CS-218
DATA STRUCTURE

Dr. Hashim Yasin

National University of Computer and Emerging Sciences,

Faisalabad, Pakistan.

BREADTH FIRST SEARCH

Breath First Search

Breadth-First Search

BFS is useful to,

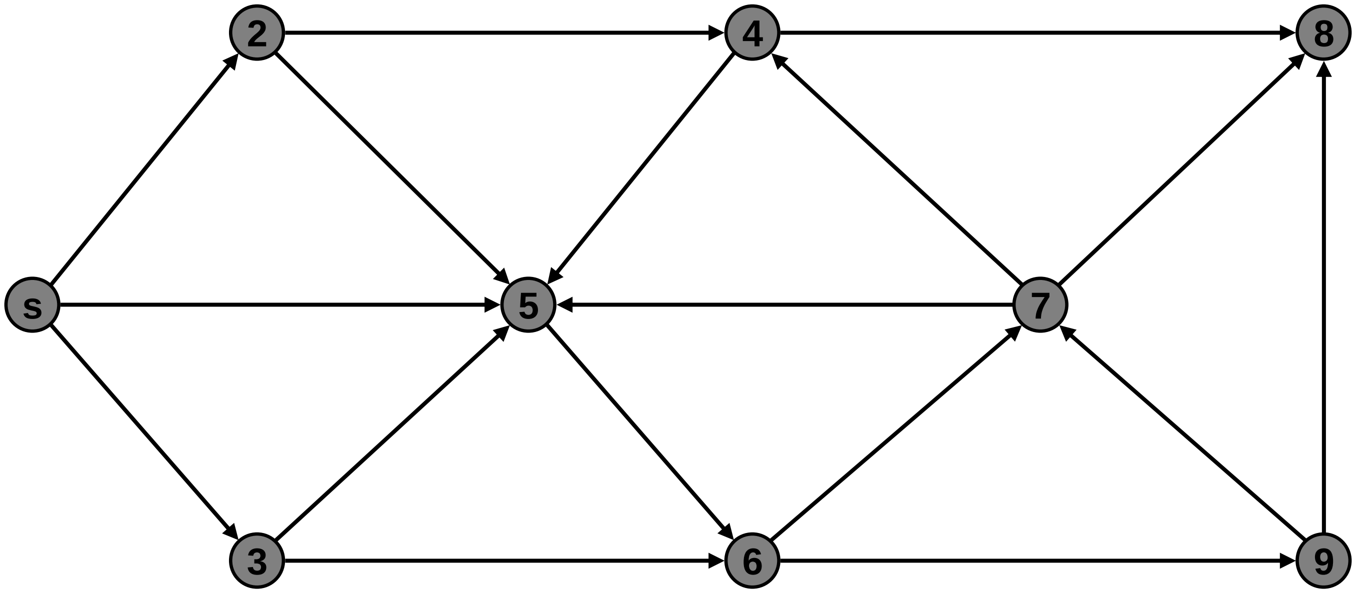
- Find the shortest path from a vertex s to a vertex v.
- Find the length of such a path.
- Find if a strongly connected directed graph contains cycles
- Construct a BFS tree/forest from a graph

Breath First Search … Algorithm

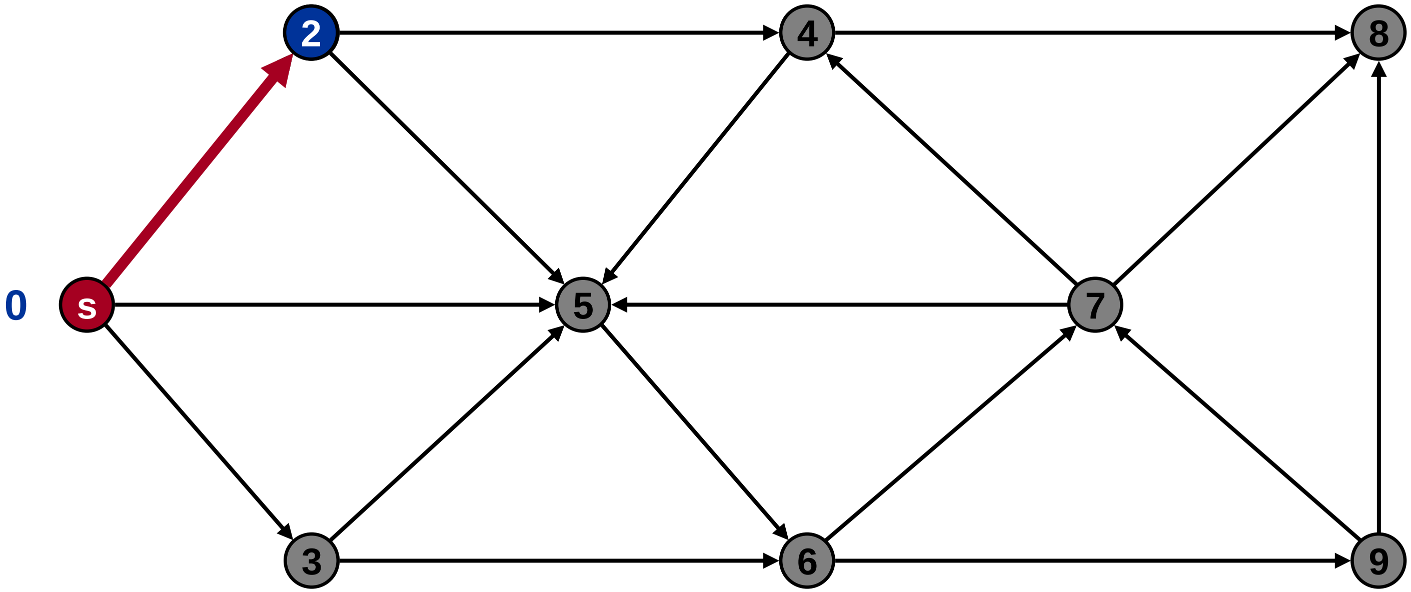
- The algorithm uses a **queue data structure** to store intermediate results as it traverses the graph, as follows:

1. Enqueue the root node
2. Dequeue a node and examine it
    1. If the element required is found in this node, quit the search and return a result.
    2. Otherwise enqueue any successors (the direct child nodes) that have not yet been discovered.
3. If the queue is empty, every node on the graph has been examined – quit the search and return "not found".
4. If the queue is not empty, repeat from Step 2.

Breadth First Search



Breadth First Search



Undiscovered

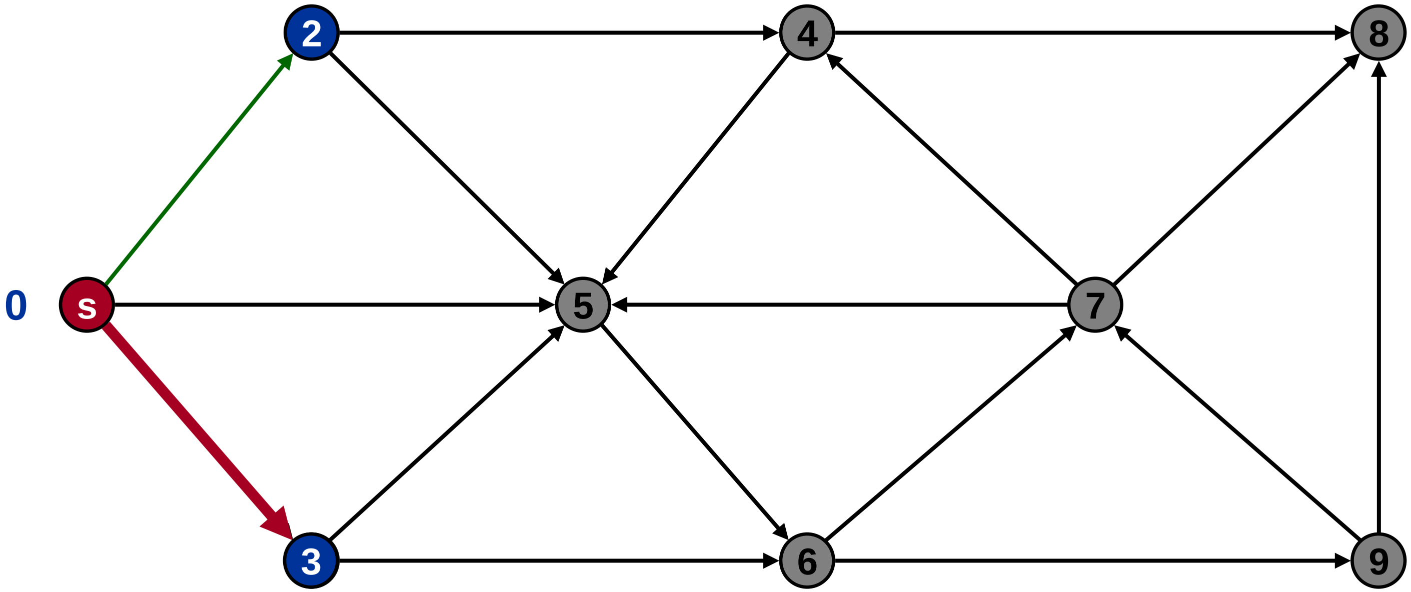
Discovered

Top of queue

Finished

Queue: s

Breadth First Search



Undiscovered

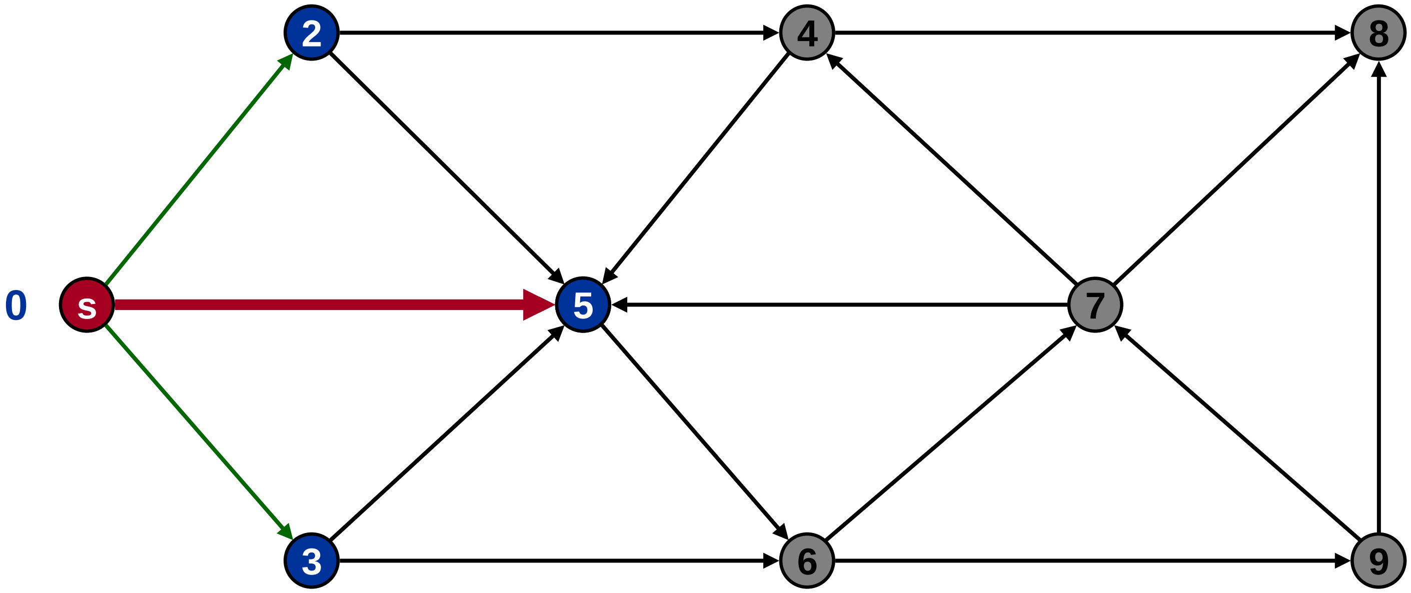
Discovered

Top of queue

Finished

Queue: s 2

Breadth First Search



Undiscovered

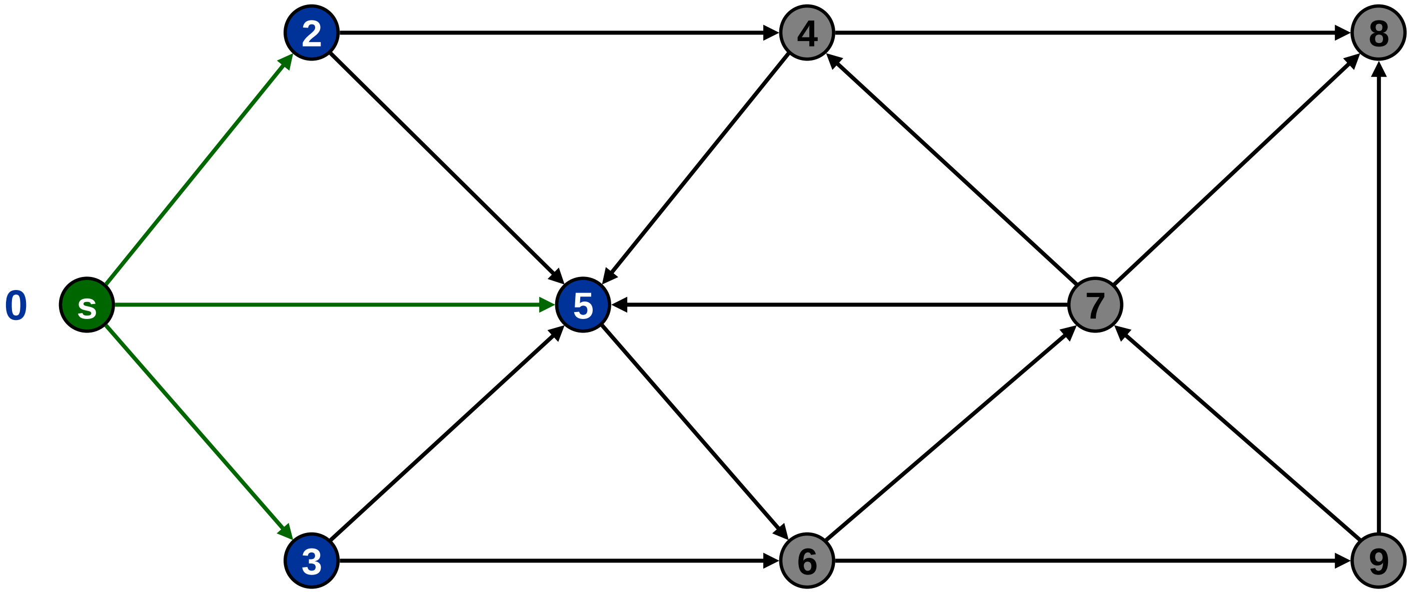
Discovered

Top of queue

Finished

Queue: s 2 3

Breadth First Search



Undiscovered

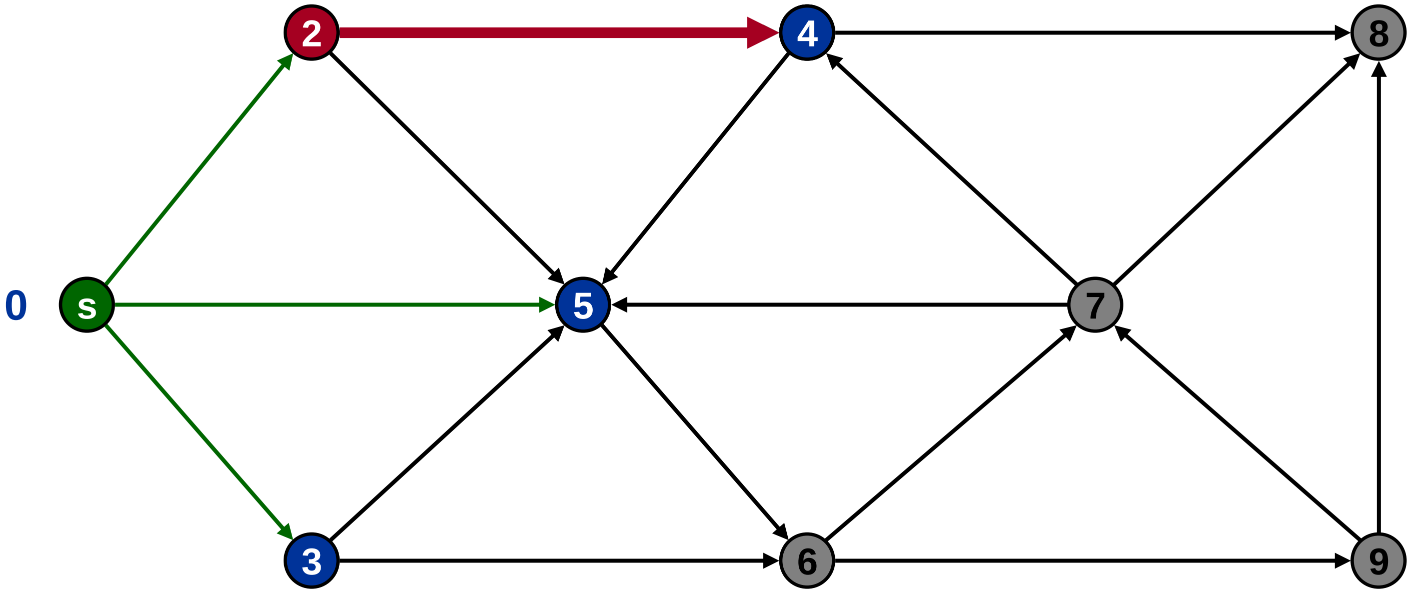
Discovered

Top of queue

Finished

Queue: 2 3 5

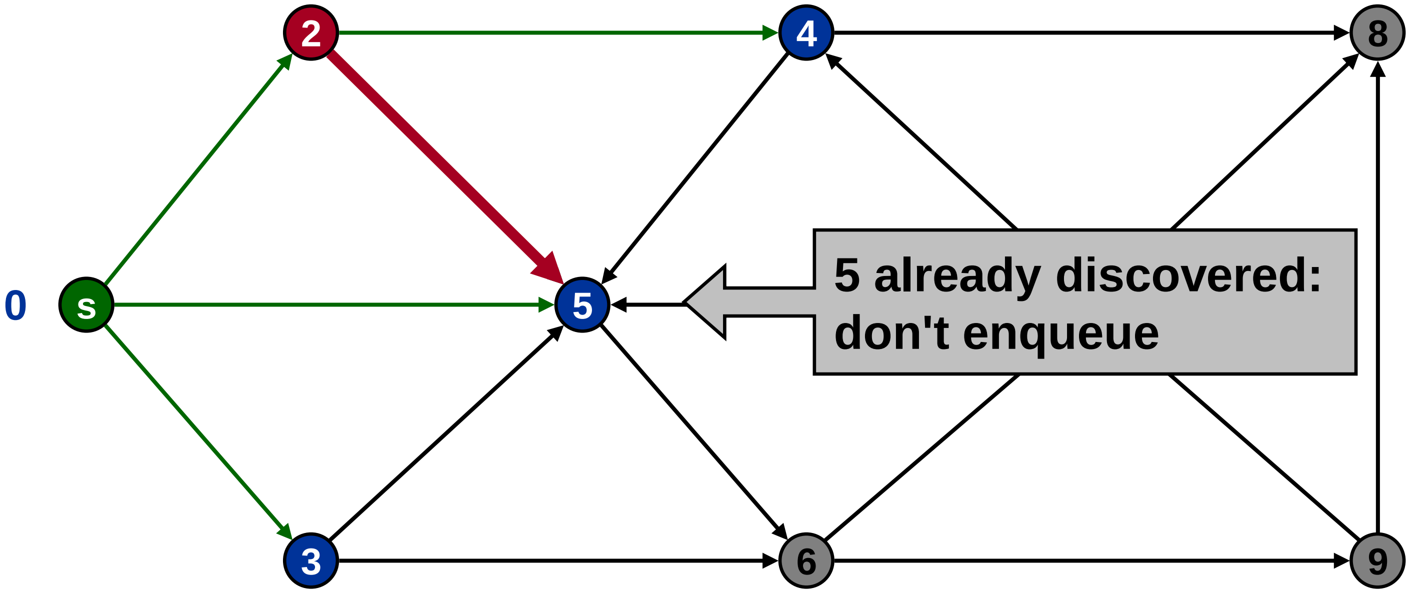
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5

Breadth First Search



5 already discovered: don't enqueue

Undiscovered

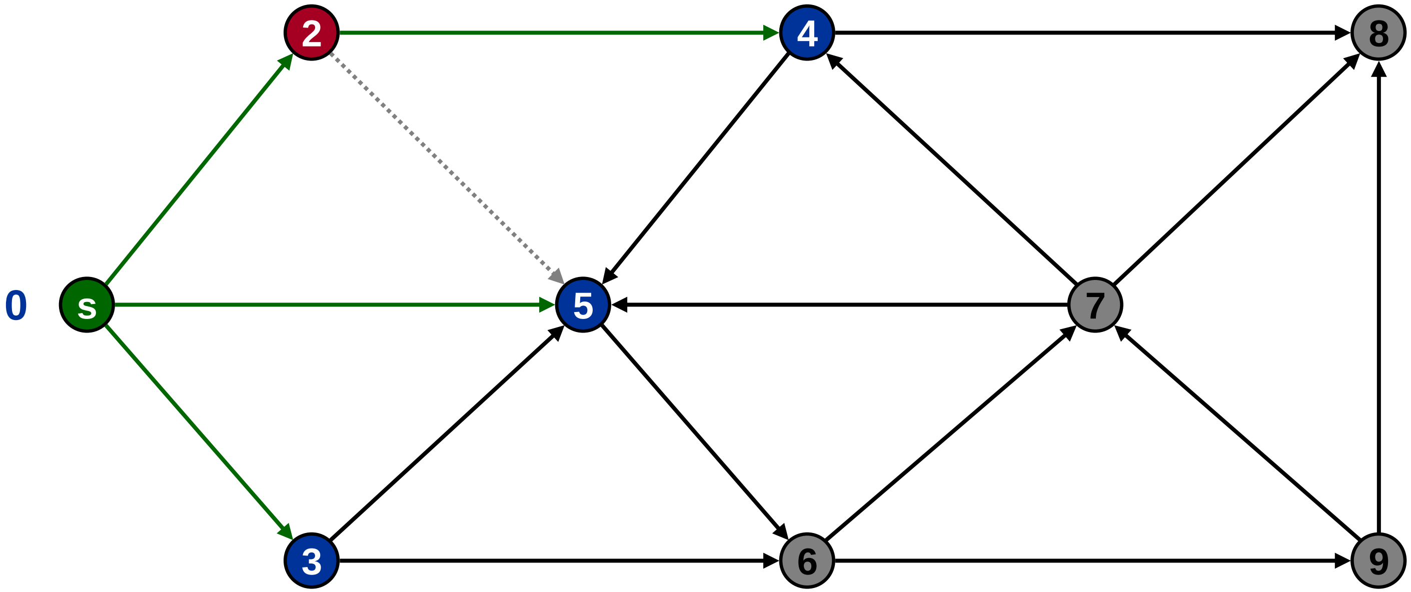
Discovered

Top of queue

Finished

Queue: 2 3 5 4

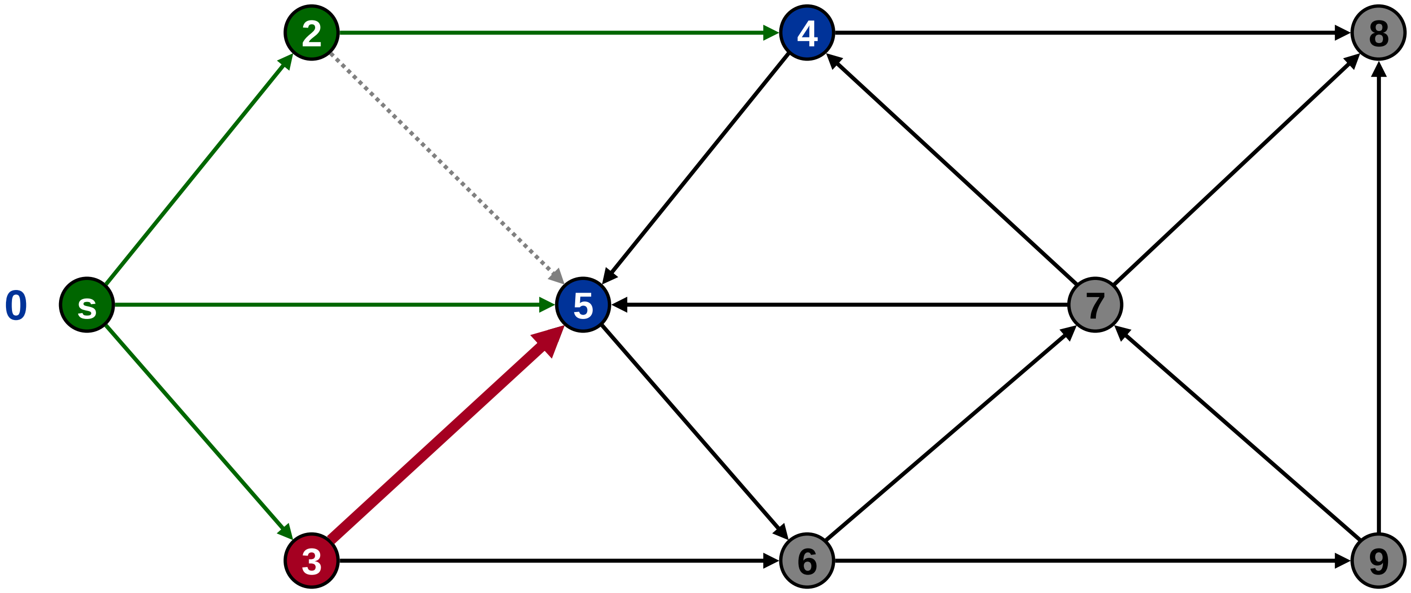
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5 4

Breadth First Search



Undiscovered

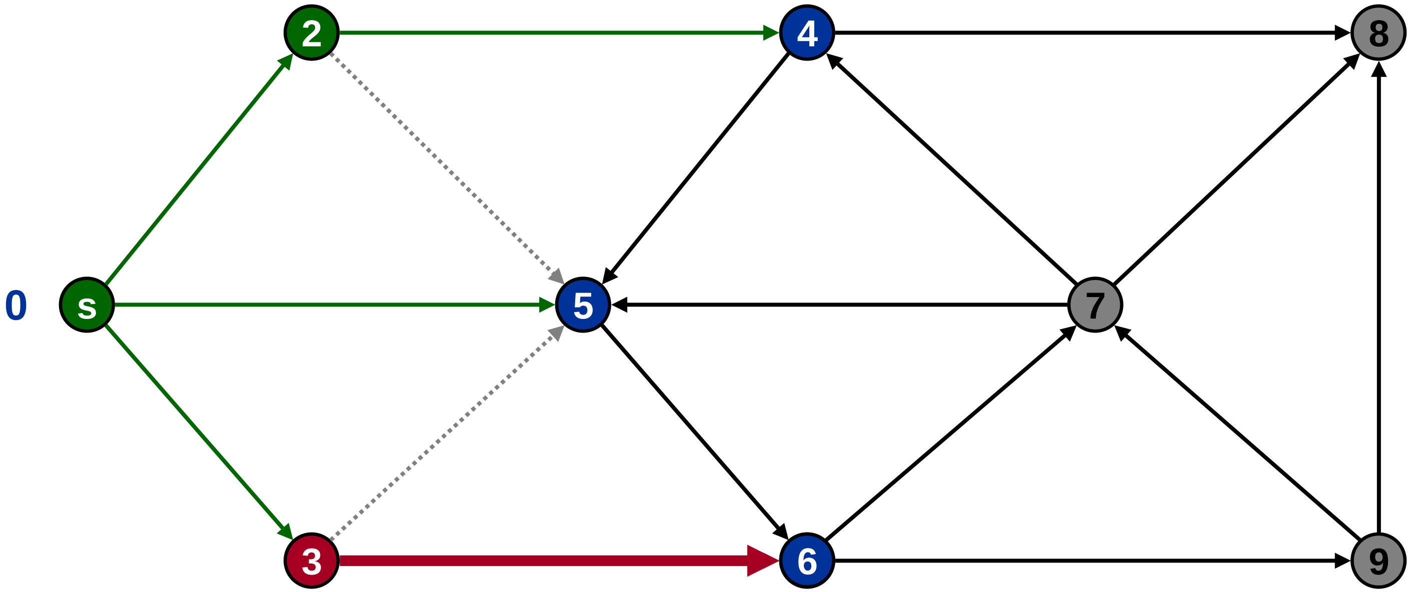
Discovered

Top of queue

Finished

Queue: 3 5 4

Breadth First Search



Undiscovered

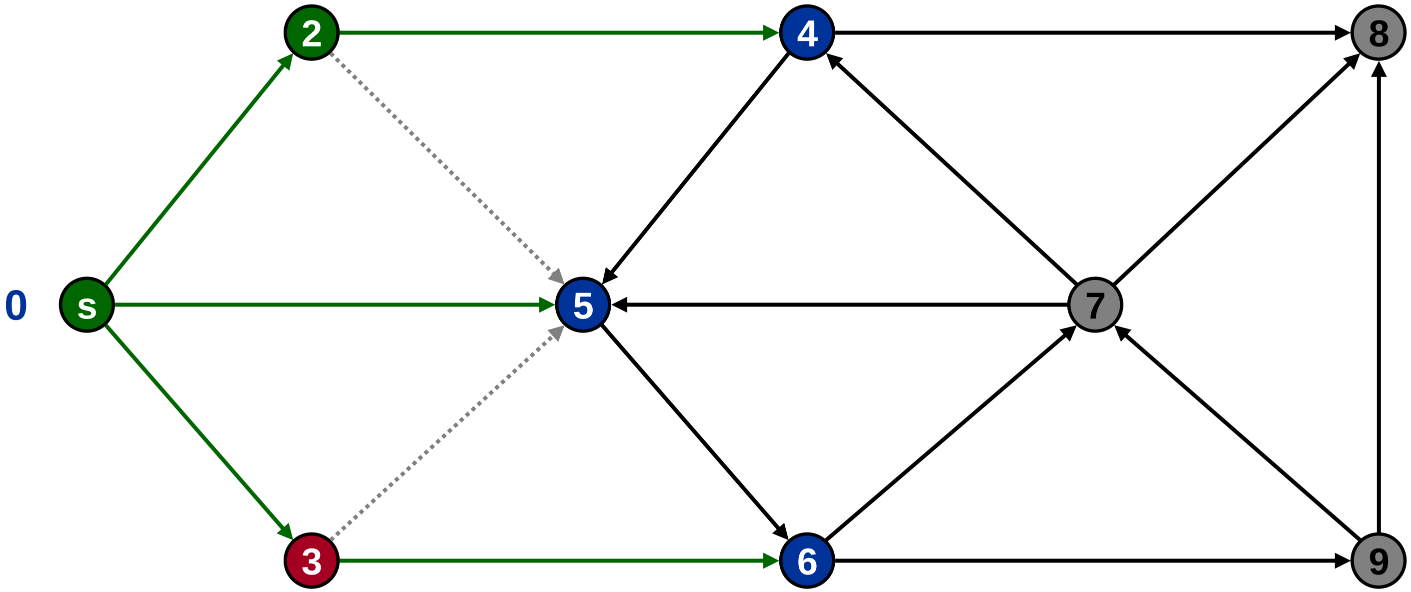
Discovered

Top of queue

Finished

Queue: 3 5 4

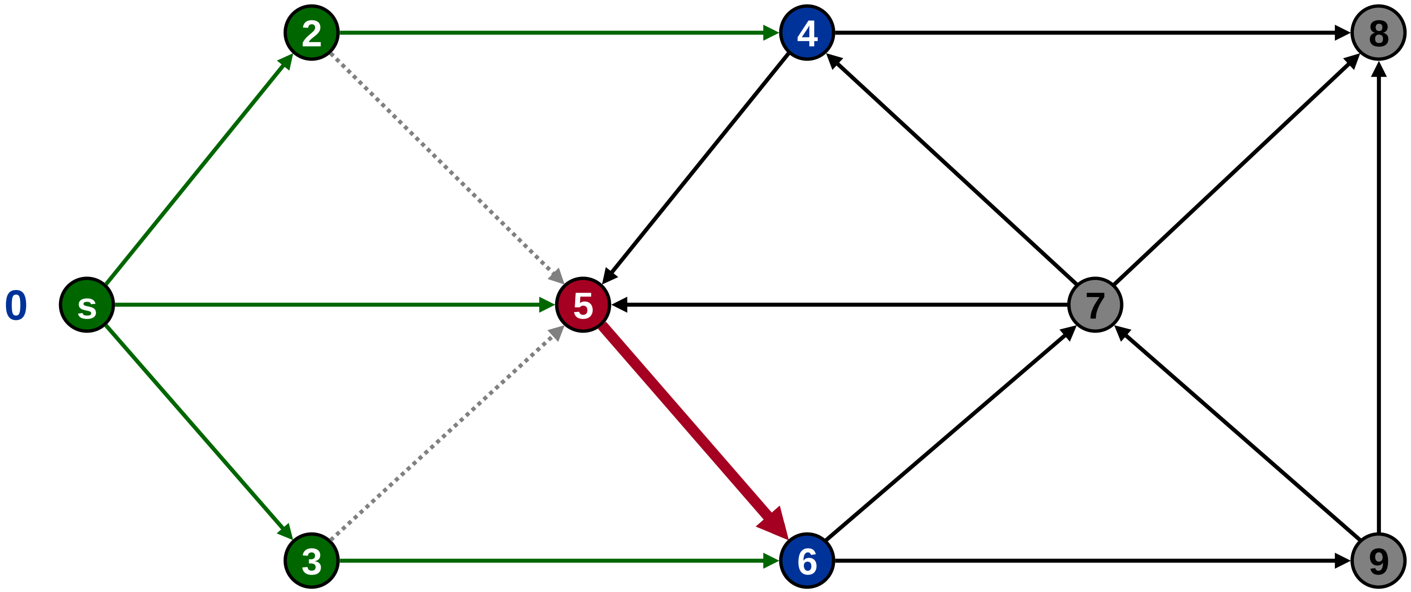
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 3 5 4 6

Breadth First Search



Undiscovered

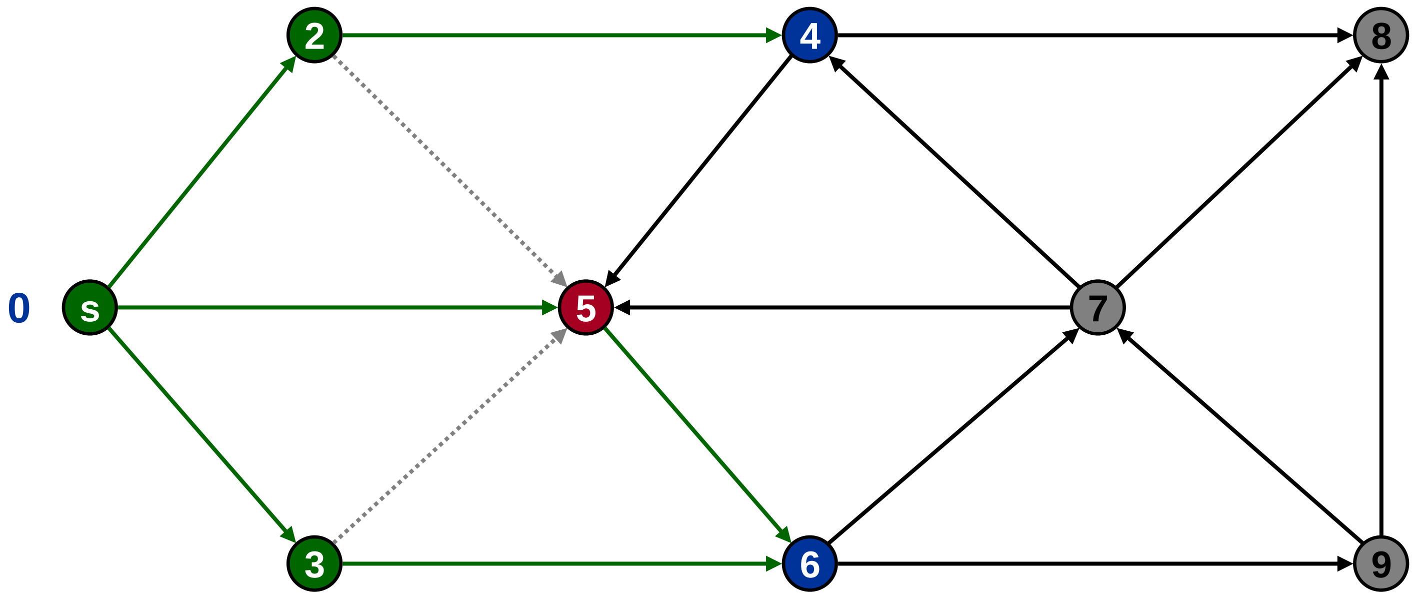
Discovered

Top of queue

Finished

Queue: 5 4 6

Breadth First Search



Undiscovered

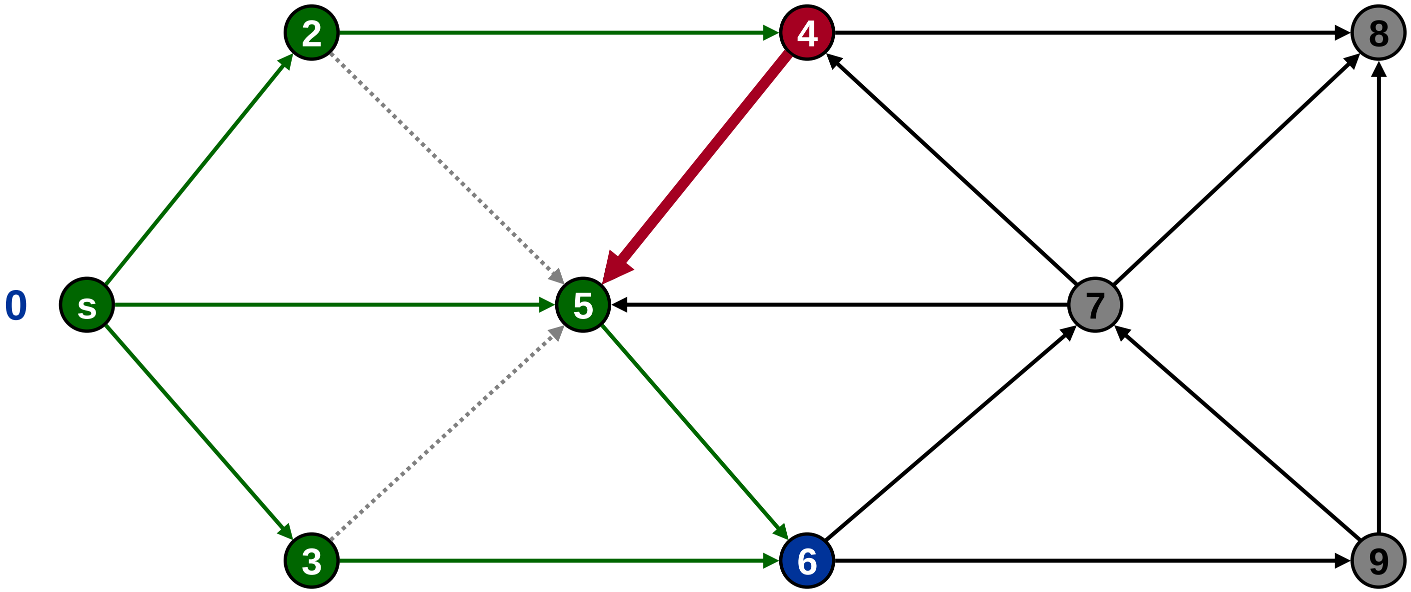
Discovered

Top of queue

Finished

Queue: 5 4 6

Breadth First Search



Undiscovered

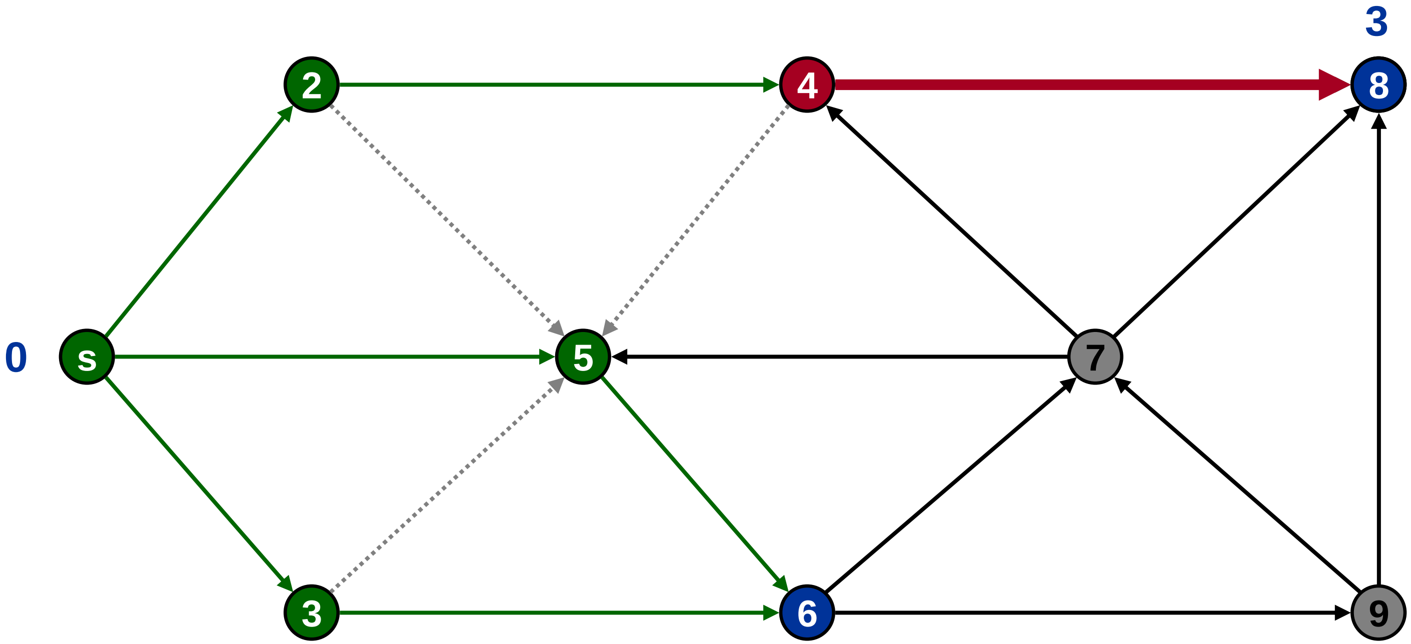
Discovered

Top of queue

Finished

Queue: 4 6

Breadth First Search



Undiscovered

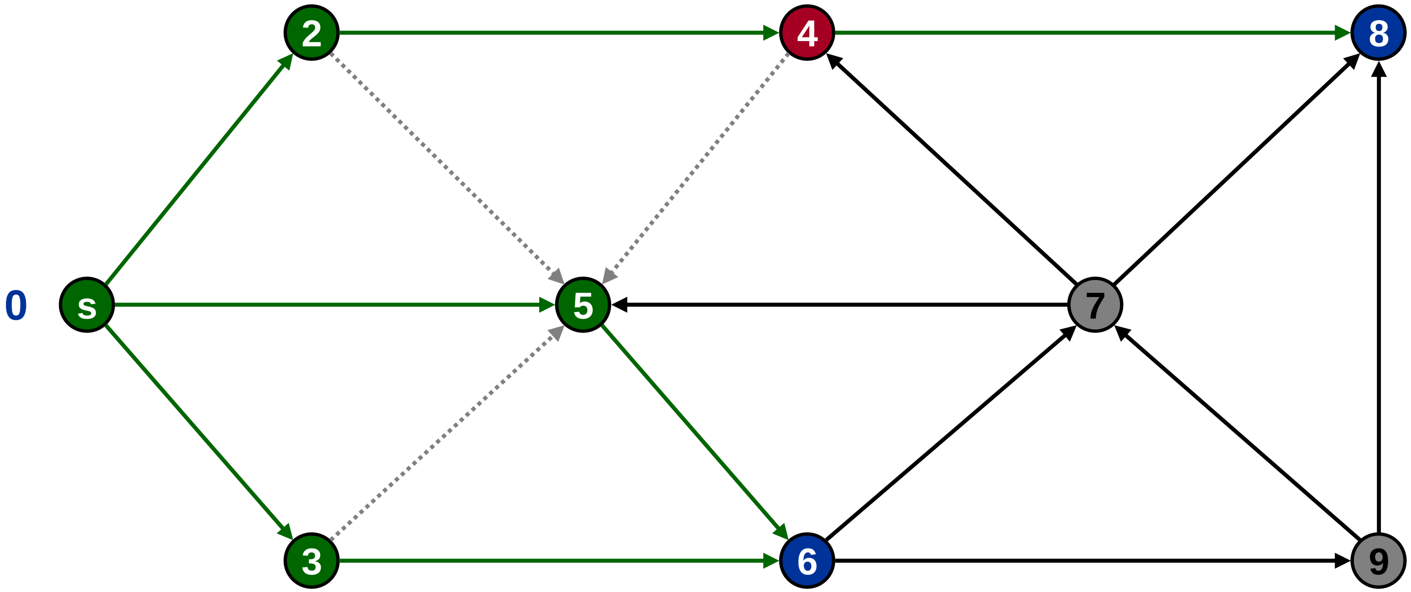
Discovered

Top of queue

Finished

Queue: 4 6

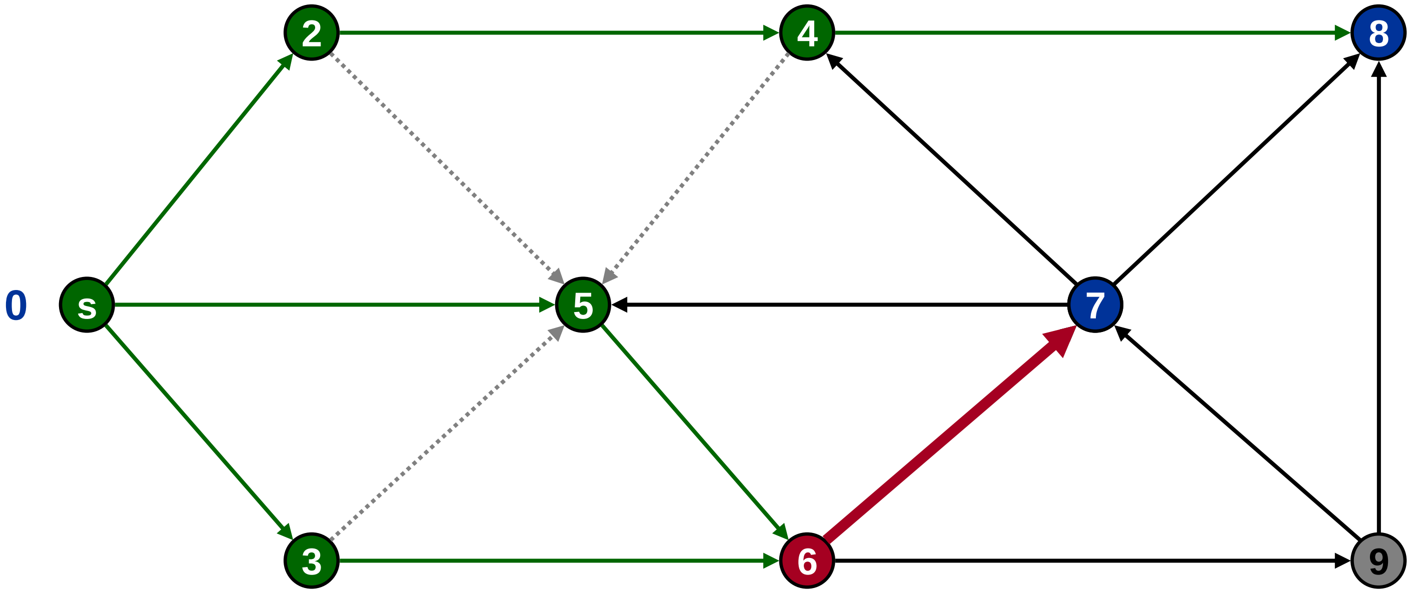
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 4 6 8

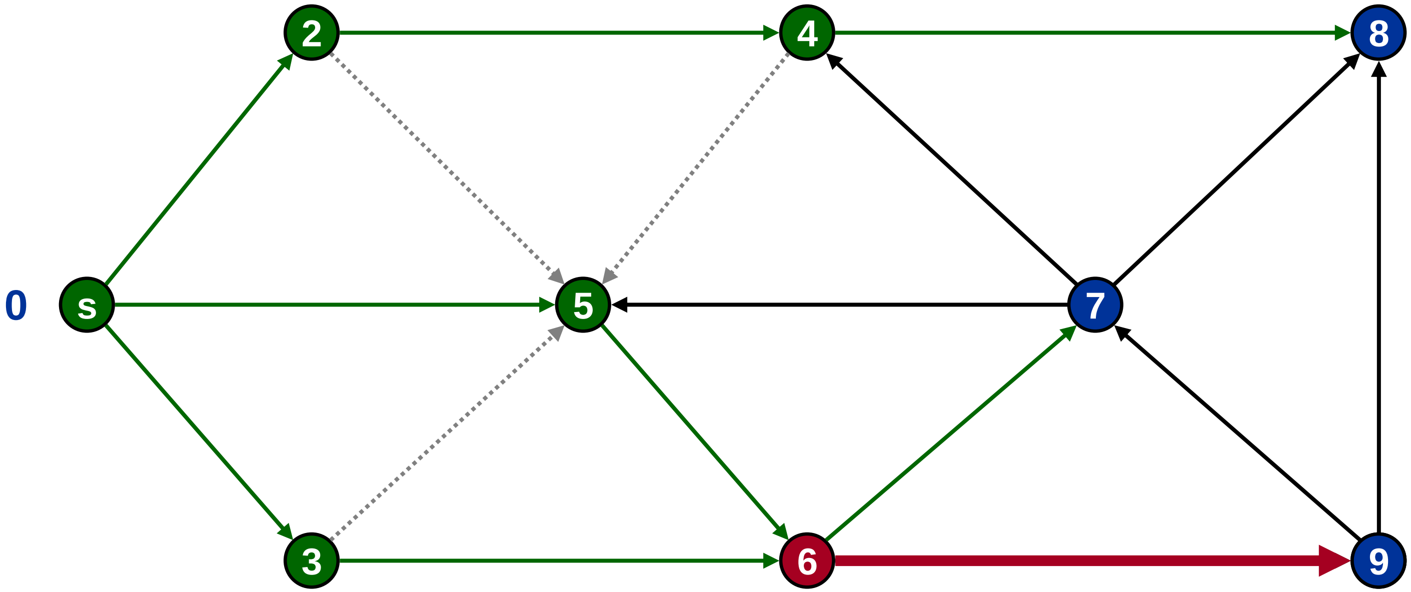
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 6 8

Breadth First Search



Undiscovered

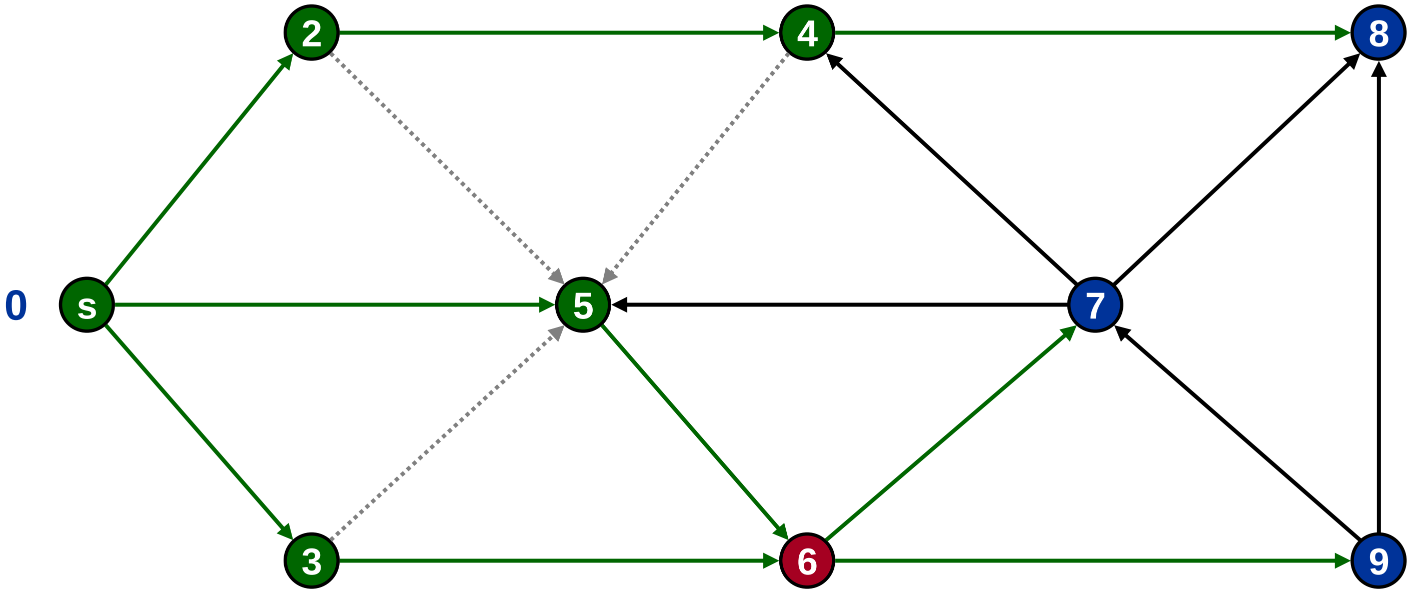
Discovered

Top of queue

Finished

Queue: 6 8 7

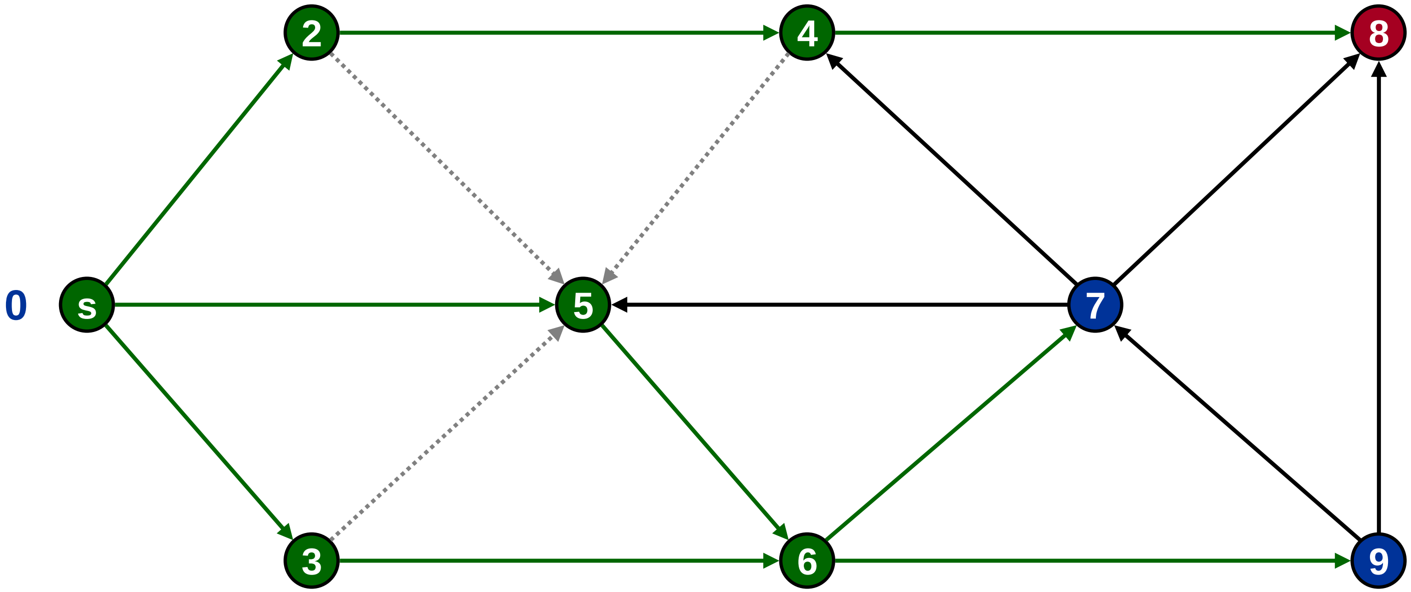
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 6 8 7 9

Breadth First Search



Undiscovered

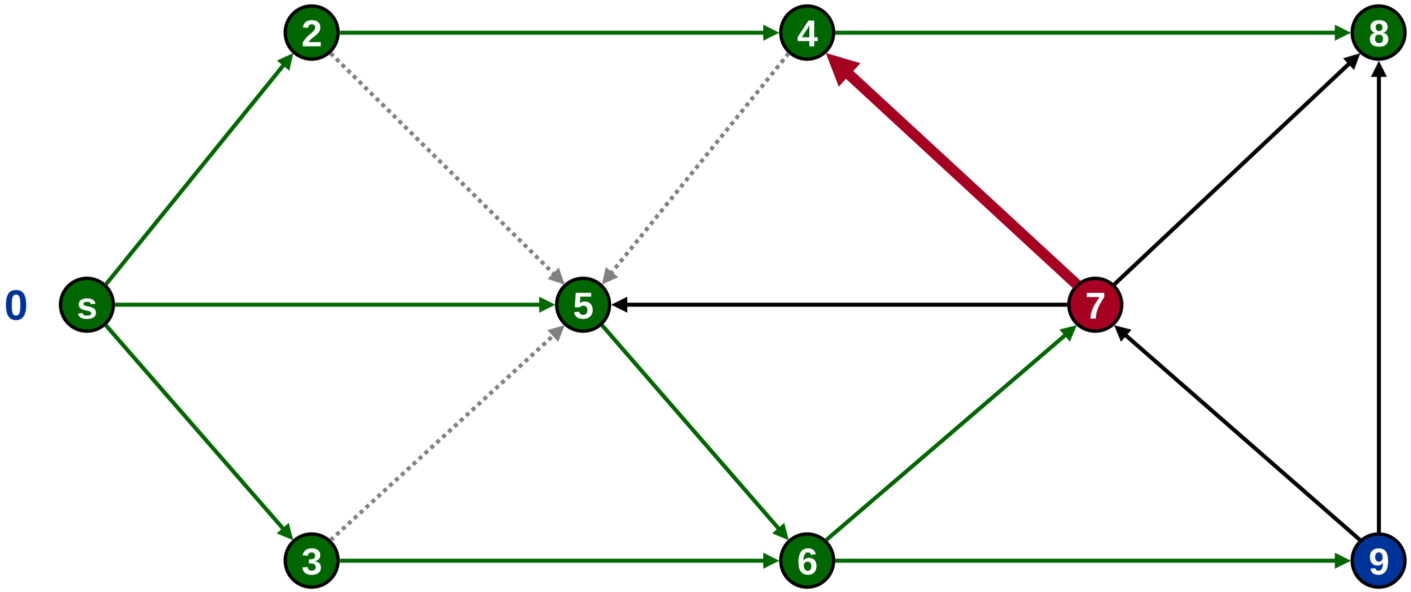
Discovered

Top of queue

Finished

Queue: 8 7 9

Breadth First Search



Undiscovered

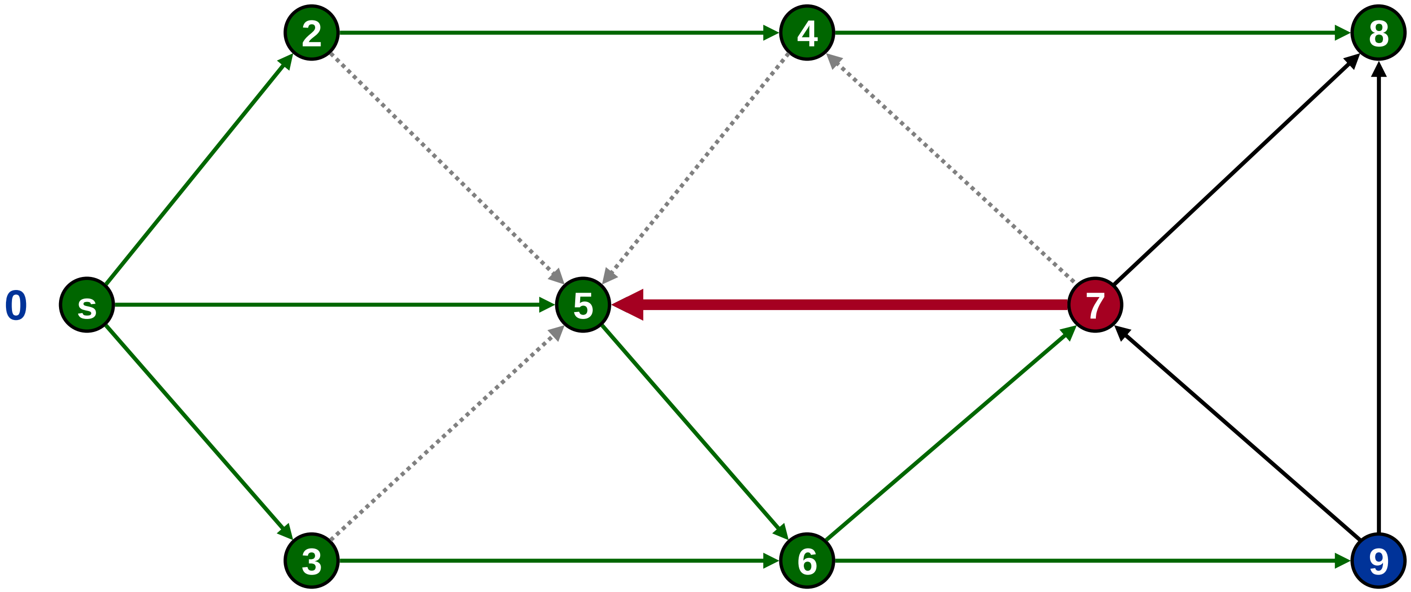
Discovered

Top of queue

Finished

Queue: 7 9

Breadth First Search



Undiscovered

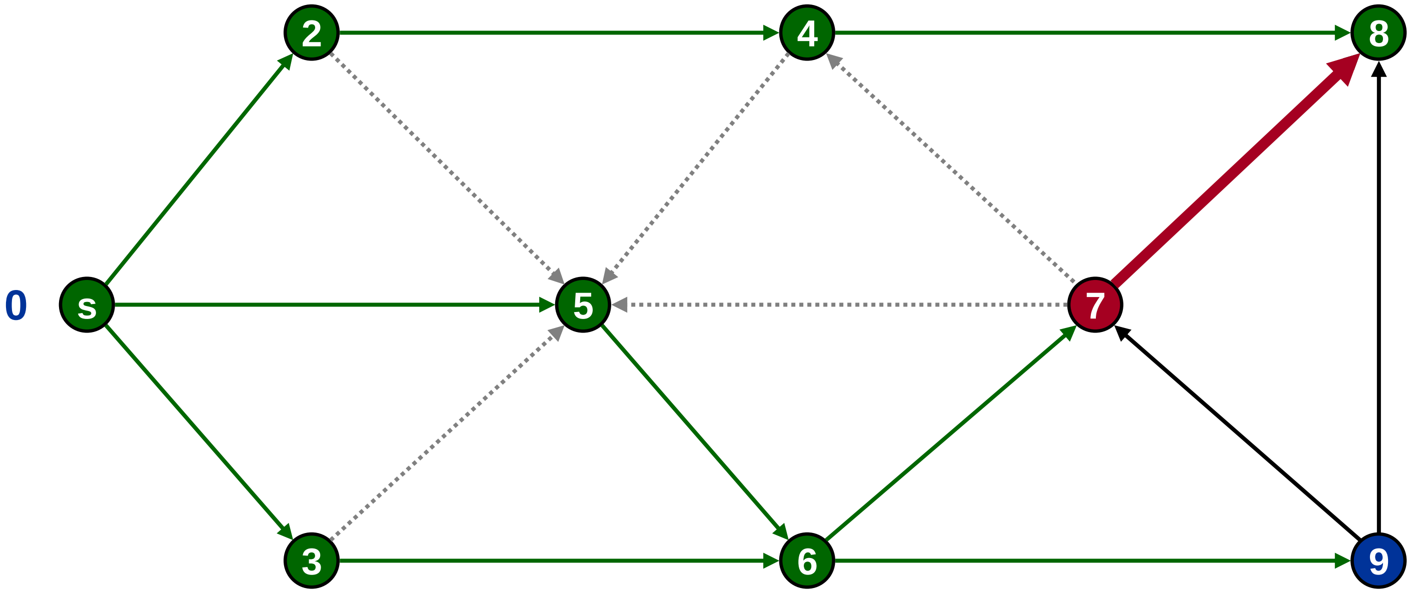
Discovered

Top of queue

Finished

Queue: 7 9

Breadth First Search



Undiscovered

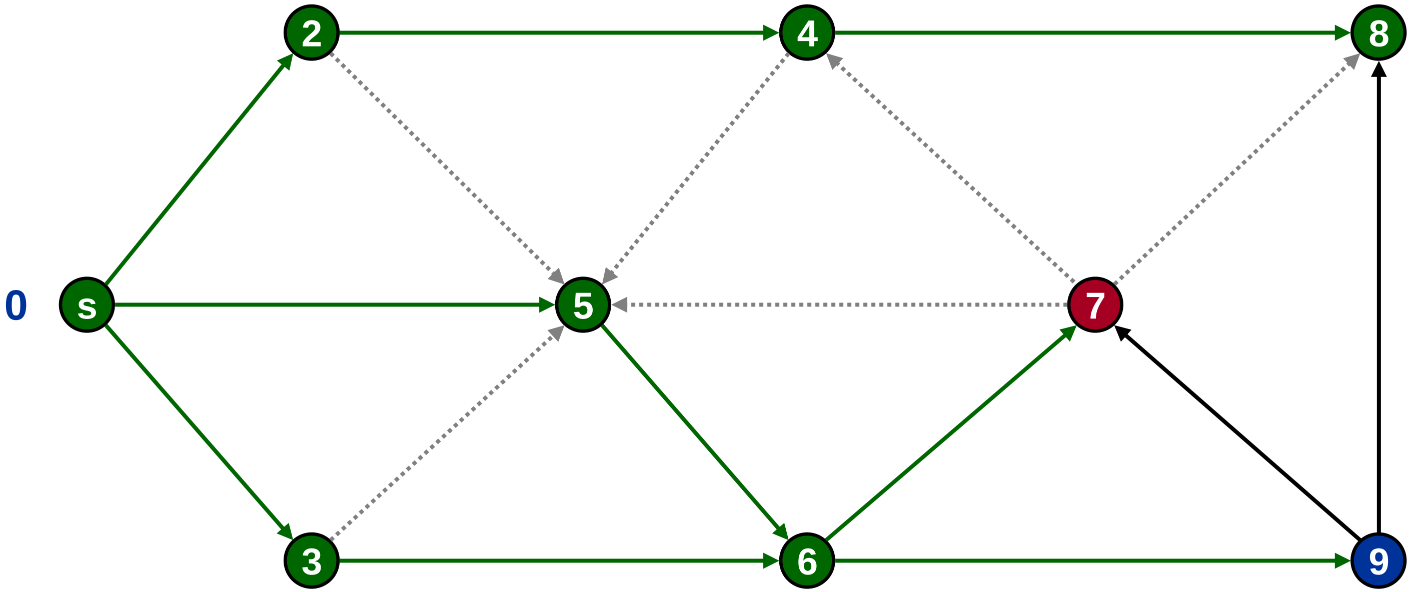
Discovered

Top of queue

Finished

Queue: 7 9

Breadth First Search



Undiscovered

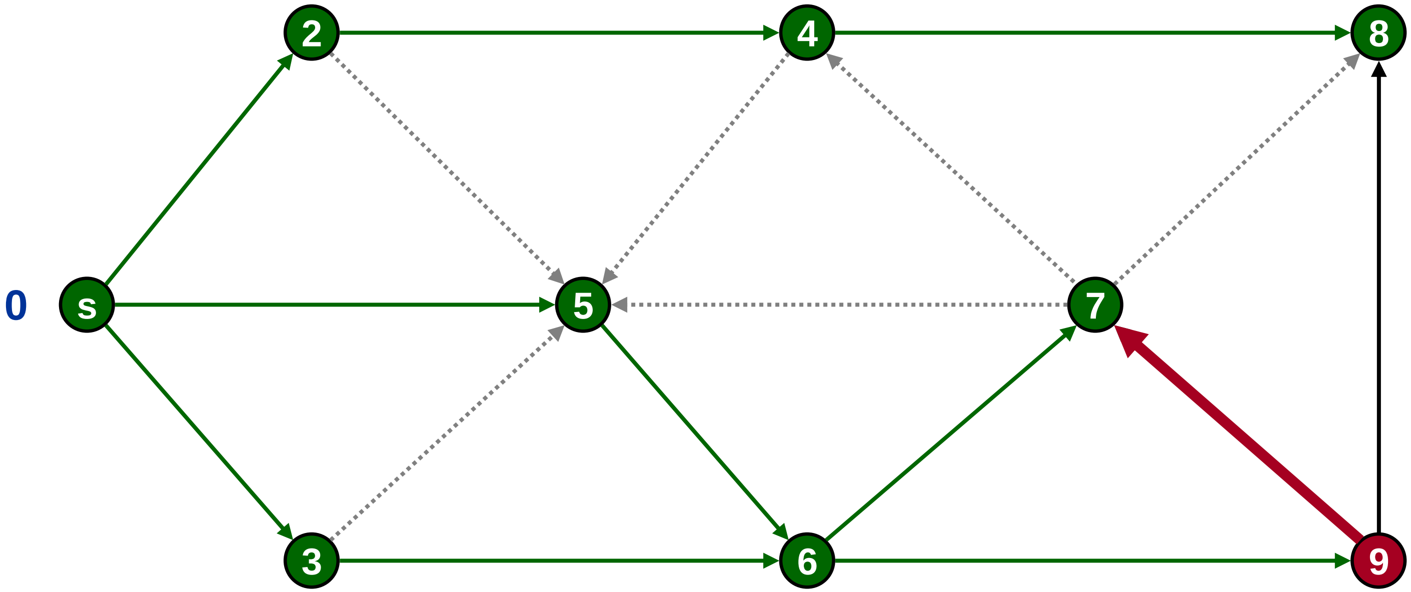
Discovered

Top of queue

Finished

Queue: 7 9

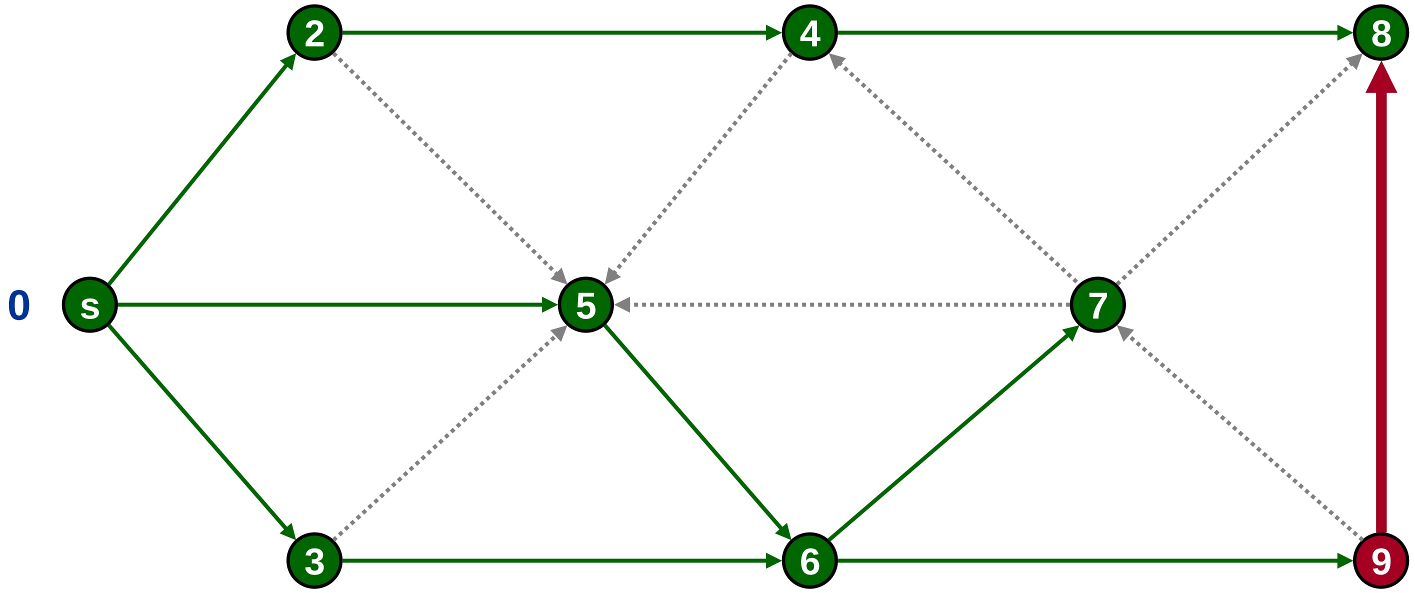
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 9

Breadth First Search



Undiscovered

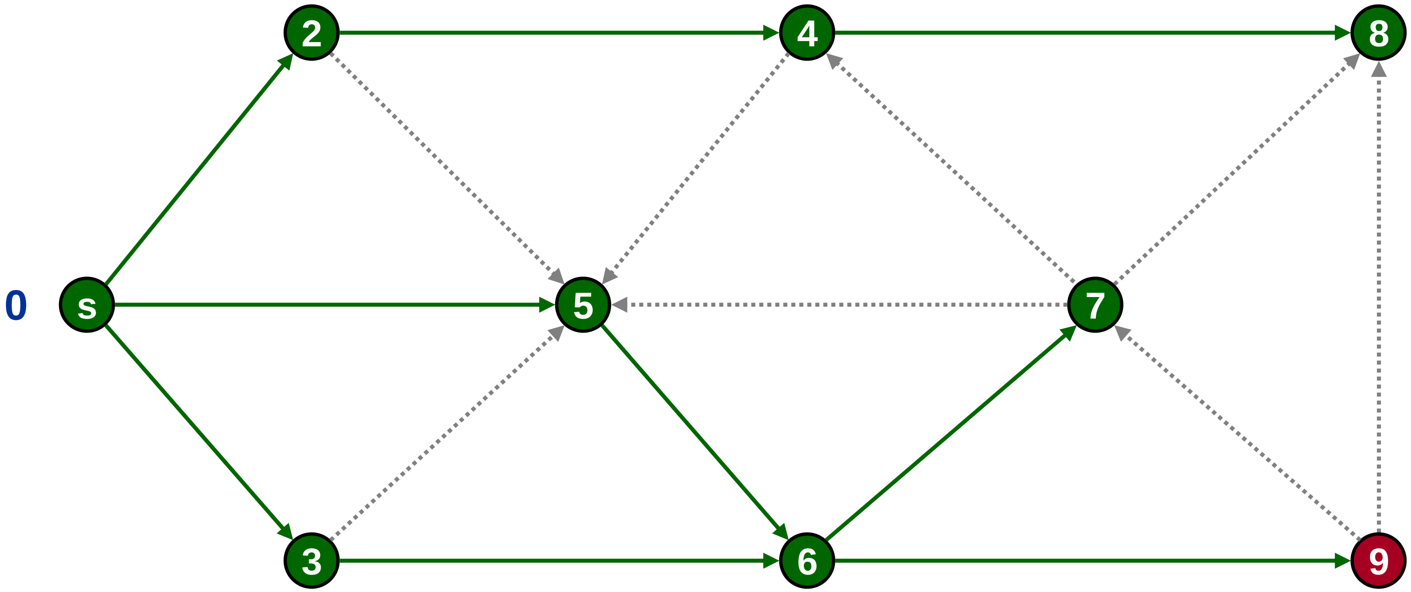
Discovered

Top of queue

Finished

Queue: 9

Breadth First Search



Undiscovered

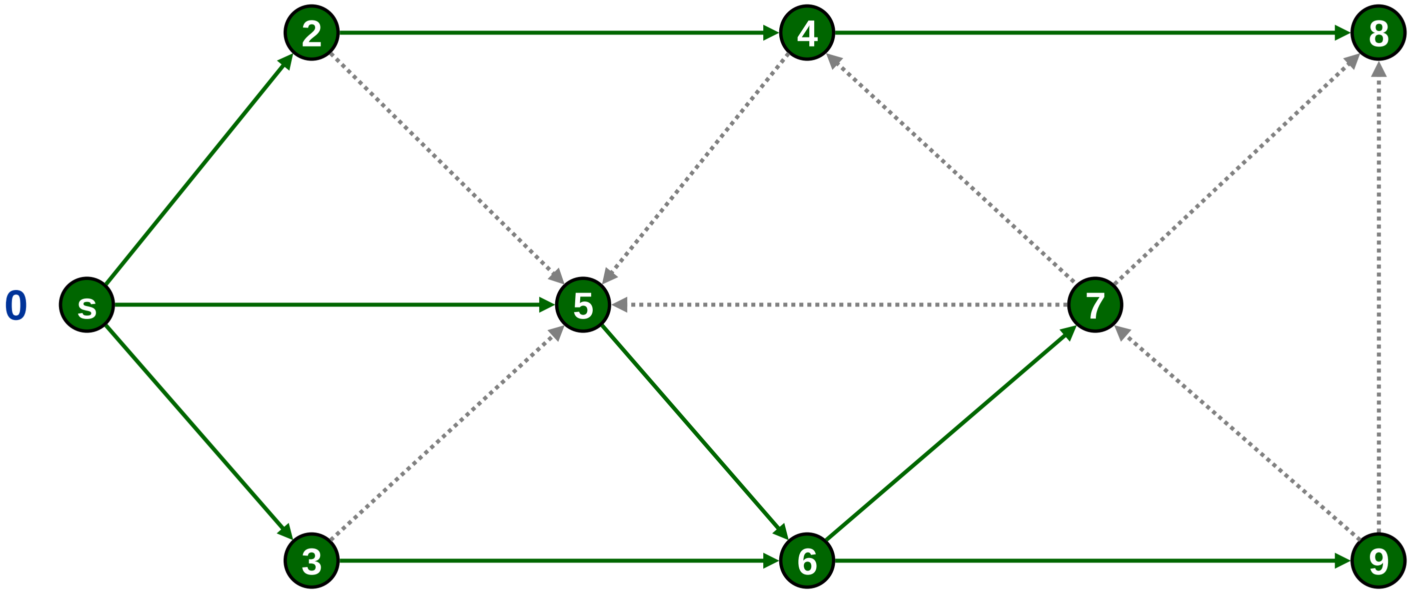
Discovered

Top of queue

Finished

Queue: 9

Breadth First Search



Undiscovered

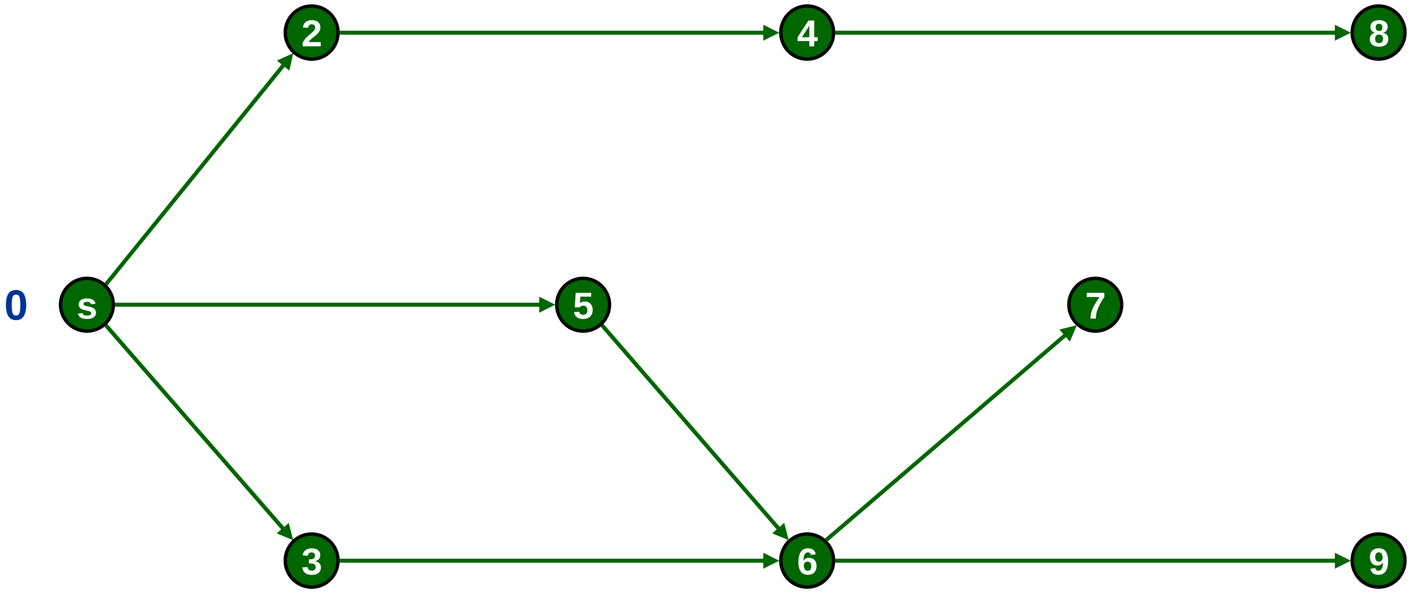
Discovered

Top of queue

Finished

Queue:

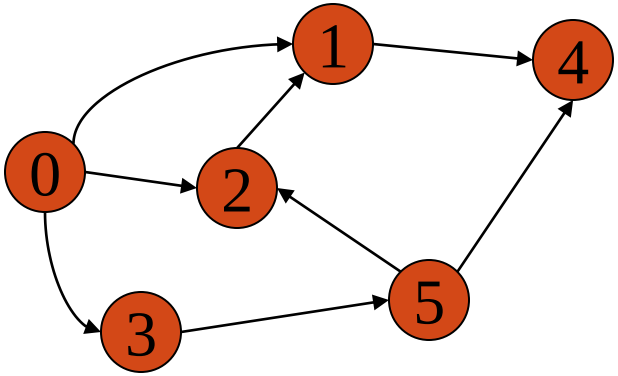
Breadth First Search



Level Graph

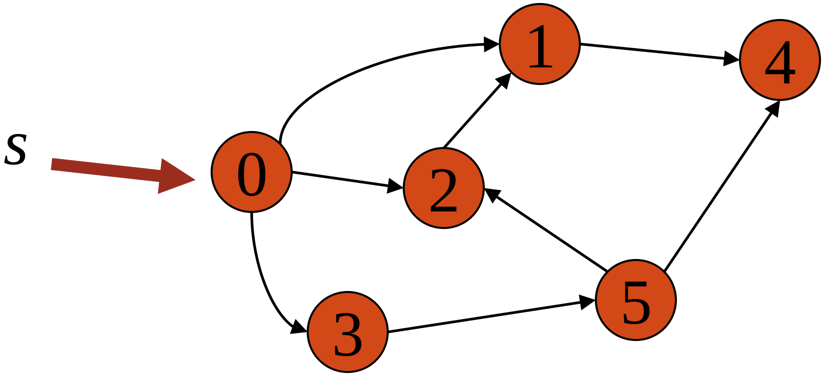
Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
      $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



Breadth First Search Algorithm

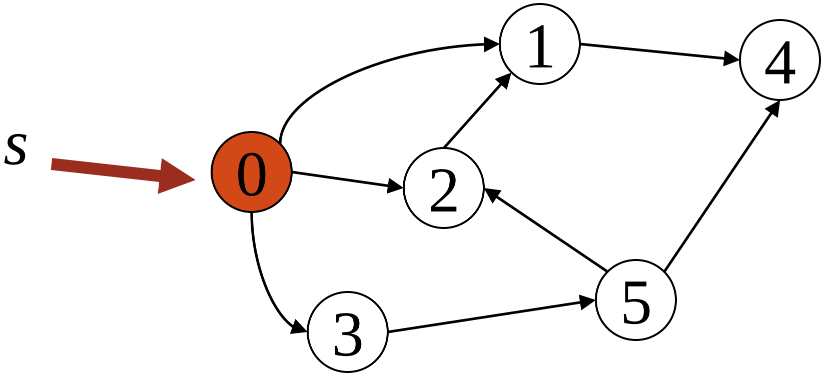
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q = \varnothing$

Breadth First Search Algorithm

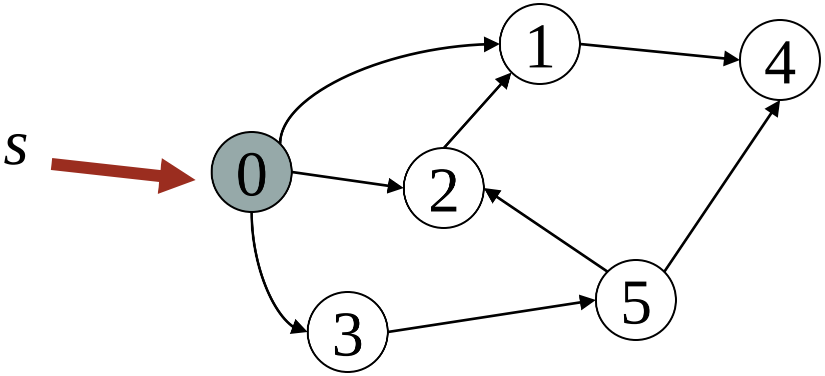
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q = \varnothing$

Breadth First Search Algorithm

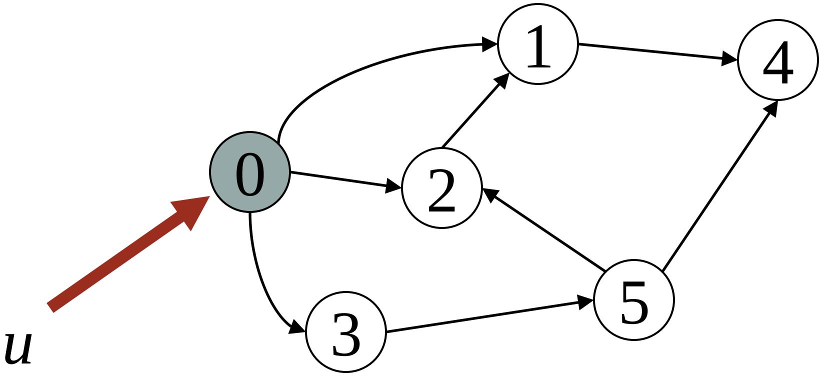
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
     $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
     $u \leftarrow head[Q];$
     for each $v \in Adjacent[u]$
              if $color[v] = white$
              {
                   $color[v] \leftarrow gray$
                   Enqueue$(Q,v)$
              }
     Dequeue$(Q)$
     $color[u] \leftarrow black;$
}
```



Q = 0

Breadth First Search Algorithm

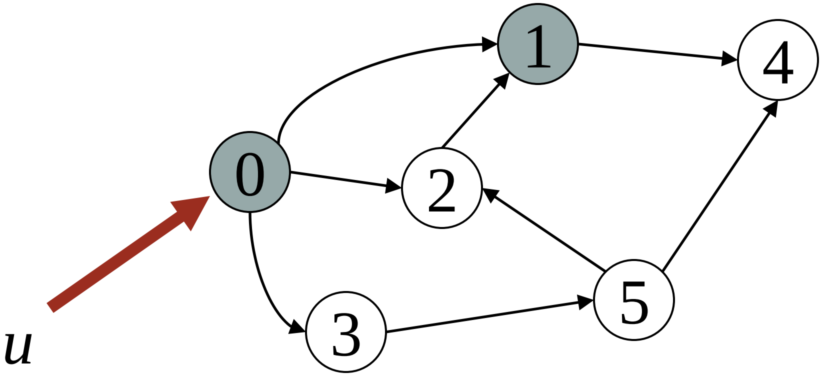
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
      $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
    {        if $color[v] = white$
             {
                   $color[v] \leftarrow gray$
                   Enqueue$(Q,v)$
             } }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q = \boxed{0}$

Breadth First Search Algorithm

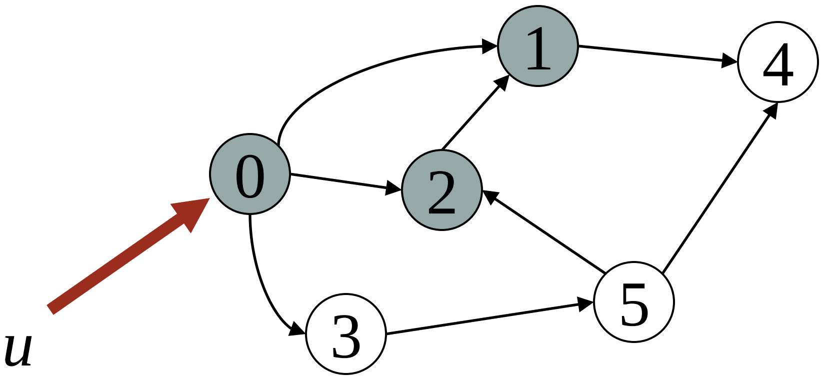
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q =$ | 0 | 1 |

Breadth First Search Algorithm

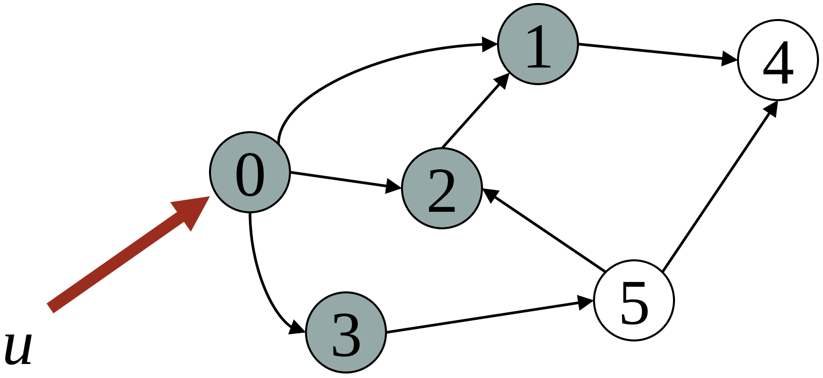
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q =$ | 0 | 1 | 2 |

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

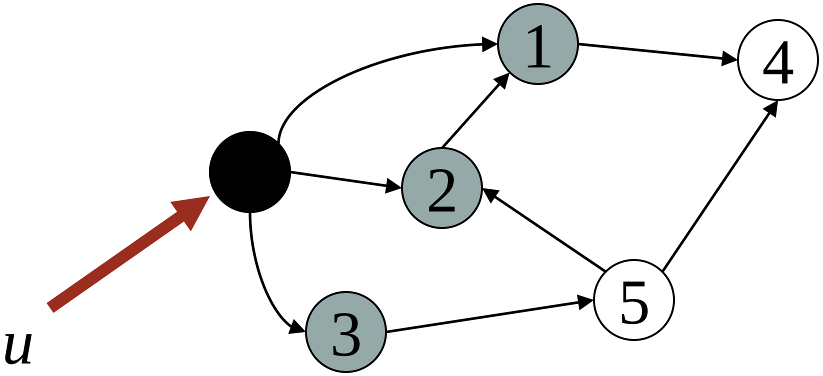


$Q =$

0	1	2	3

Breadth First Search Algorithm

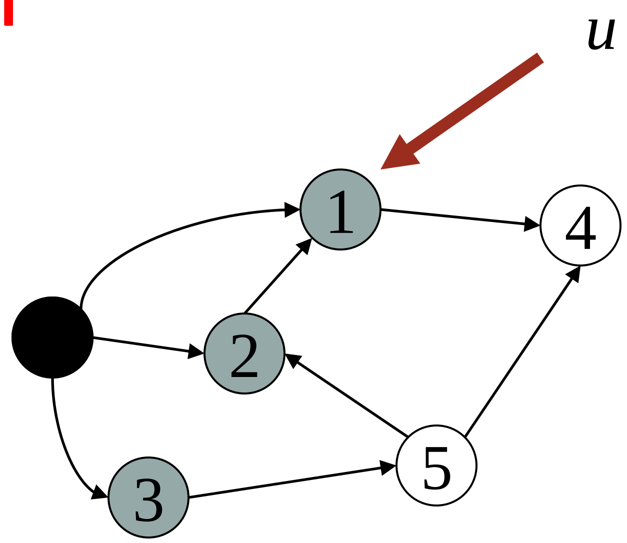
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



Q = | 1 | 2 | 3 |

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

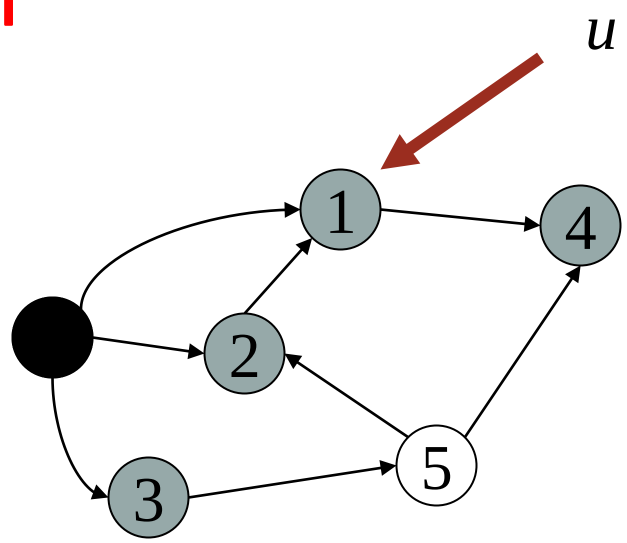


$Q =$

1	2	3

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

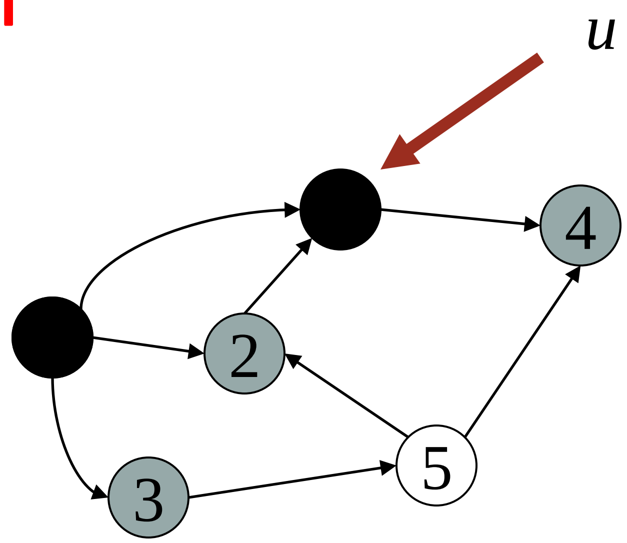


$Q =$

1	2	3	4

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

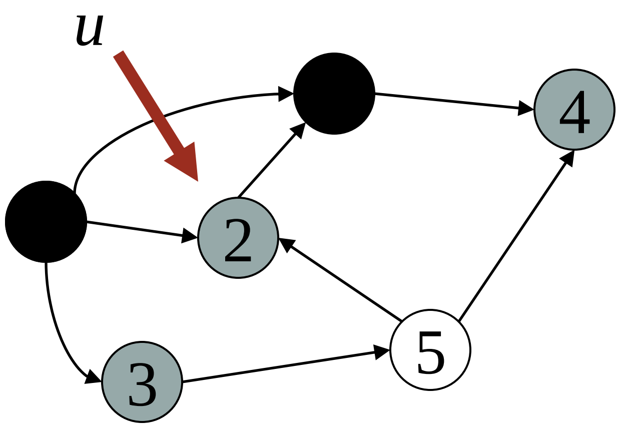


$Q =$

2	3	4

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

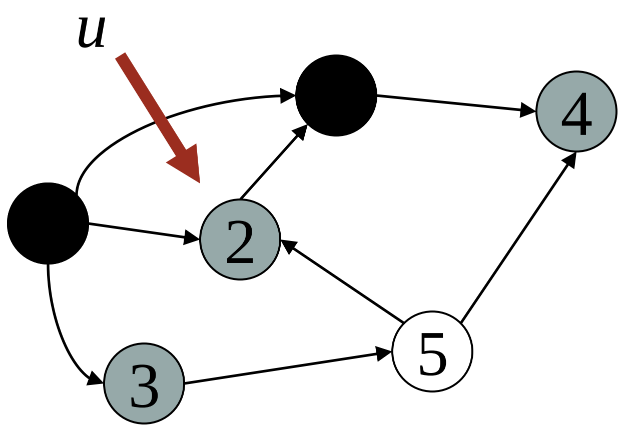


$Q =$

2	3	4

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

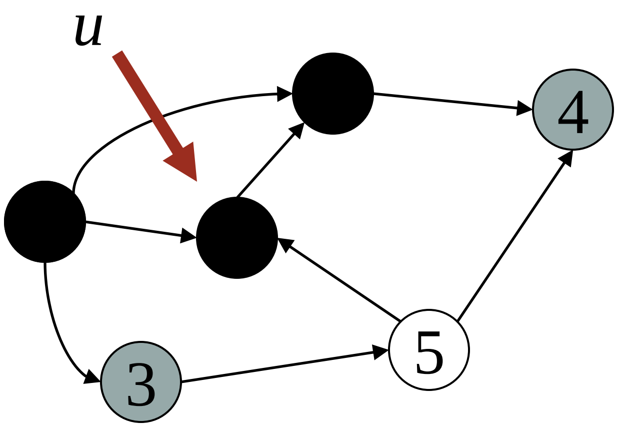


$Q =$

2	3	4

Breadth First Search Algorithm

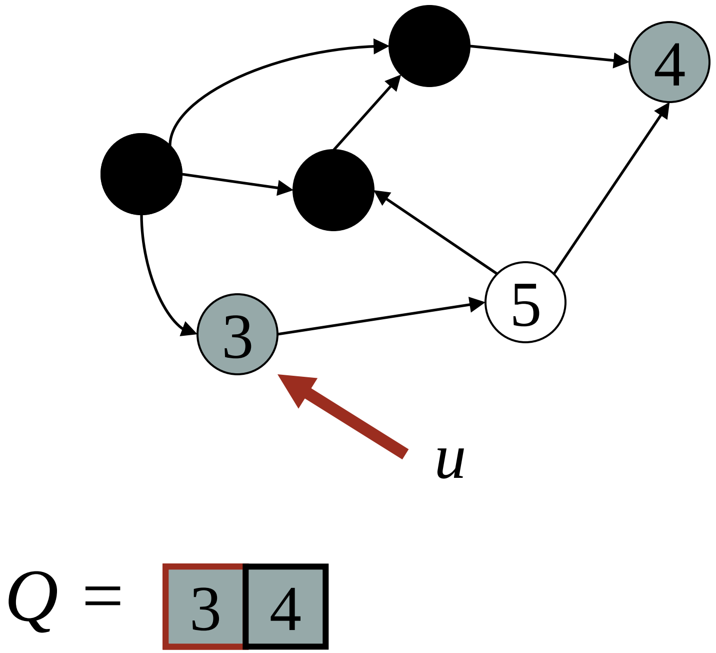
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q =$ | 3 | 4 |

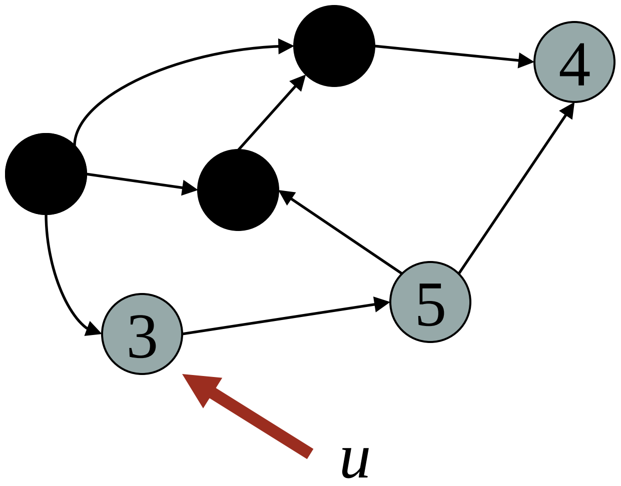
Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u]$ ← $white$
$color[s]$ ← $gray$
$Q$ ← $\{s\}$
While $Q \neq \emptyset$
{
    $u$ ← $head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v]$ ← $gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u]$ ← $black;$
}
```



Breadth First Search Algorithm

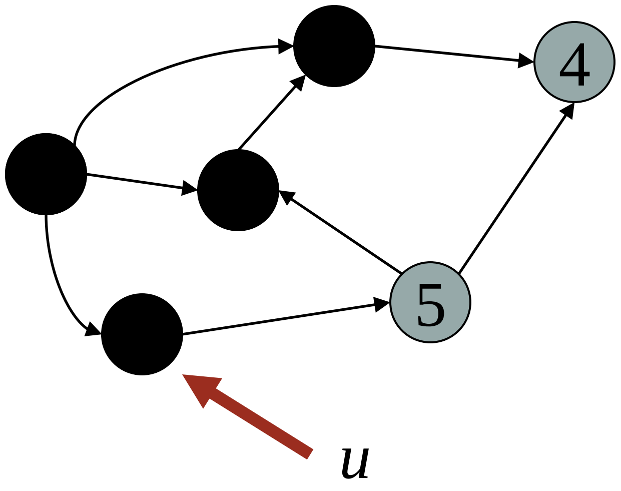
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



Q = | 3 | 4 | 5 |

Breadth First Search Algorithm

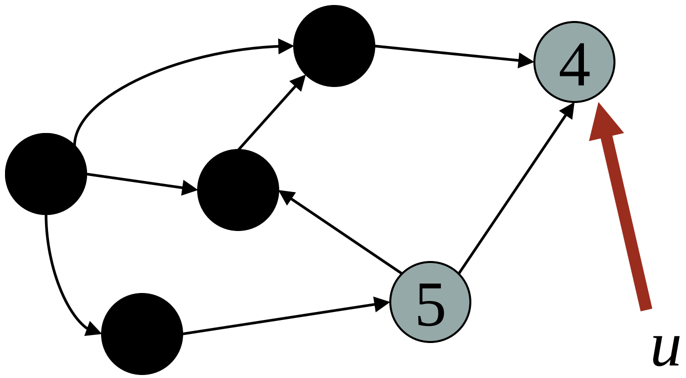
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```

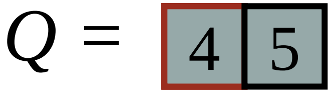


$Q = \boxed{4}\boxed{5}$

Breadth First Search Algorithm

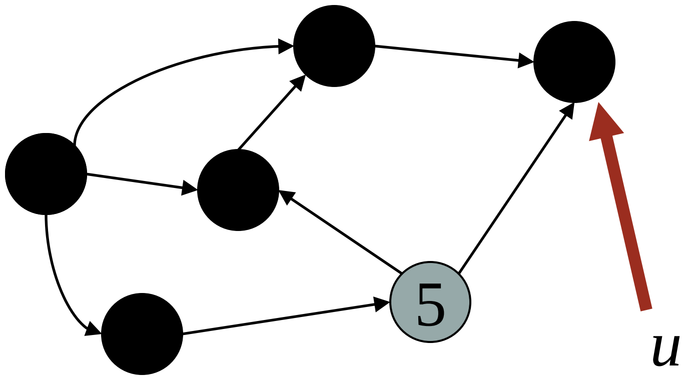
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
        if $color[v] = white$
        {
            $color[v] \leftarrow gray$
            Enqueue$(Q,v)$
        }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```





Breadth First Search Algorithm

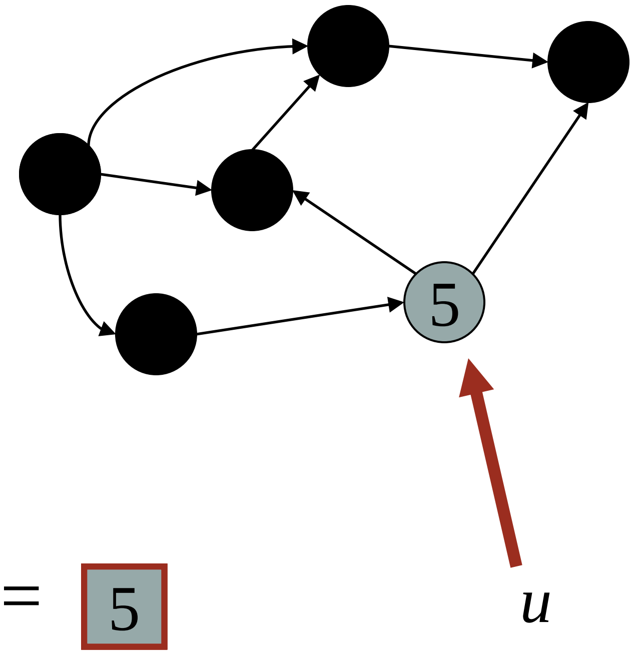
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q = \boxed{5}$

Breadth First Search Algorithm

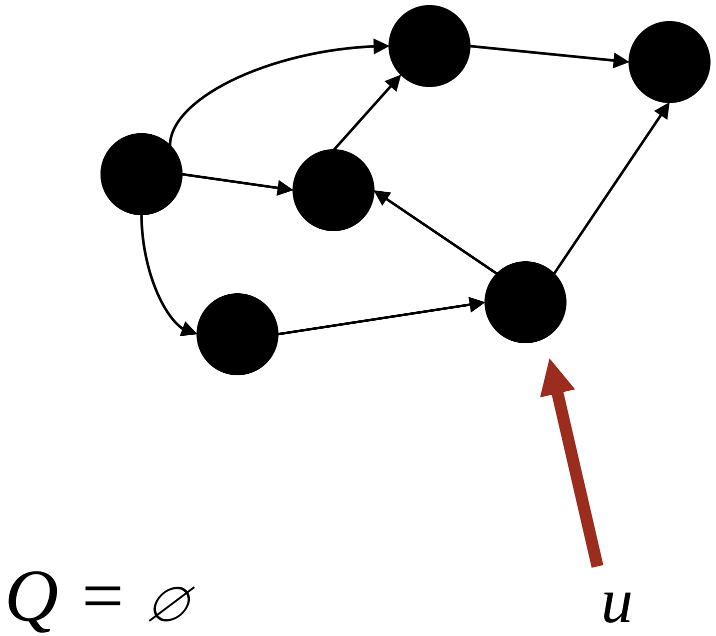
```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \emptyset$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```



$Q = \boxed{5}$

Breadth First Search Algorithm

```
Given graph $G=(V,E)$ and source vertex $s \in V$
Create a queue Q
For each vertex $u \in V - \{s\}$
    $color[u] \leftarrow white$
$color[s] \leftarrow gray$
$Q \leftarrow \{s\}$
While $Q \neq \varnothing$
{
    $u \leftarrow head[Q];$
    for each $v \in Adjacent[u]$
            if $color[v] = white$
            {
                $color[v] \leftarrow gray$
                Enqueue$(Q,v)$
            }
    Dequeue$(Q)$
    $color[u] \leftarrow black;$
}
```


$Q = \varnothing$
u