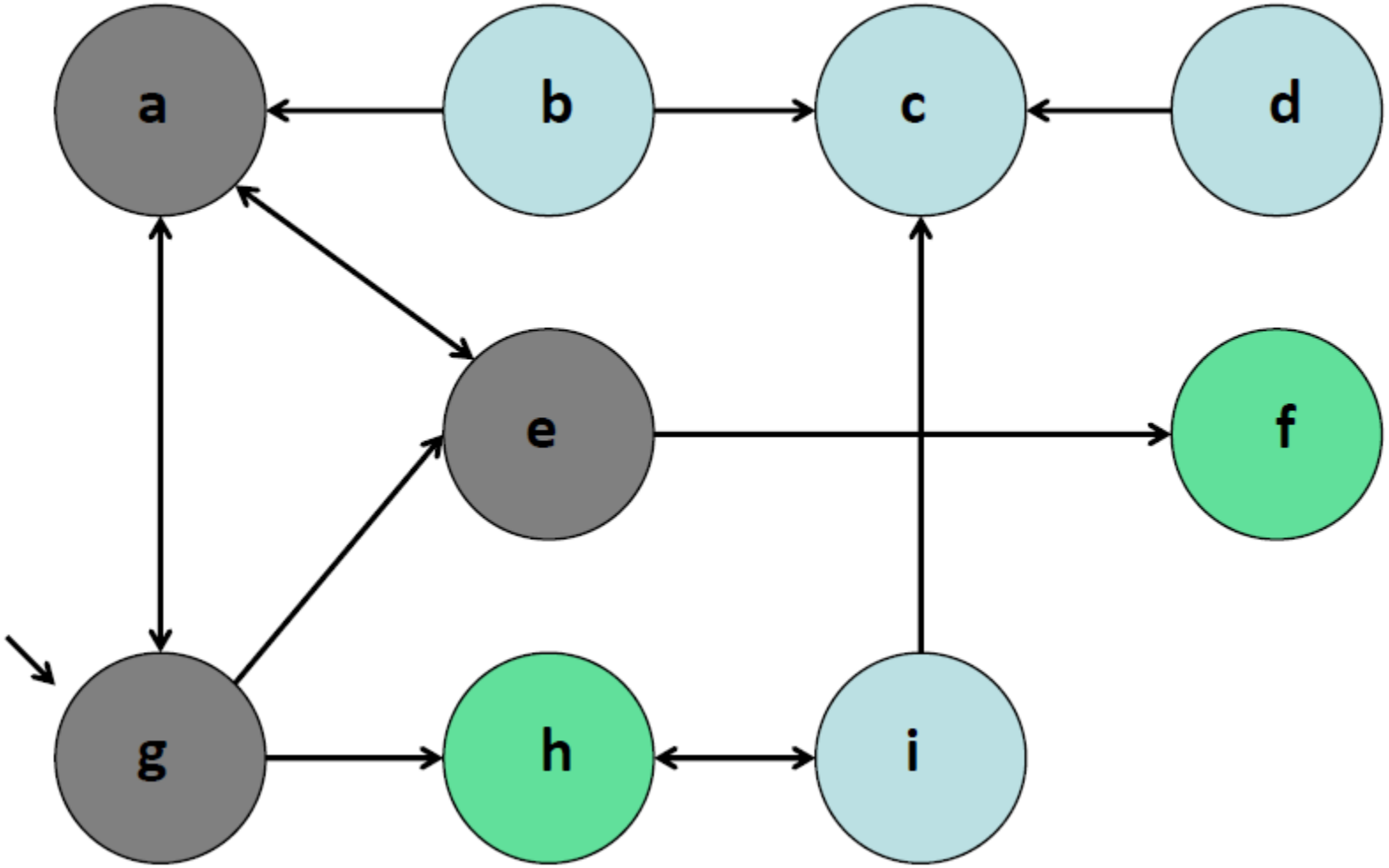
Breadth-First Search : Example



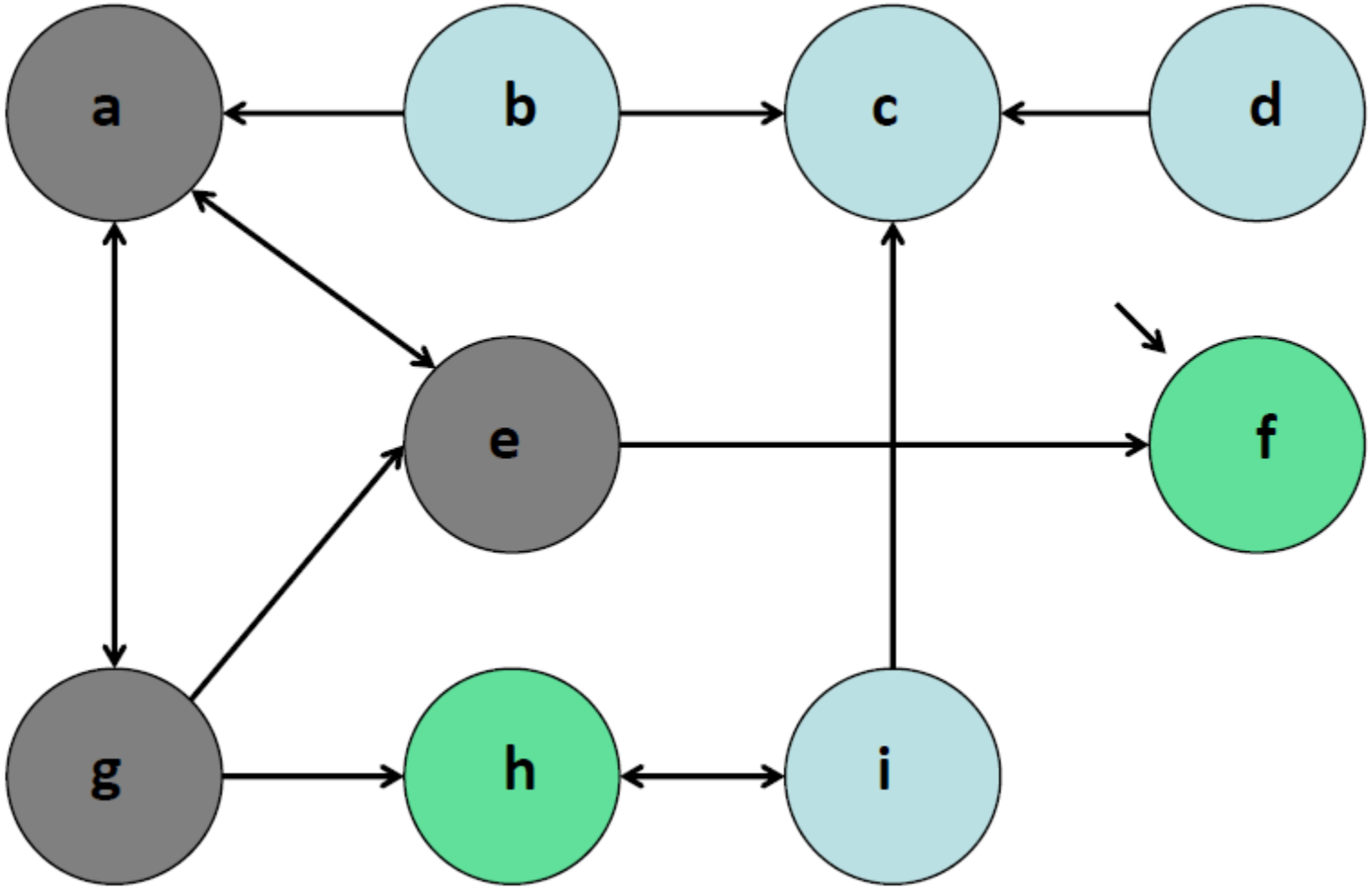
Breadth-First Search : Example



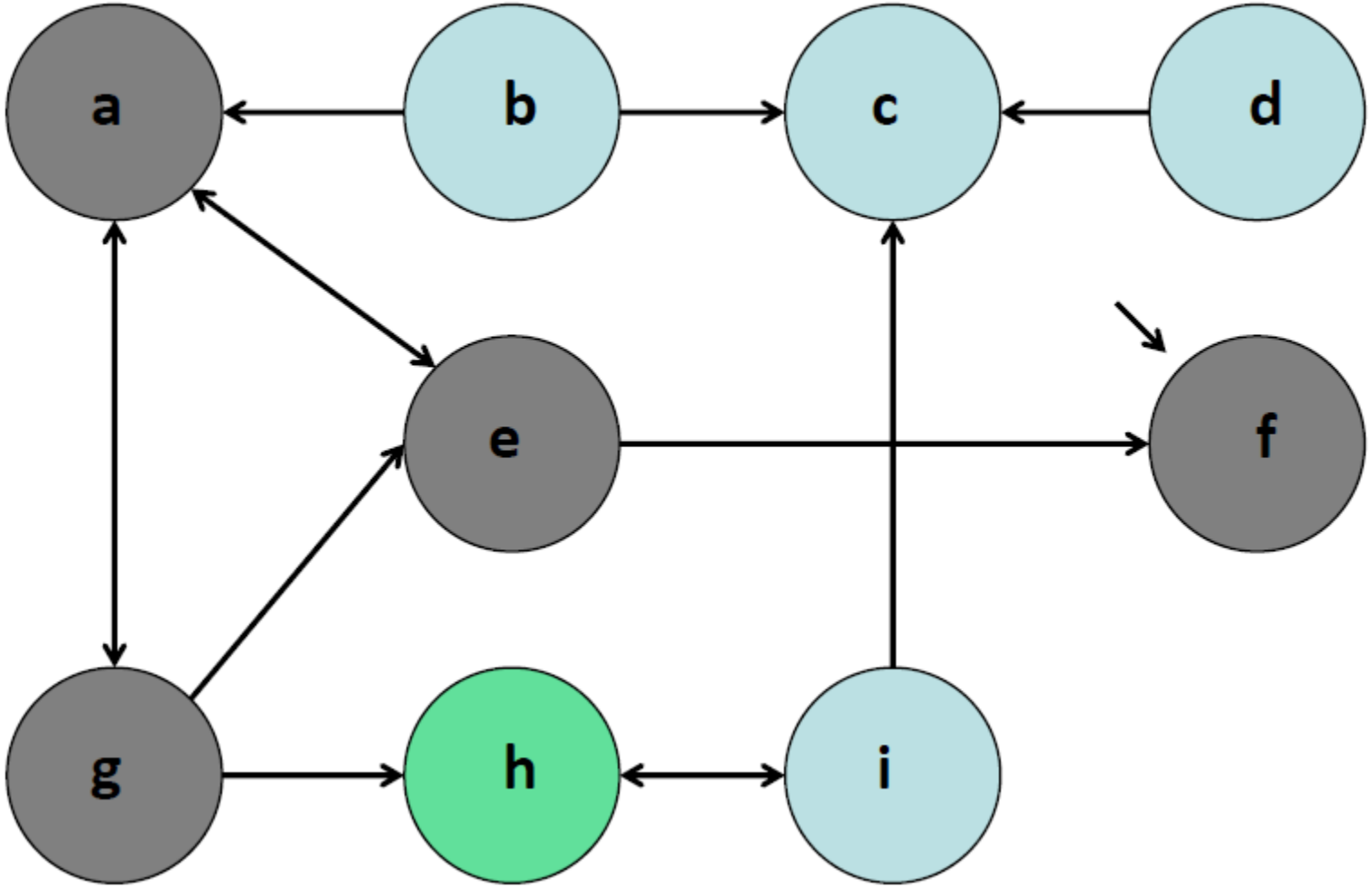
Breadth-First Search : Example



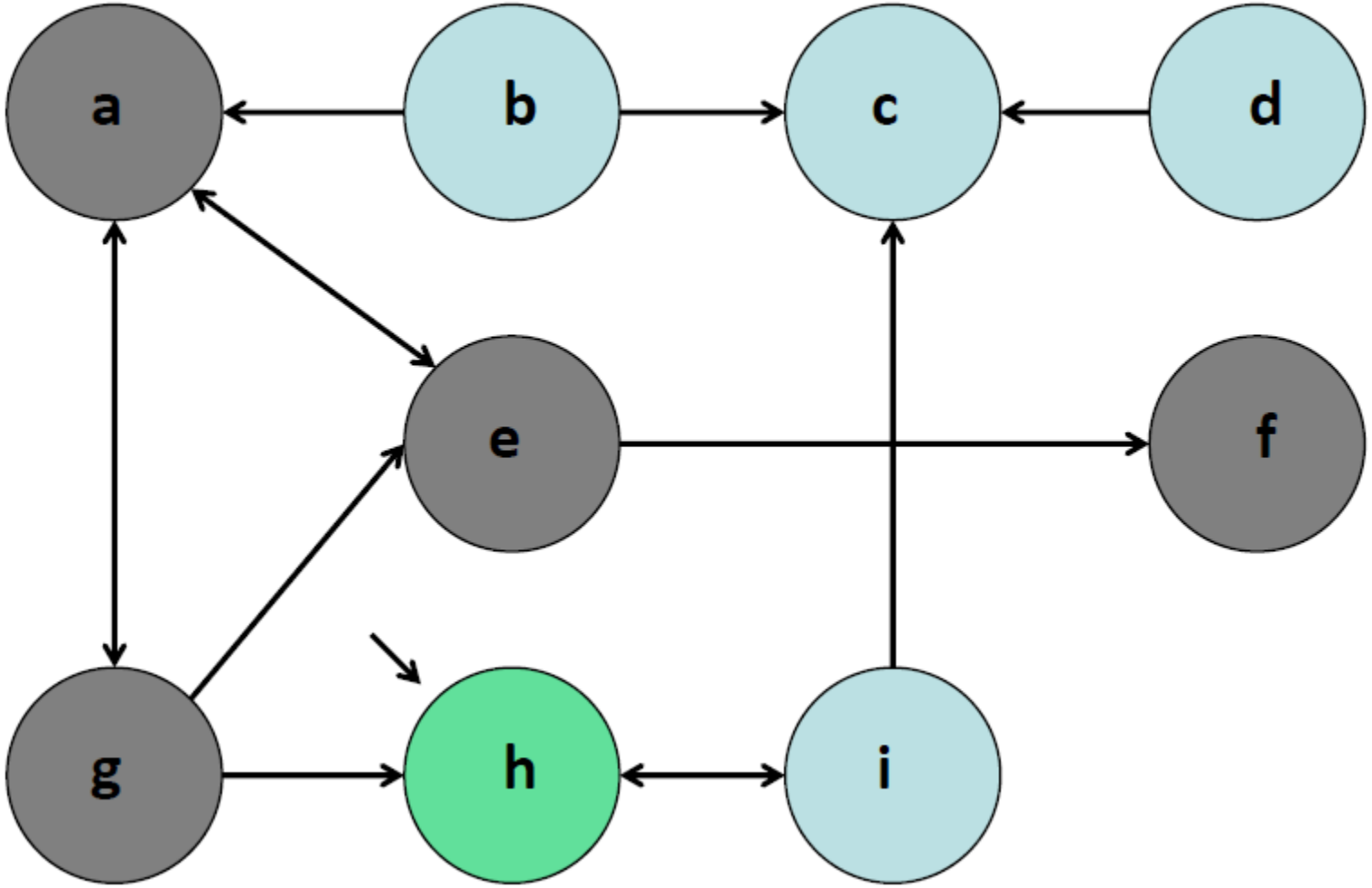
Breadth-First Search : Example



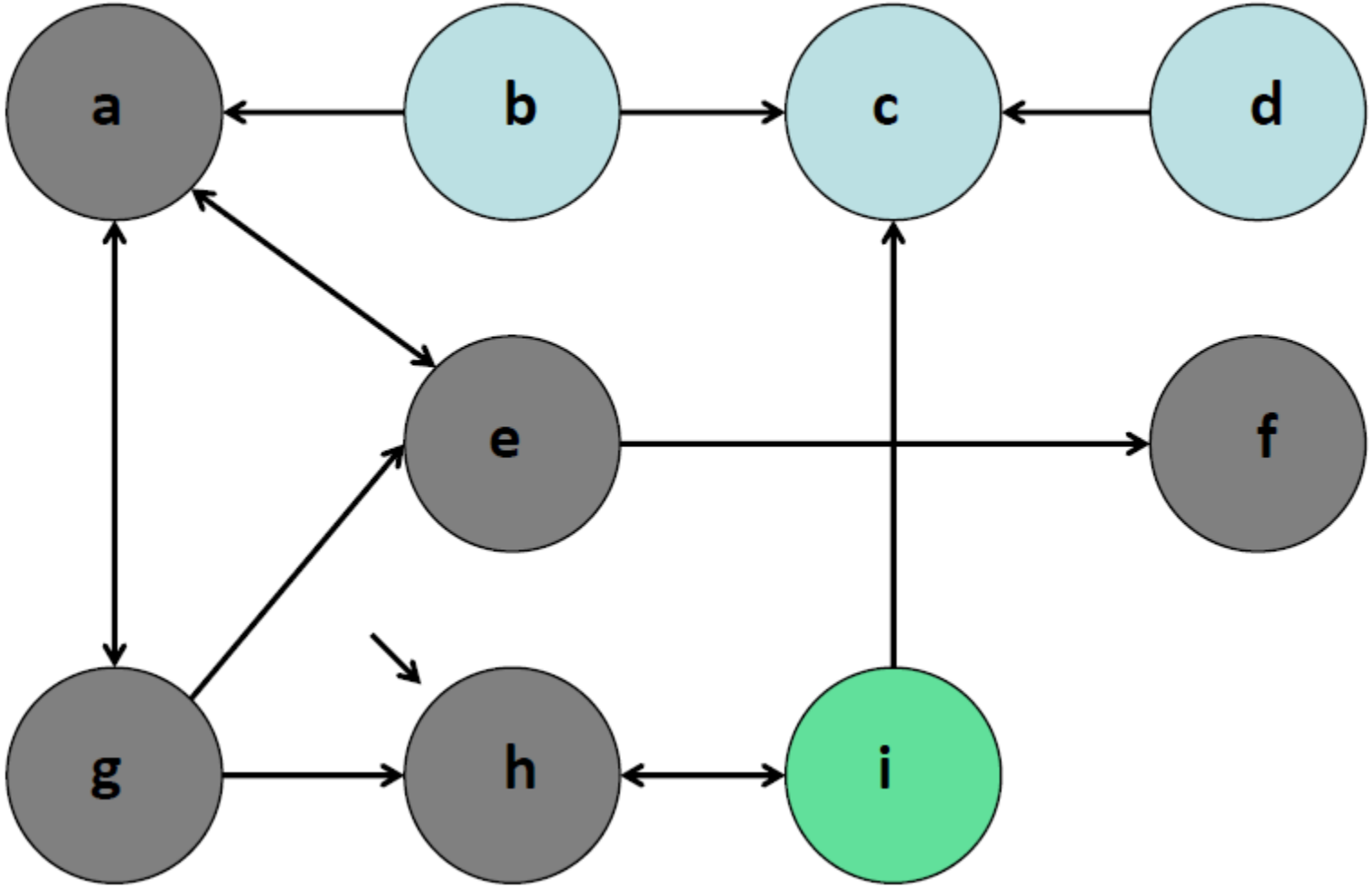
Breadth-First Search : Example



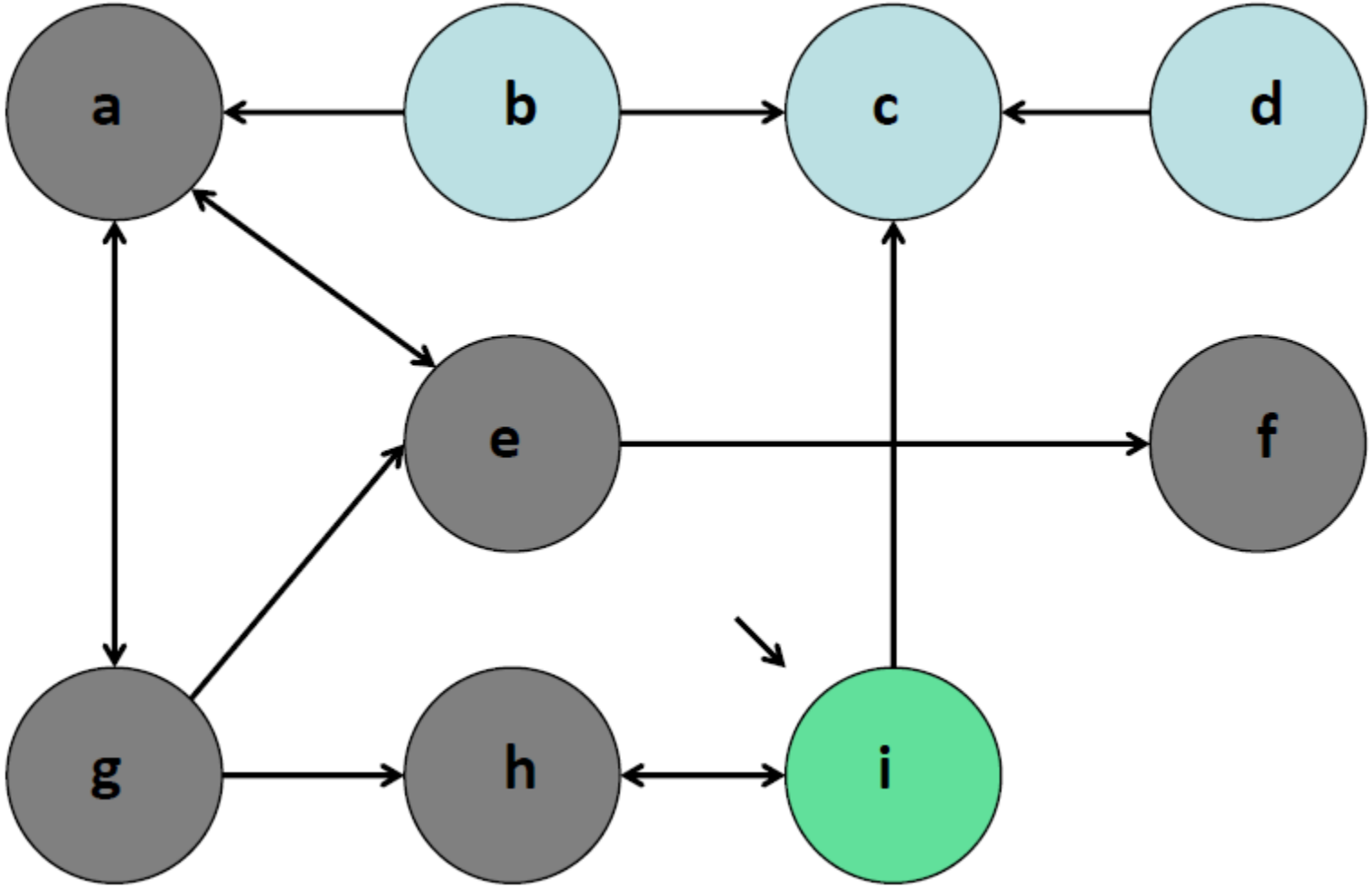
Breadth-First Search : Example



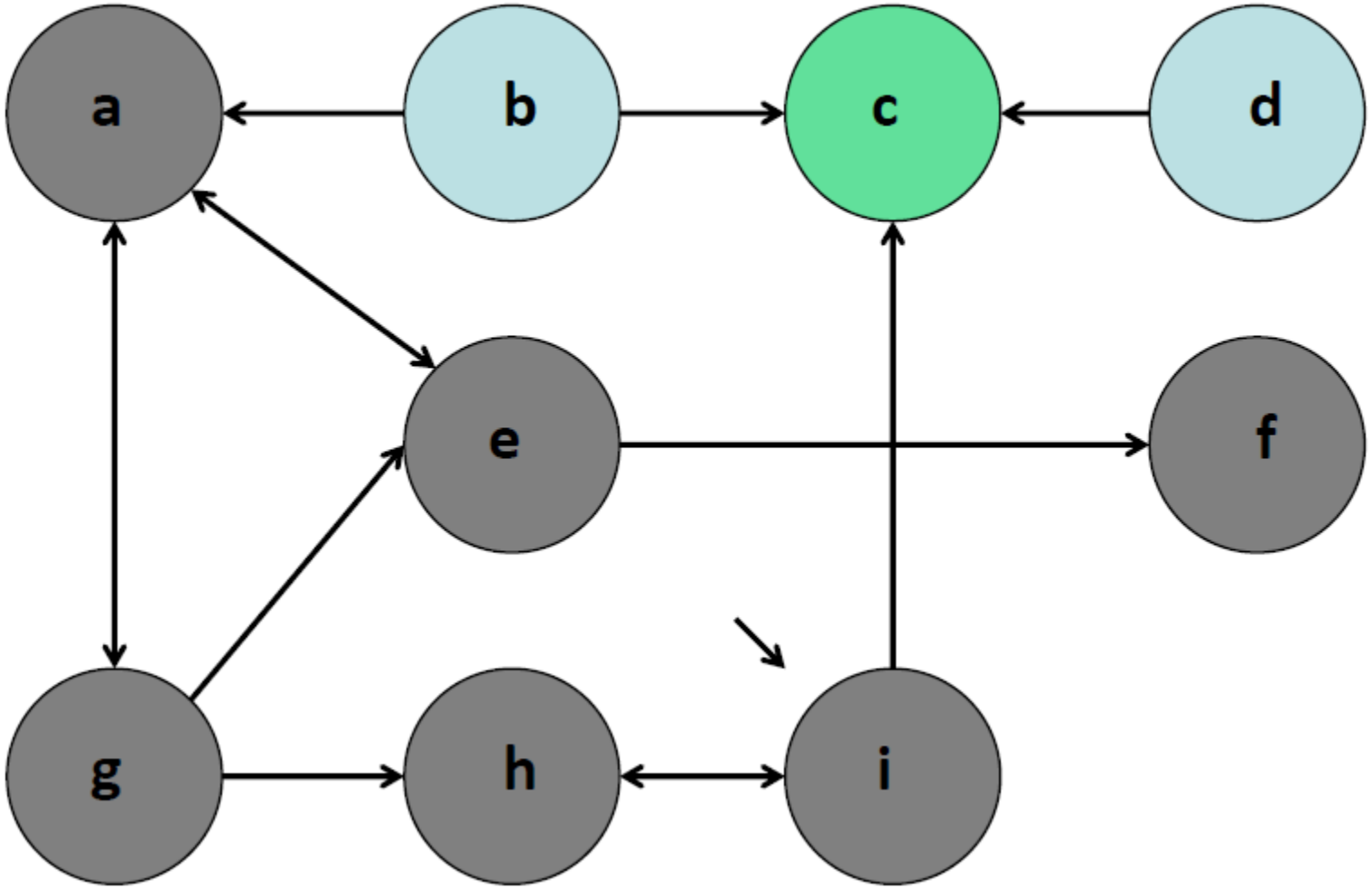
Breadth-First Search : Example



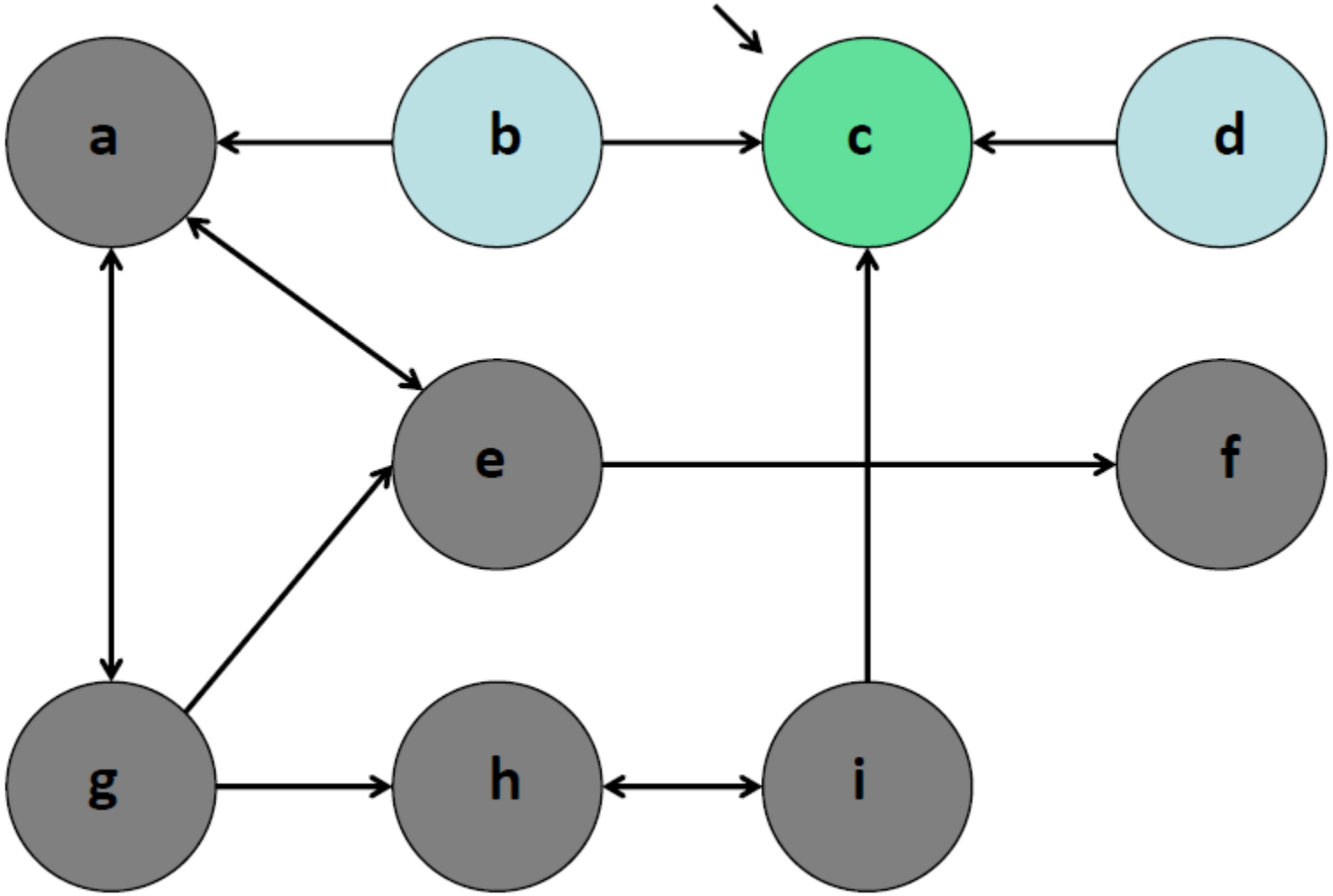
Breadth-First Search : Example



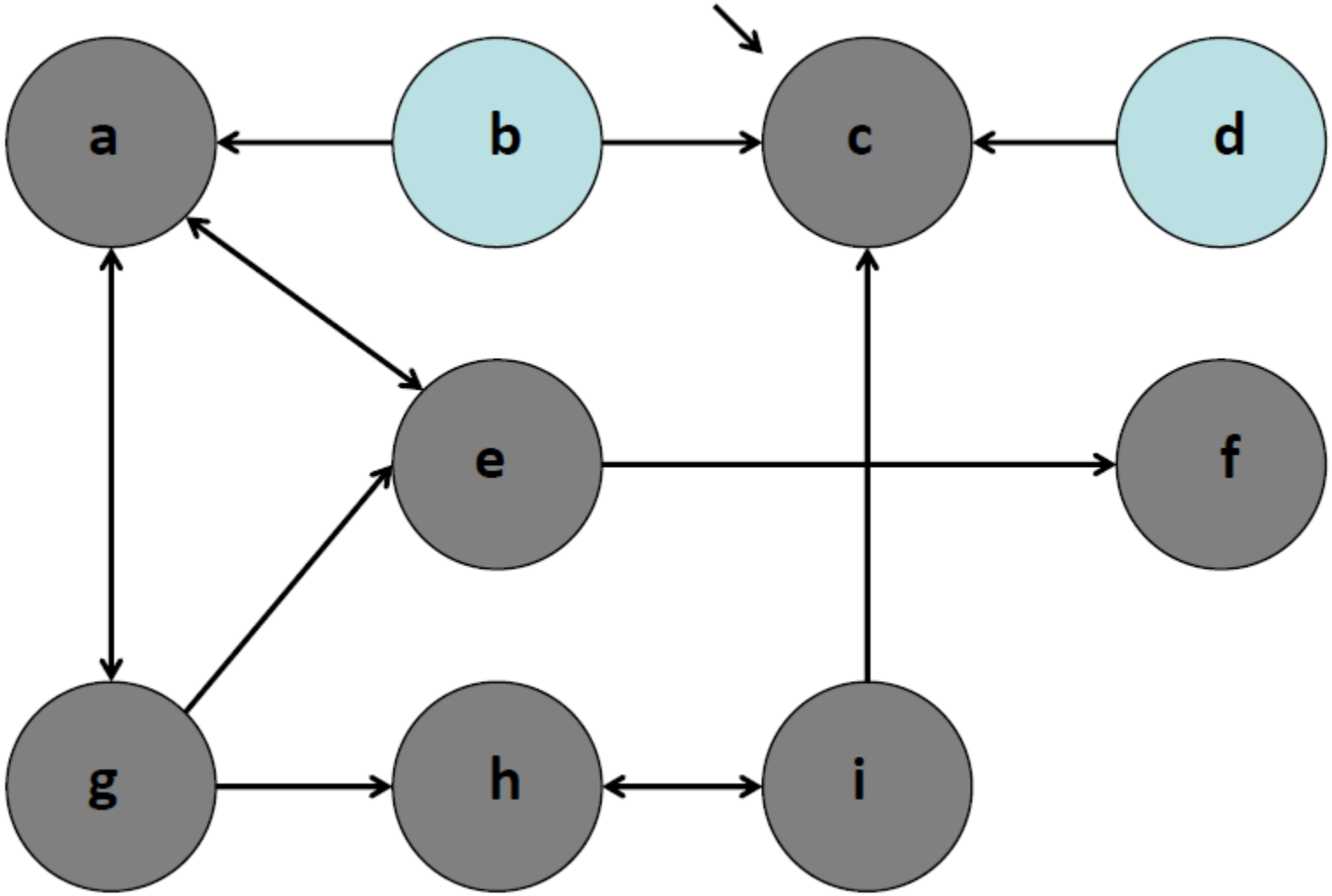
Breadth-First Search : Example



Breadth-First Search : Example



Breadth-First Search : Example



Comparison between DFS v.s BFS

DFS(G)

for each $u \in V$ **do**

state[u]← *un-visited*

put the start node u* into a stack S

While S is !empty

n = S.pop()

if state[n] == un-visited

S.push(n.Adj)

state[n] = visited

BFS(G)

for each $u \in V$ **do**

state[u]← *un-visited*

put the start node u* into a queue Q

While Q is !empty

n = S.pop()

if state[n] == un-visited

S.push(n.Adj)

state[n] = visited

DFS vs BFS

- What nodes DFS expand?
  - Some left prefix of the tree.
  - Could process the whole tree!
  - If m is finite, takes time $O(b^m)$
- How much space does the fringe take?
  - Only has siblings on path to root, so O(bm)
- Is it complete?
  - m could be infinite, so only if we prevent cycles
- Is it optimal?
  - No, it finds the "leftmost" solution, regardless of depth or cost

- What nodes does BFS expand?
  - Processes all nodes above shallowest solution
  - Let depth of shallowest solution be s
  - Search takes time $O(b^s)$
- How much space does the fringe take?
  - Has roughly the last tier, so $O(b^s)$
- Is it complete?
  - s must be finite if a solution exists
- Is it optimal?
  - Only if costs are all 1 (more on costs later)