Algorithms

Lecture 6: Representation of Graphs, BFS

Anxiao (Andrew) Jiang

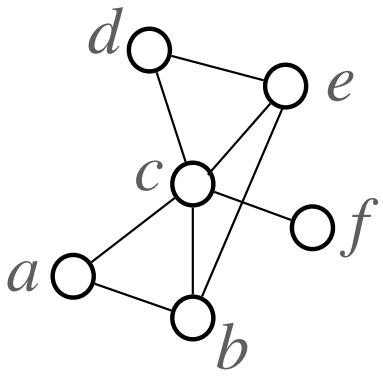
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, b), (a, c), (b, c), \cdots, (c, f)\}$$

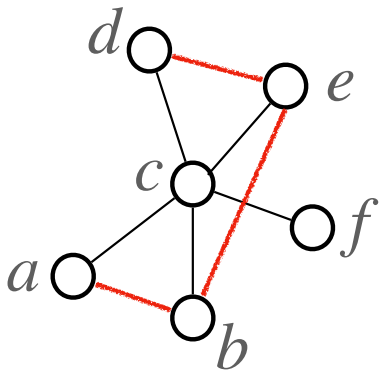
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



Path

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, b), (a, c), (b, c), \cdots, (c, f)\}$$

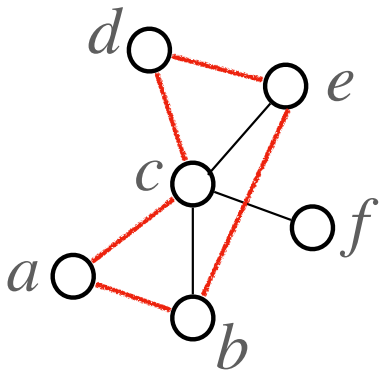
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



Cycle

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, b), (a, c), (b, c), \cdots, (c, f)\}$$

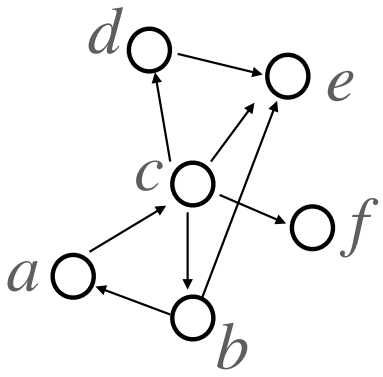
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, c), (b, a), (b, e), \cdots, (c, f)\}$$

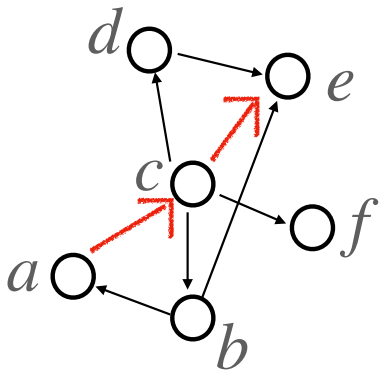
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



Path

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, c), (b, a), (b, e), \cdots, (c, f)\}$$

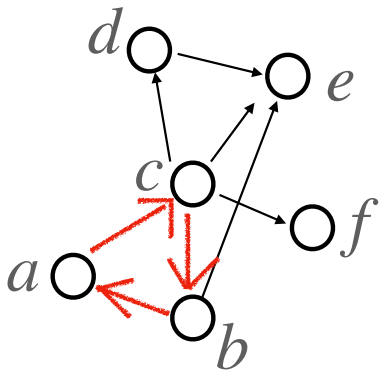
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)



Cycle

$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, c), (b, a), (b, e), \cdots, (c, f)\}$$

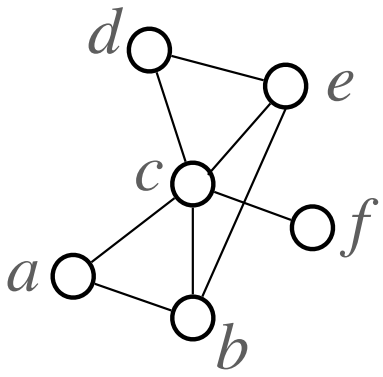
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)

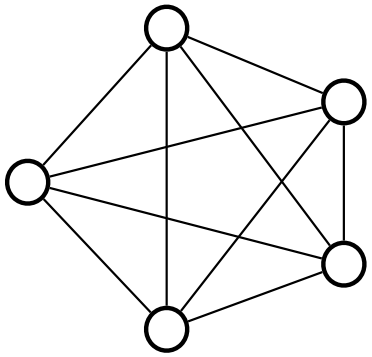


$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, b), (a, c), (b, c), \cdots, (c, f)\}$$

Complete Graph:



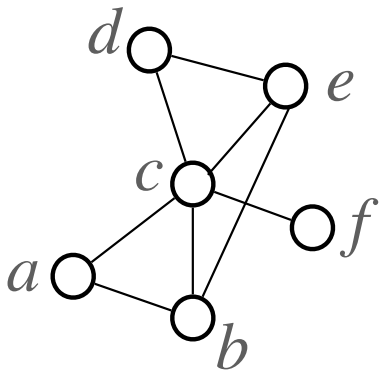
CH 22. Elementary Graph Algorithms

22.1 Representation of Graphs

Graph

Node
(vertex)

Edge
(Undirected,
Directed)

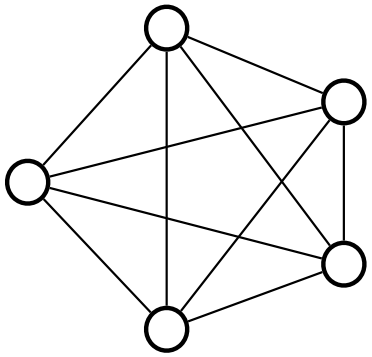


$$G = (V, E)$$

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(a, b), (a, c), (b, c), \cdots, (c, f)\}$$

Complete Graph:



Tree: connected acyclic graph.

Number of edges = number of nodes - 1

CH 22. Elementary Graph Algorithms

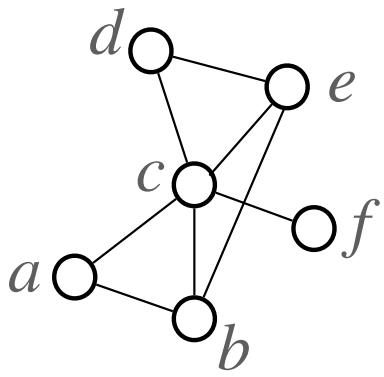
22.1 Representation of Graphs

Adjacency List

a : b, c
b : a, c, e
c : a, b, d, e, f
d : c, e
e : b, c, d
f : c

Size: $O(|V| + |E|)$

$O(V + E)$



CH 22. Elementary Graph Algorithms

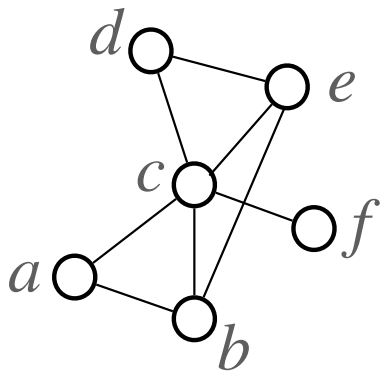
22.1 Representation of Graphs

Adjacency List

Size: $O(|V| + |E|)$

$O(V + E)$

a : b, c
b : a, c, e
c : a, b, d, e, f
d : c, e
e : b, c, d
f : c



Adjacency Matrix

Size: $O(|V|^2)$

$O(V^2)$

	a	b	c	d	e	f
a	0	1	1	0	0	0
b	1	0	1	0	1	0
c	1	1	0	1	1	1
d	0	0	1	0	1	0
e	0	1	1	1	0	0
f	0	0	1	0	0	0

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

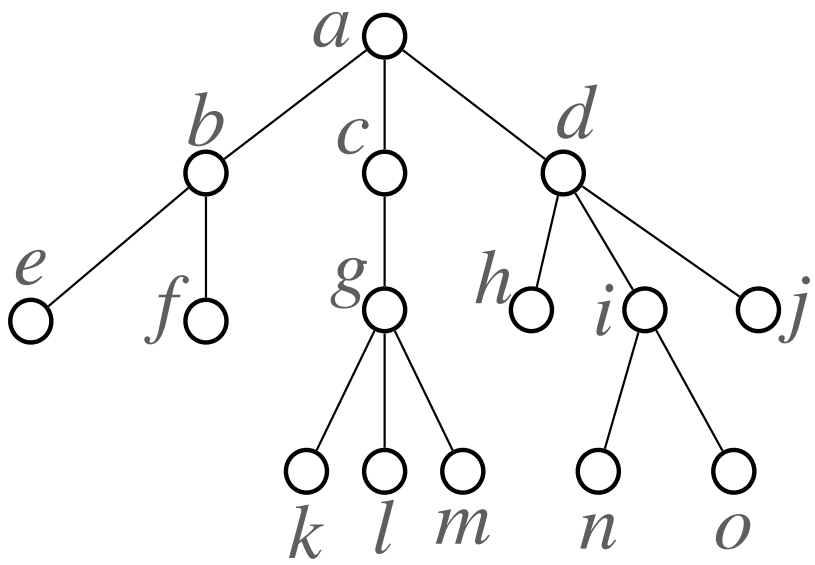
Basic idea:

Starting at a node,
We first check its 1-hop neighbors,
then check its 2-hop neighbors (neighbors of 1-hop neighbors),
then check its 3-hop neighbors (neighbors of 2-hop neighbors),
then check its 4-hop neighbors (neighbors of 3-hop neighbors),
……,
until all reachable nodes are checked.

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

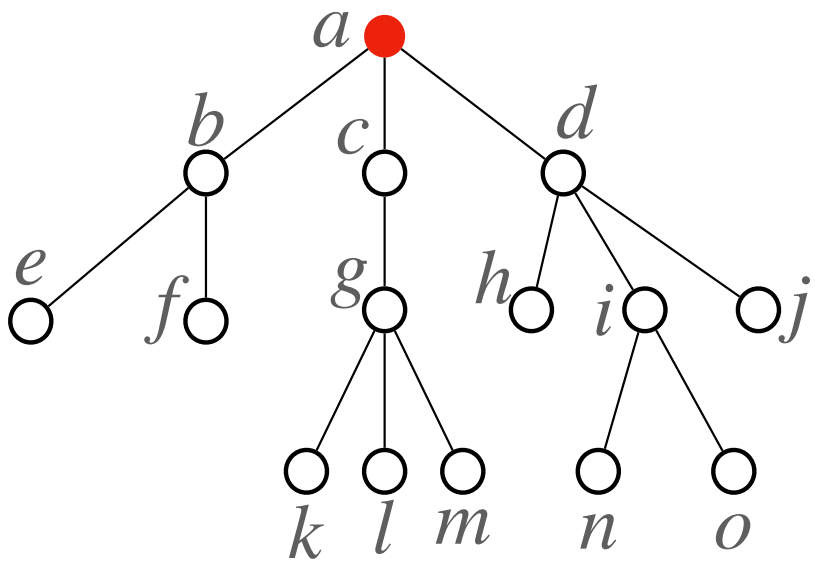
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

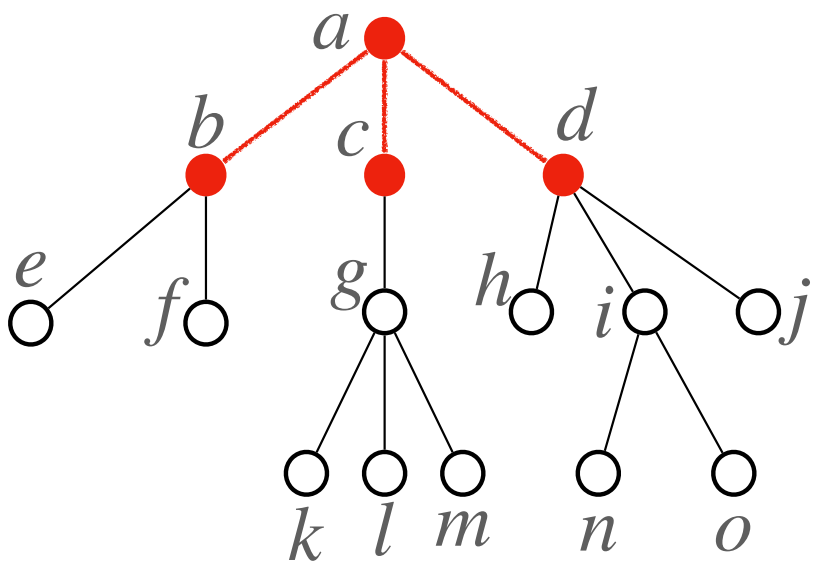
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

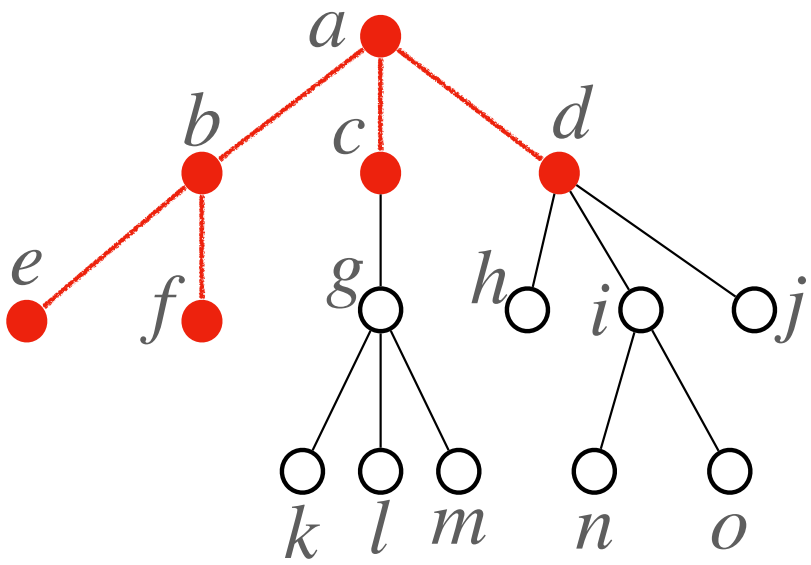
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

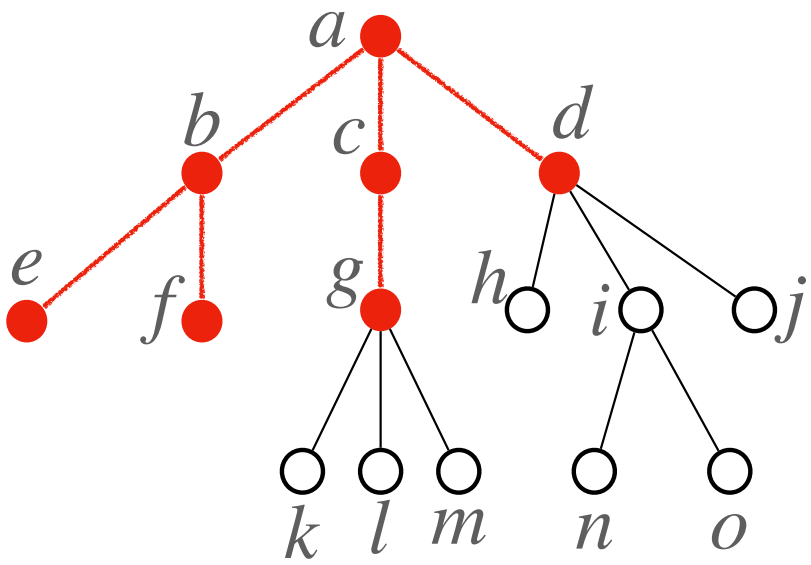
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

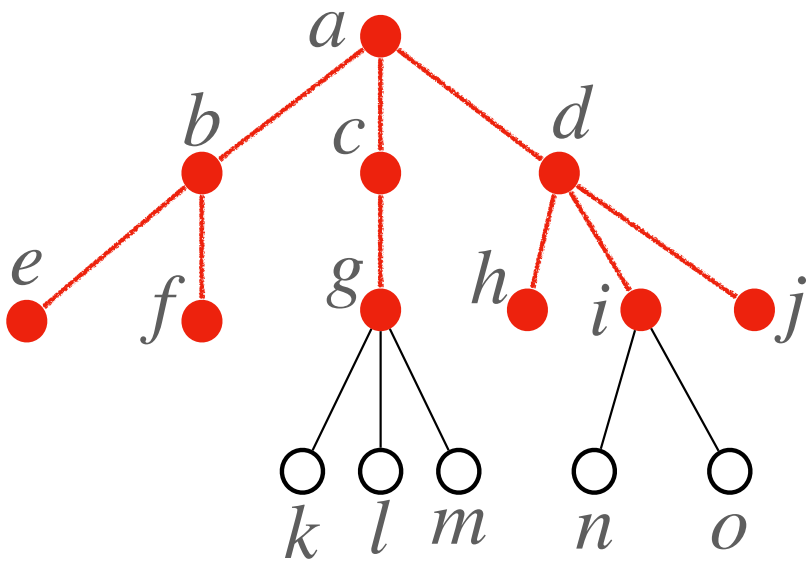
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

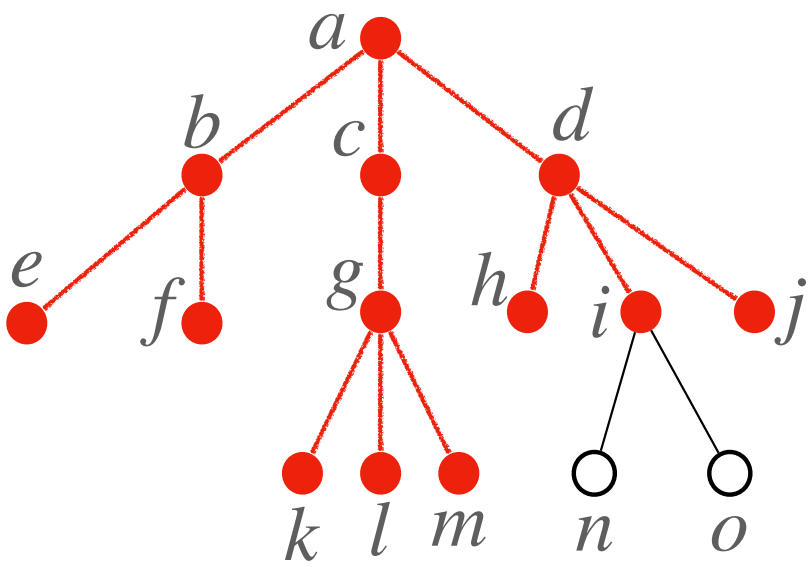
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)

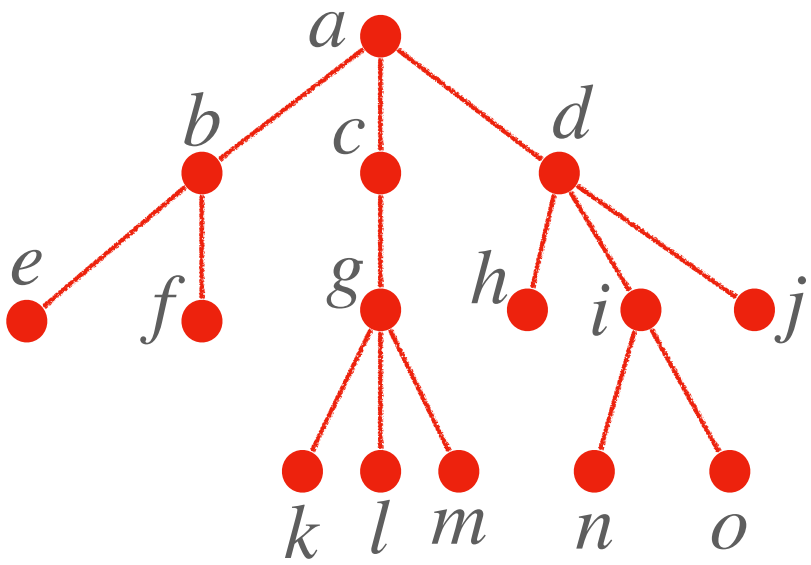
Example of BFS on a tree:



CH 22. Elementary Graph Algorithms

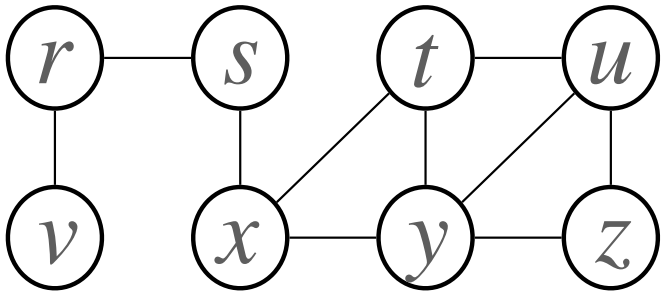
22.2 Breadth First Search (BFS)

Example of BFS on a tree:



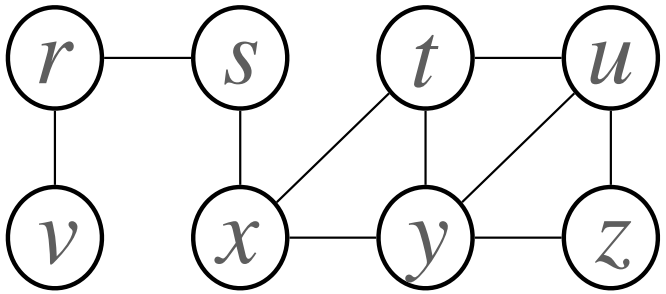
CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



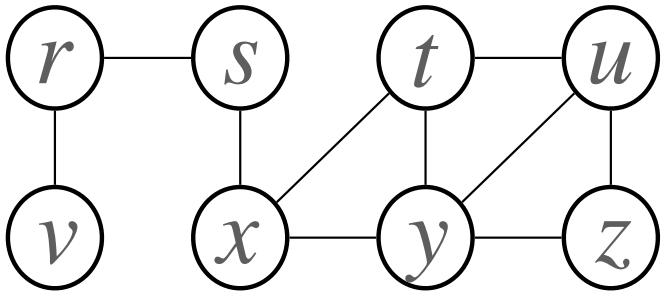
Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

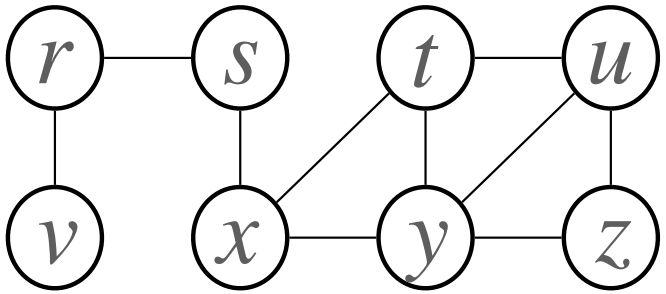
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

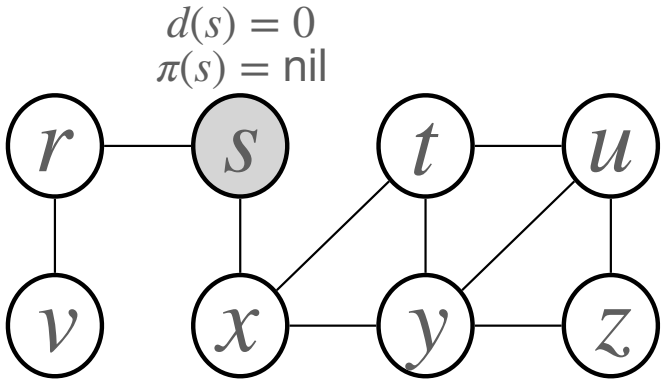
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

s

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

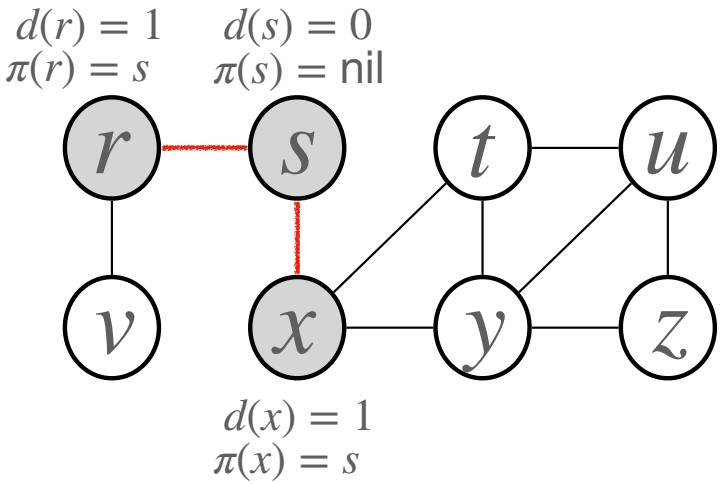
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

s r x

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

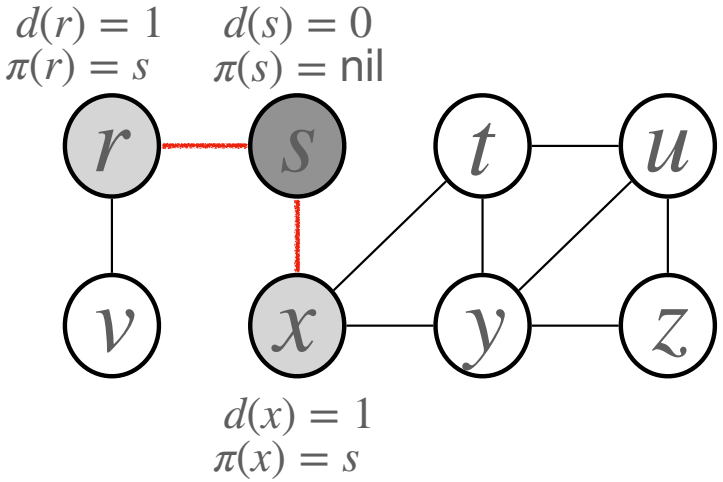
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

r x

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

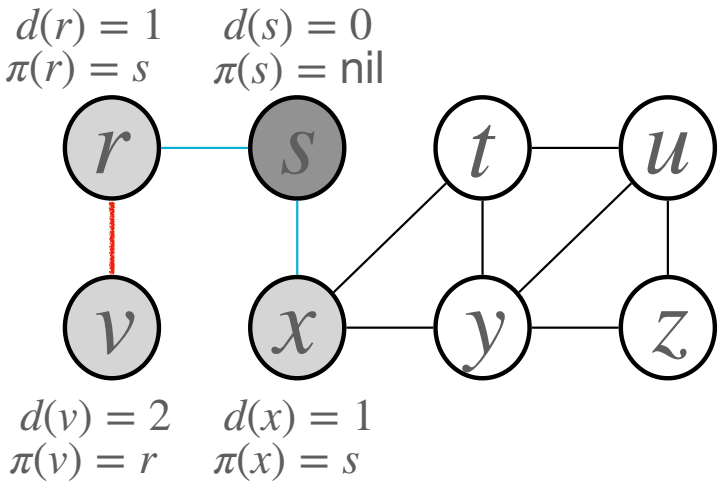
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

r x v

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

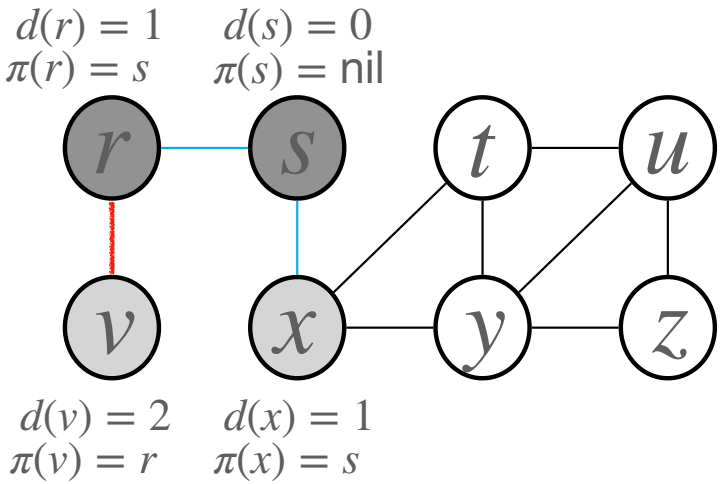
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

x v

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

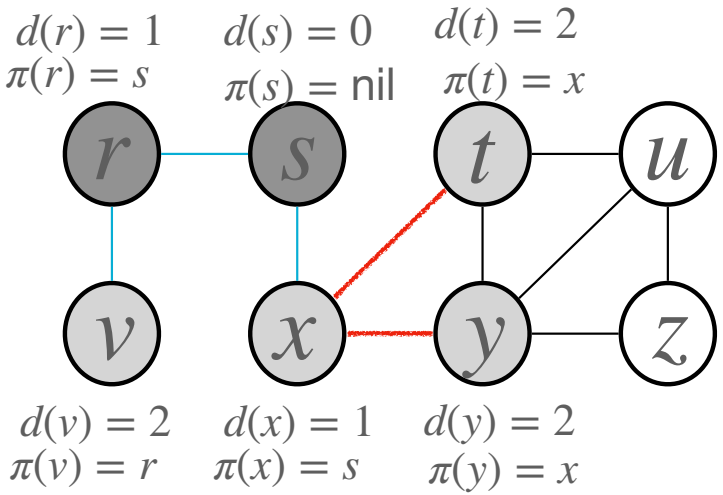
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

x v t y

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

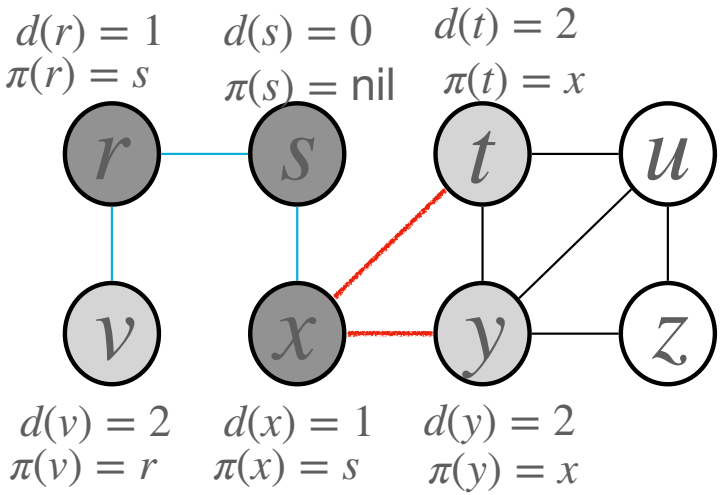
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

v t y

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

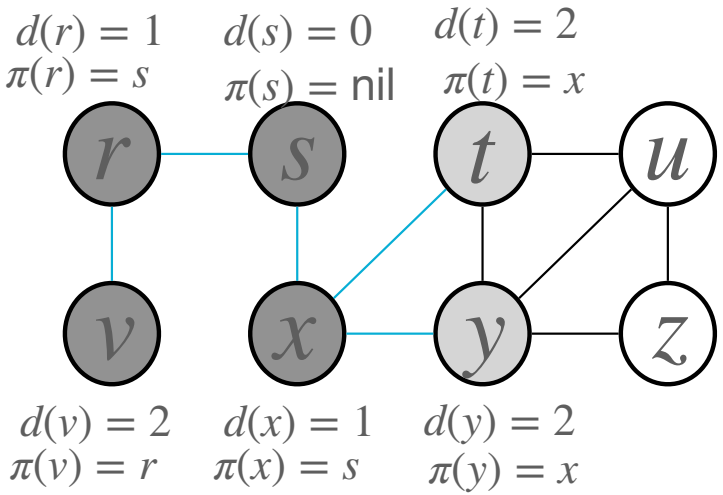
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

t y

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

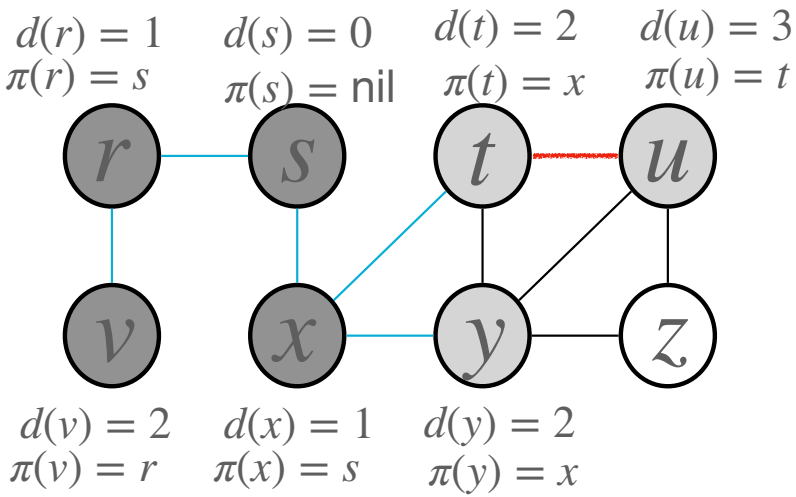
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

t y u

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

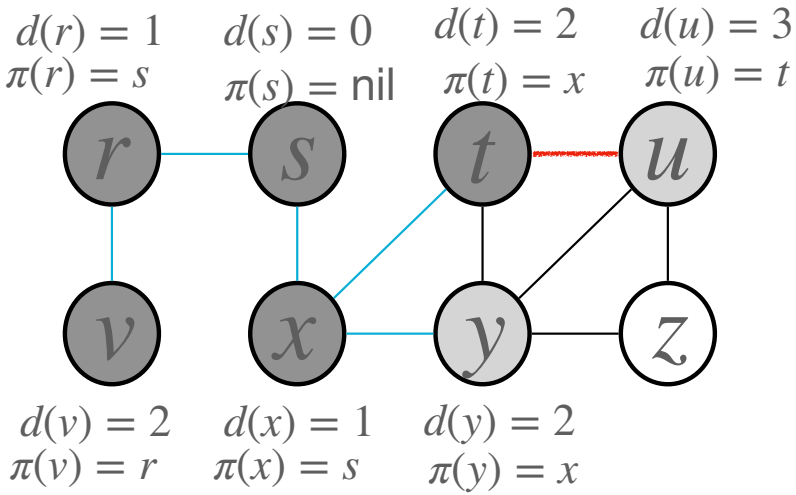
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

y u

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

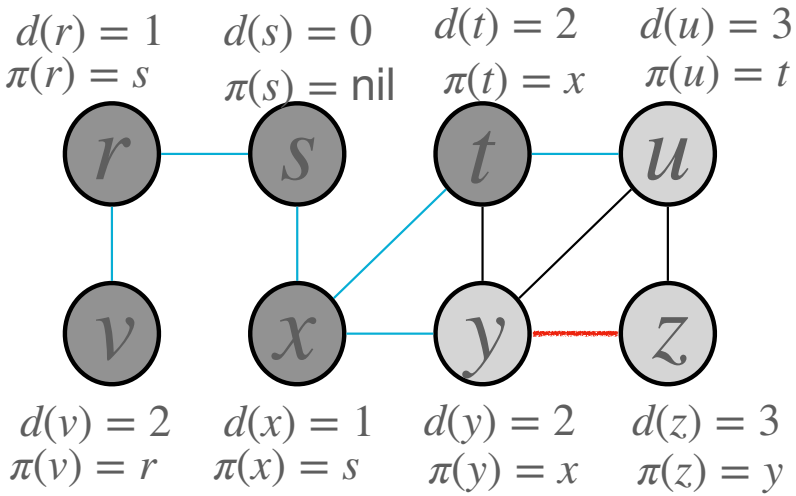
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

y u z

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

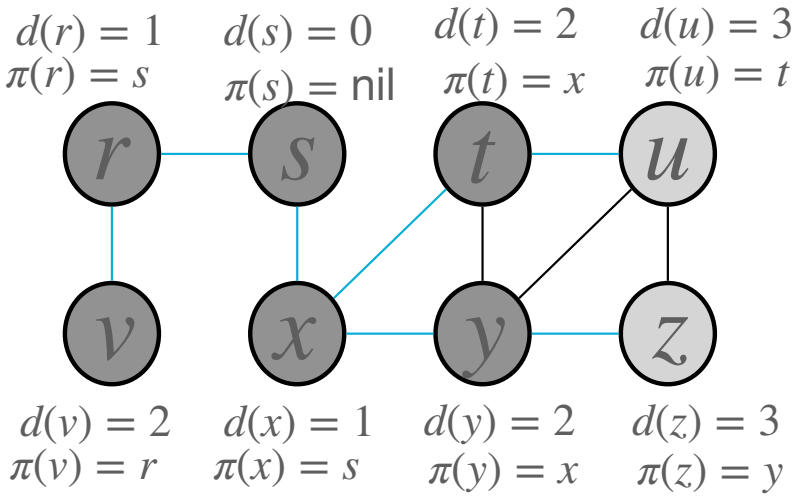
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

u z

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

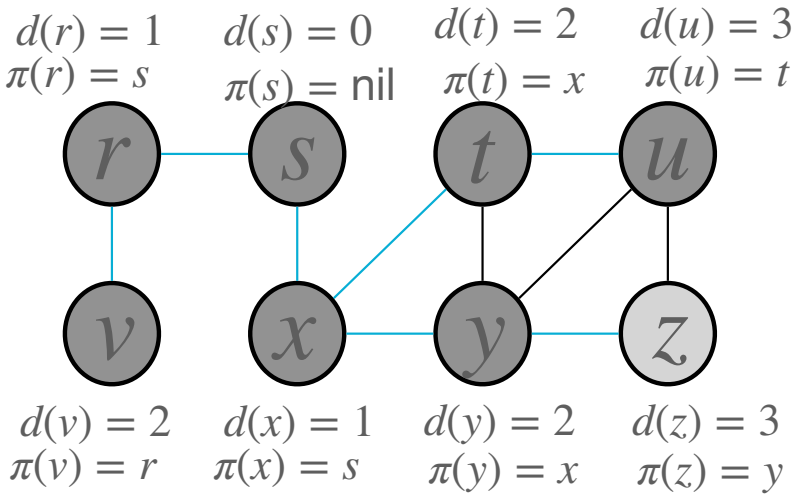
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

z

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

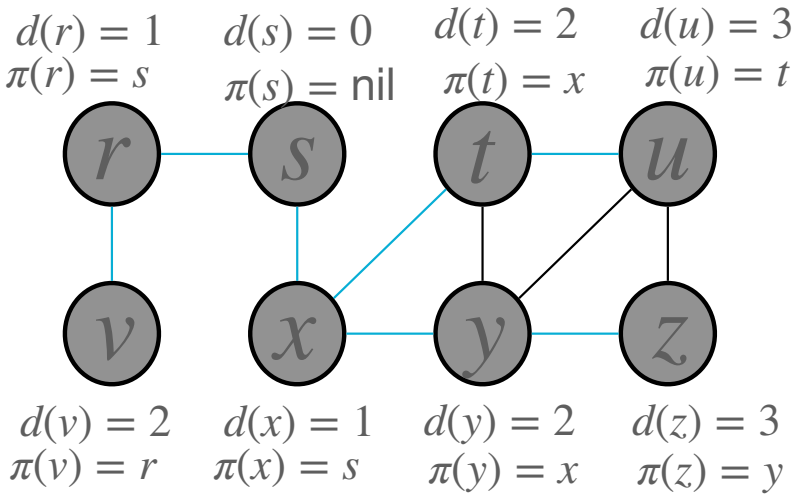
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

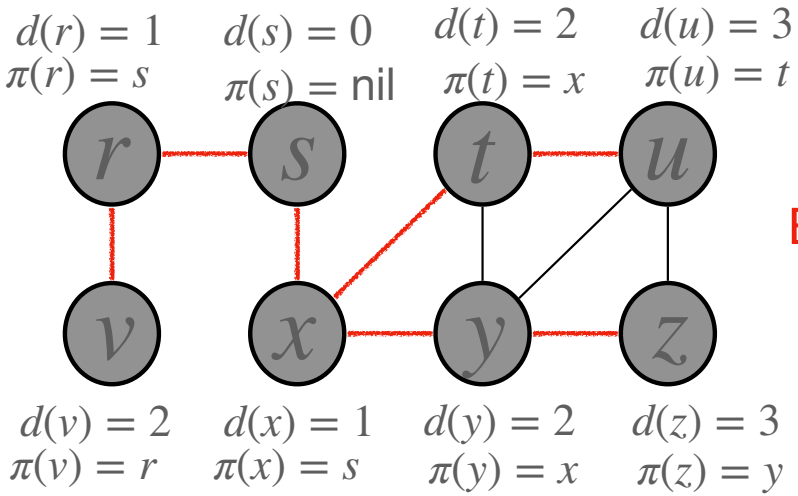
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

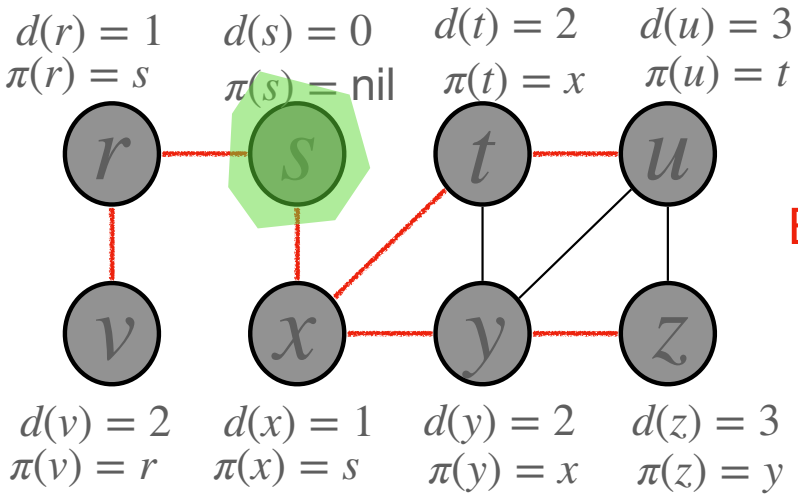
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

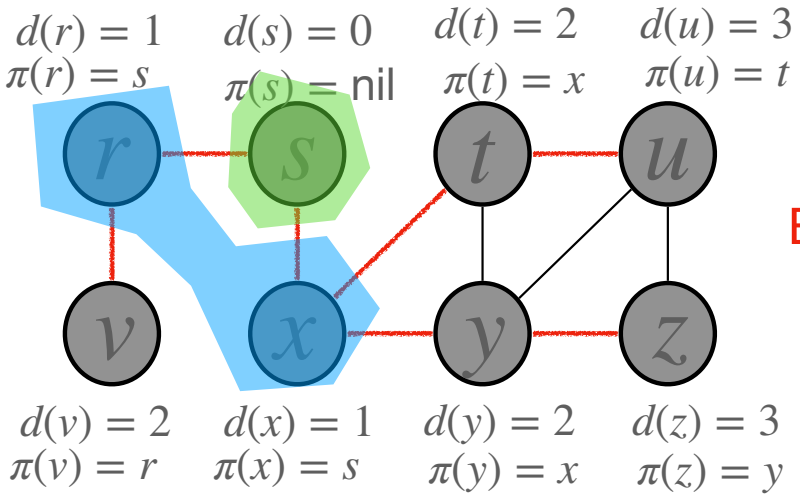
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

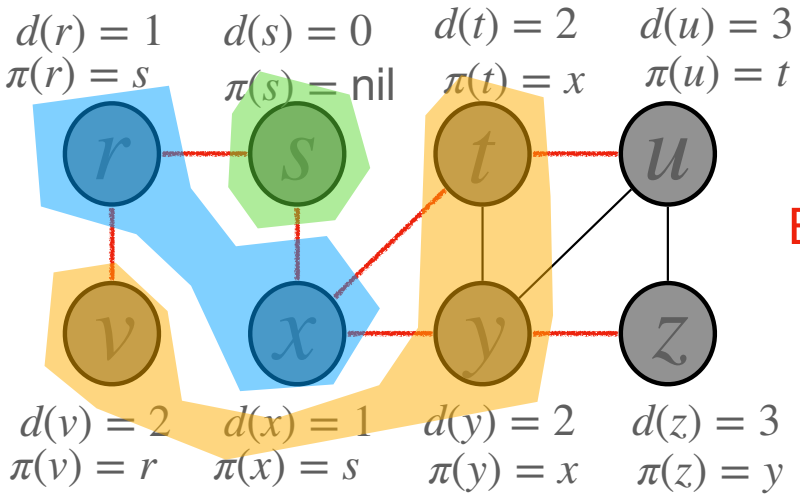
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

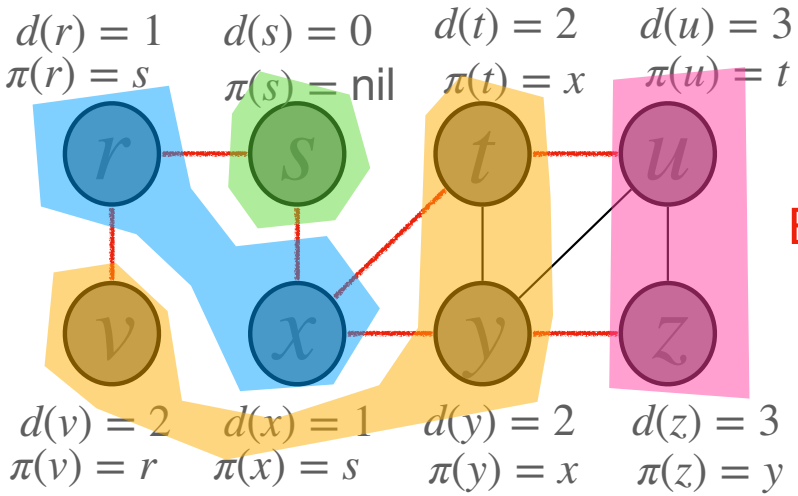
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

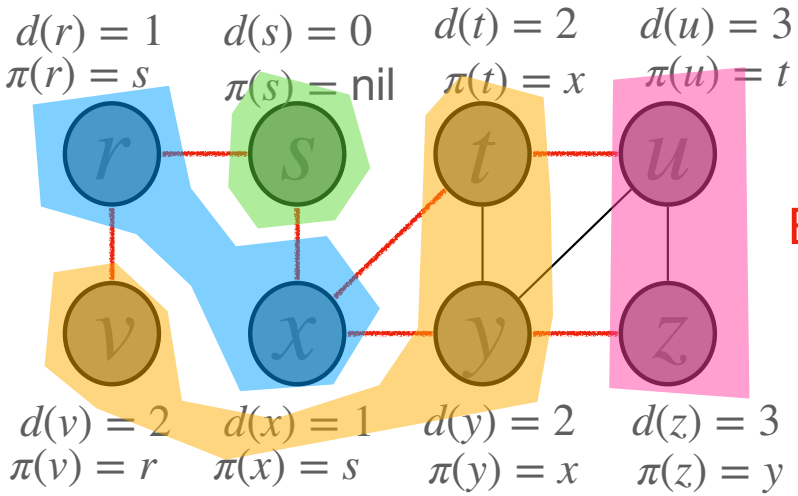
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Queue (for remembering the discovered nodes):

$s,\ r,\ x,\ v,\ t,\ y,\ u,\ z$

0 1 2 3

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

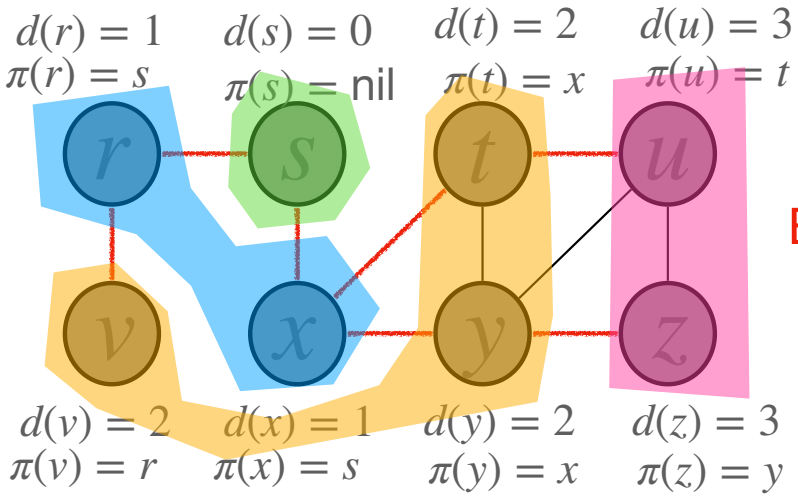
For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

CH 22. Elementary Graph Algorithms

22.2 Breadth First Search (BFS)



BFS tree

Undiscovered nodes: white

Discovered (but unfinished) nodes: gray

Finished nodes: black

For each node v :

1) distance from root to v : $d(v)$

2) the node that discovered v : $\pi(v)$

Queue (for remembering the discovered nodes):

$s,\ r,\ x,\ v,\ t,\ y,\ u,\ z$

0 1 2 3

Time complexity: $O(V+E)$