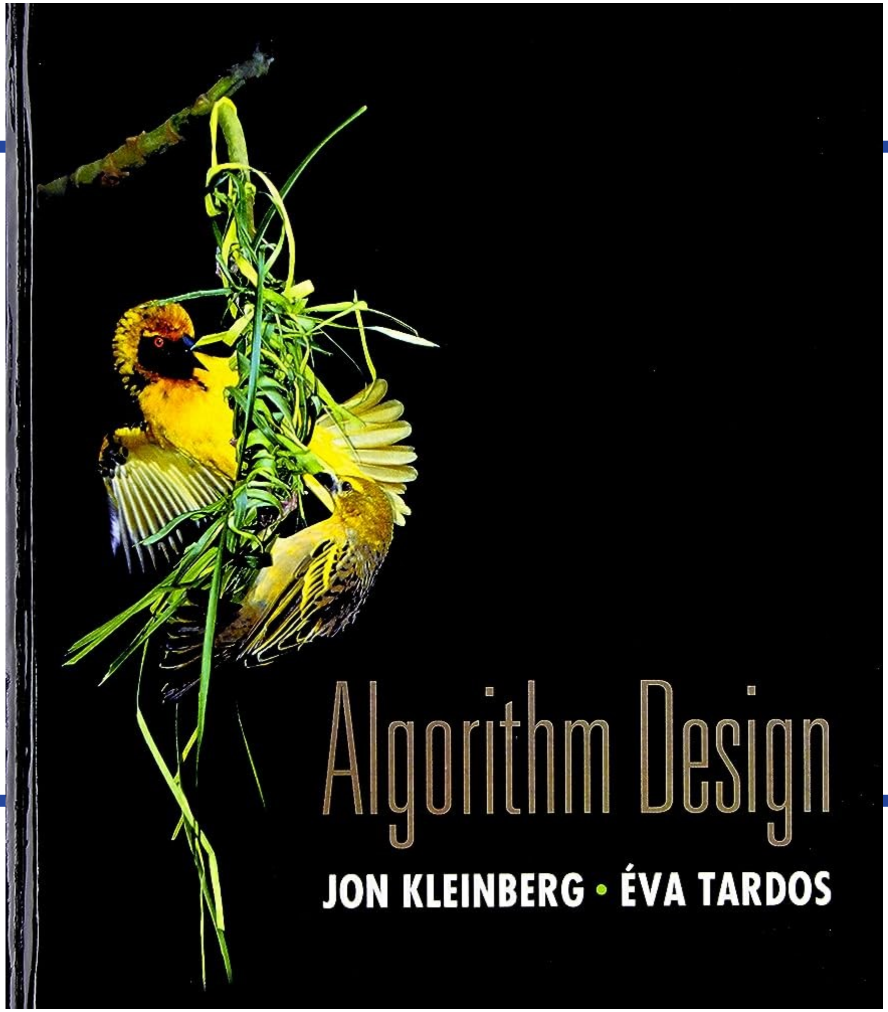
CS 310: Algorithms

Lecture 6

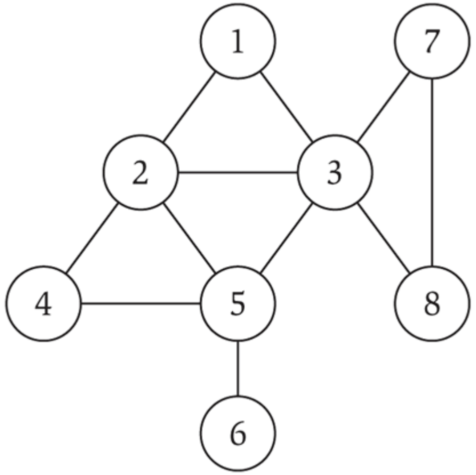
Instructor: Naveed Anwar Bhatti



Chapter 3:
Graphs

Section 3.2:
Graph Traversal

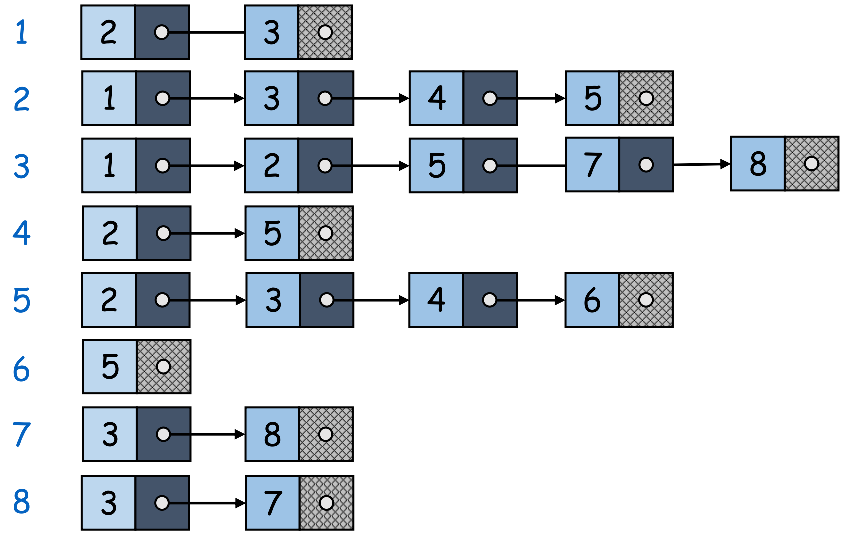
Last time: Graph Representation



Adjacency Matrix

	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Adjacency List



1: 2 → 3

2: 1 → 3 → 4 → 5

3: 1 → 2 → 5 → 7 → 8

4: 2 → 5

5: 2 → 3 → 4 → 6

6: 5

7: 3 → 8

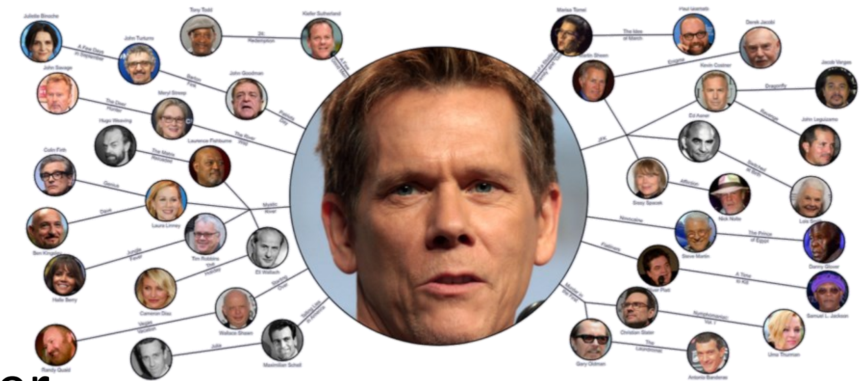
8: 3 → 7

Connectivity

- **s-t connectivity problem**: Given two node **s** and **t**, is there a path between **s** and **t**?

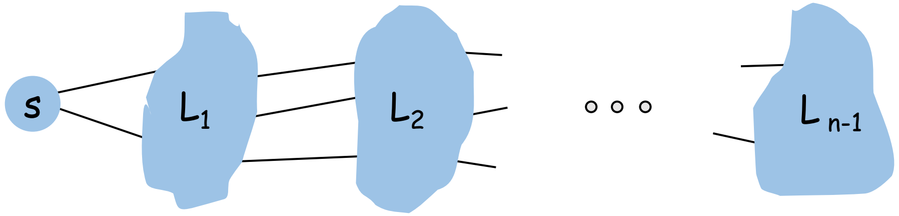
- **s-t shortest path problem:** Given two node s and t, what is the length of the shortest path between s and t?

- Applications.
  - Google Maps in f
  - Maze traversal
  - Kevin Bacon number
  - Fewest number of hops in a communication network



Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.

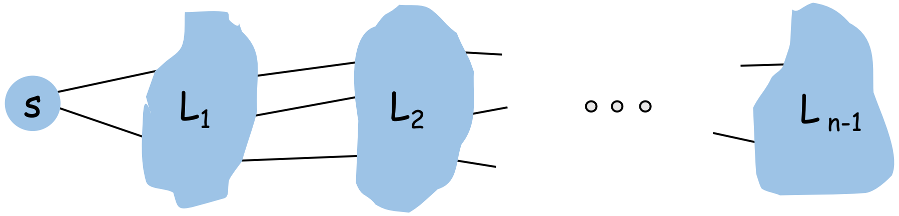


- BFS algorithm:
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

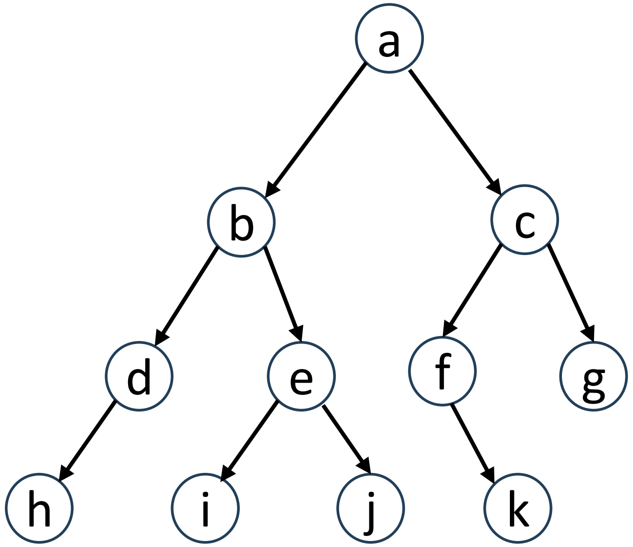
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***i*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

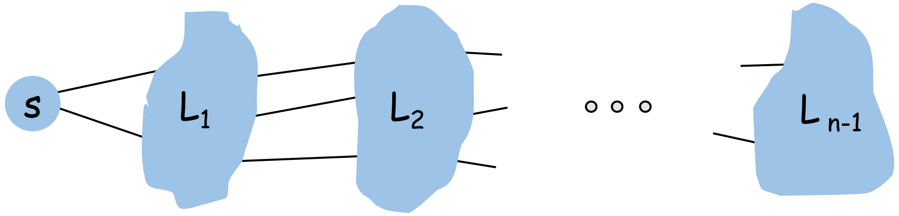


Breath First Search

Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

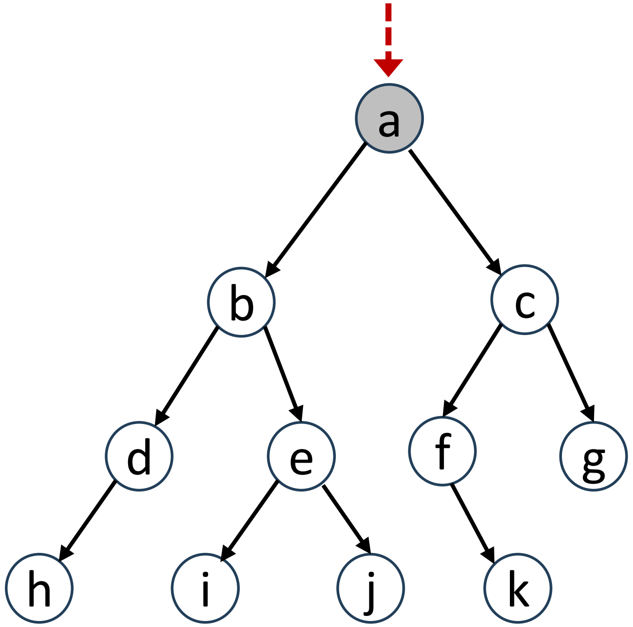
Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

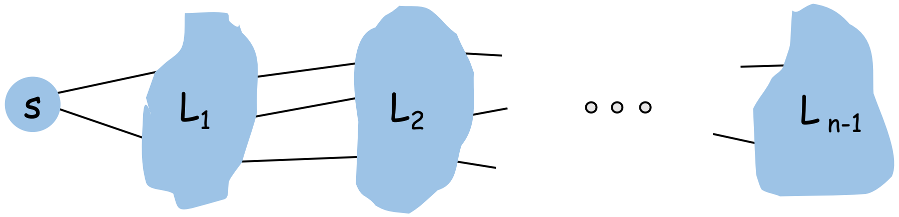
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***I*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.



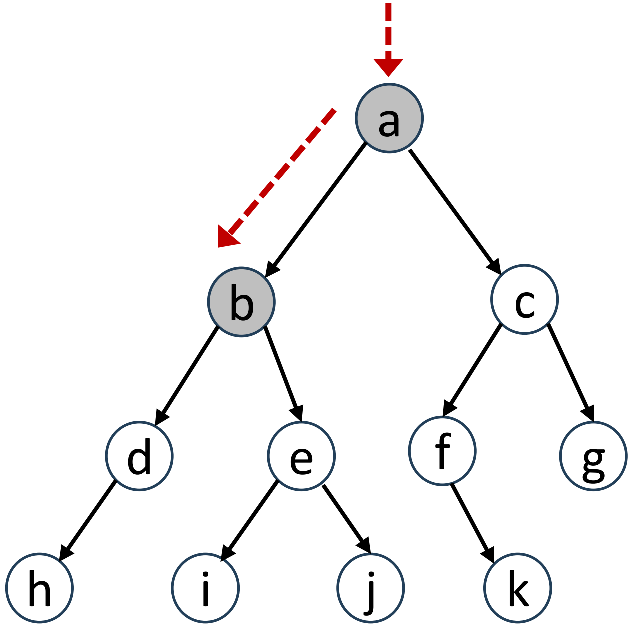
Breath First Search

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

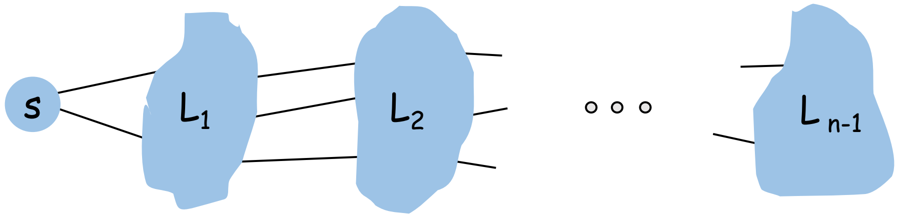


Breath First Search

Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

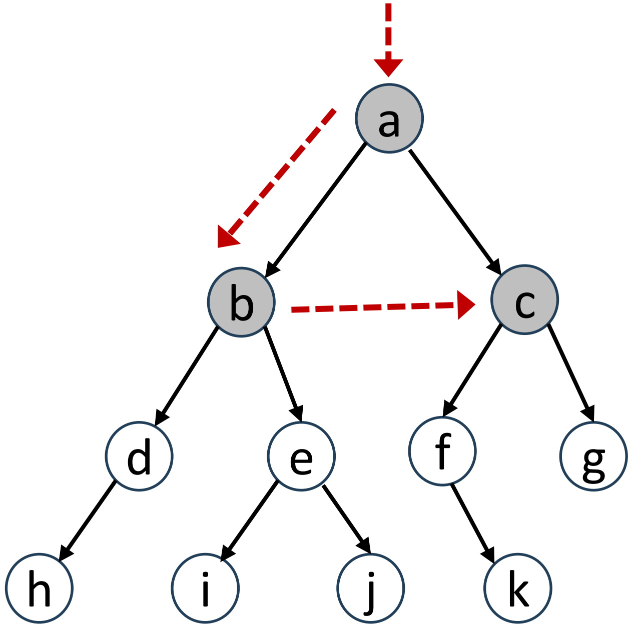
Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

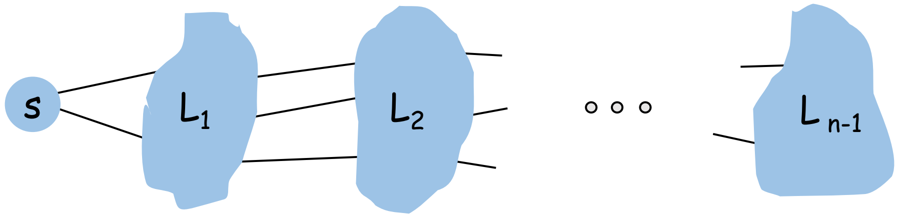
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.



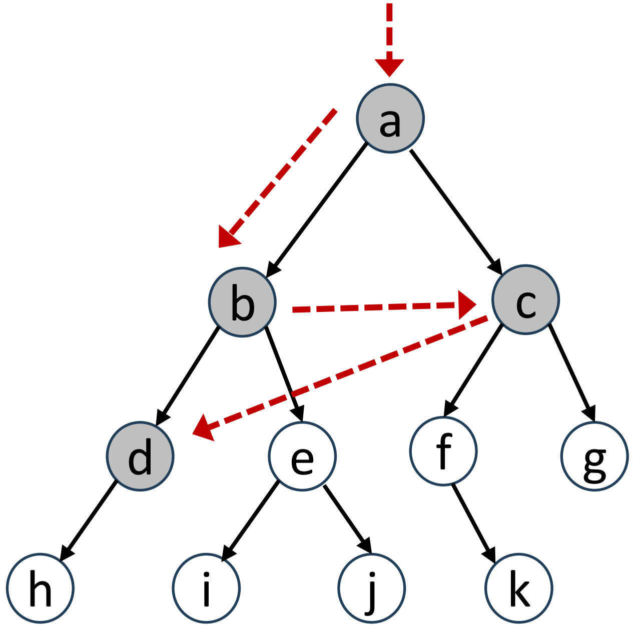
Breath First Search

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

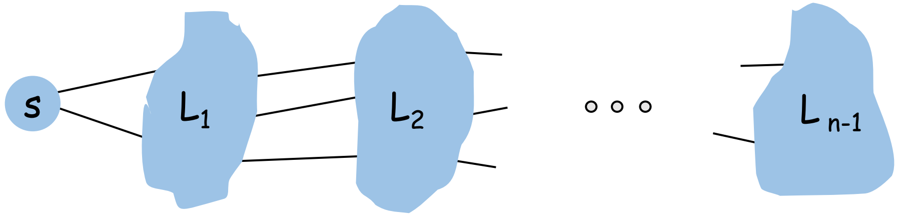


Breath First Search

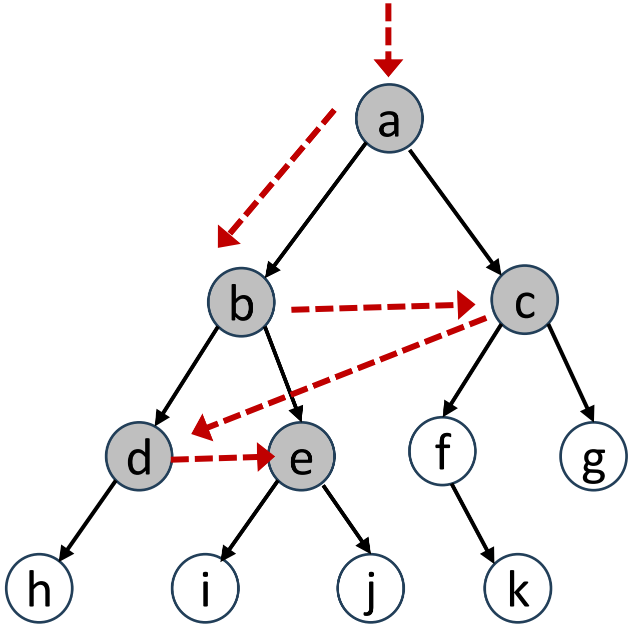
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

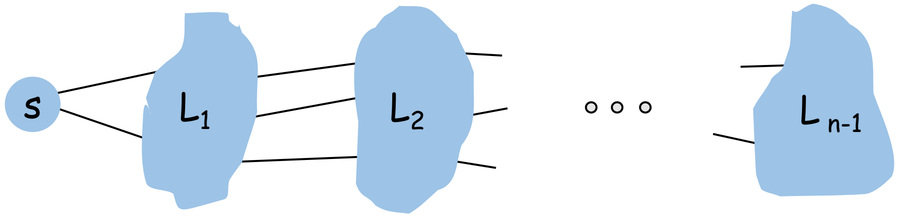


Breath First Search

Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

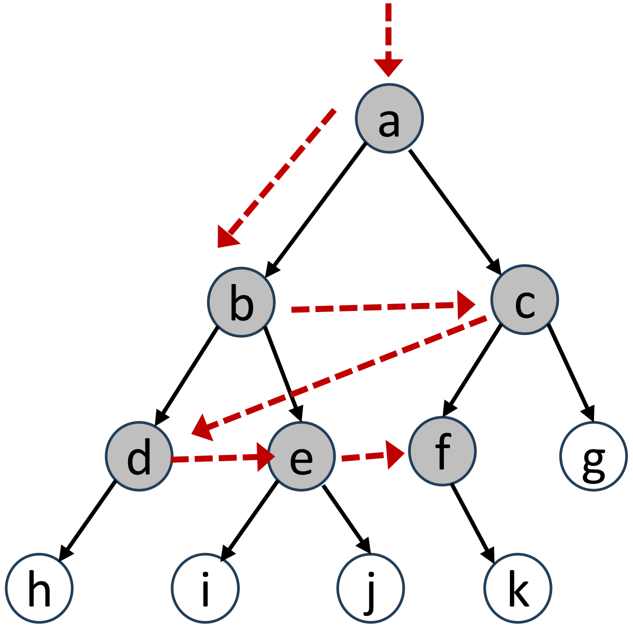
Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - $L_0 = \{ s \}$.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

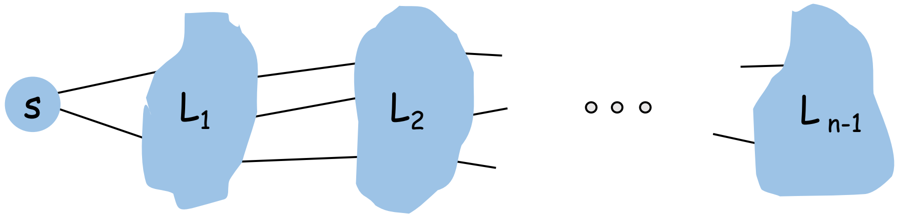
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.



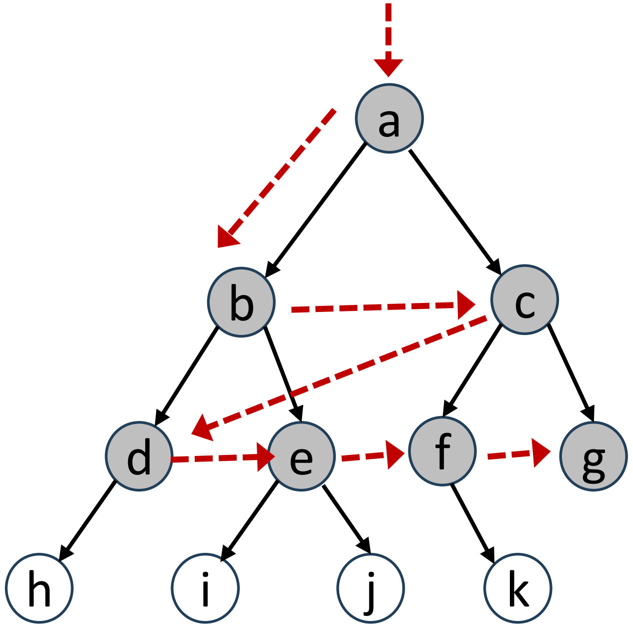
Breath First Search

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

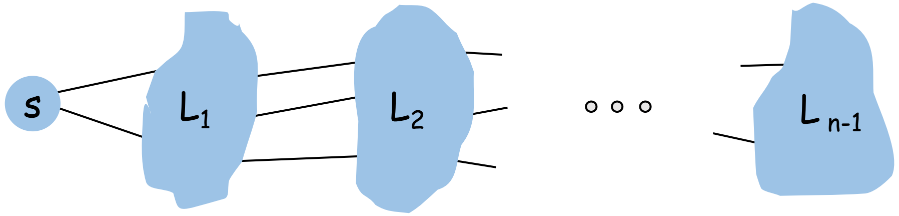


Breath First Search

Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

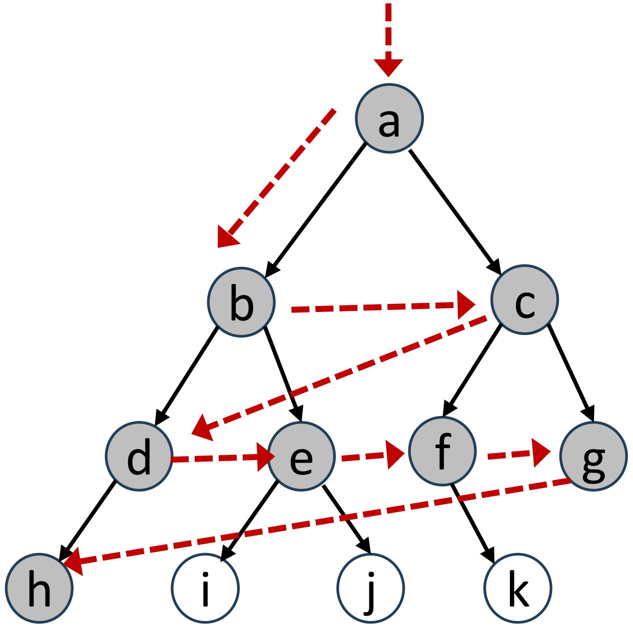
Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

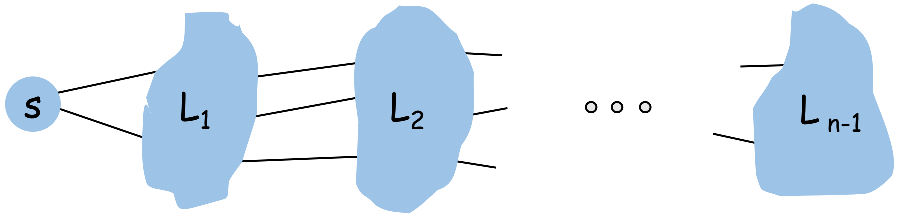
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.



Breath First Search

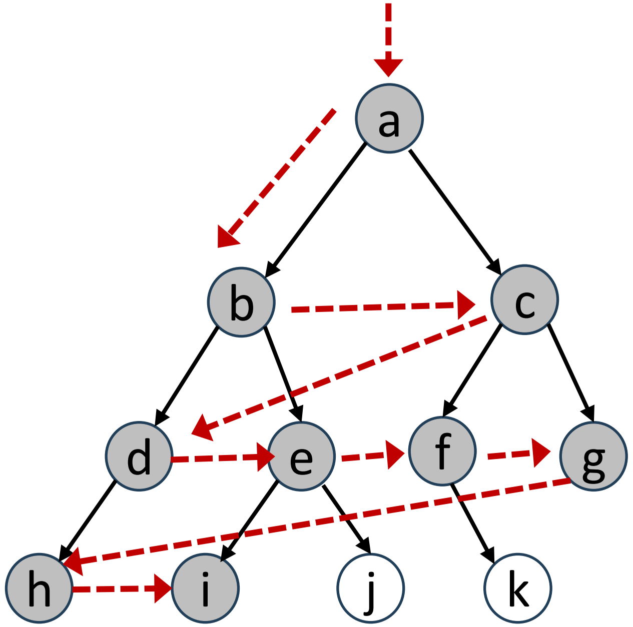
Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

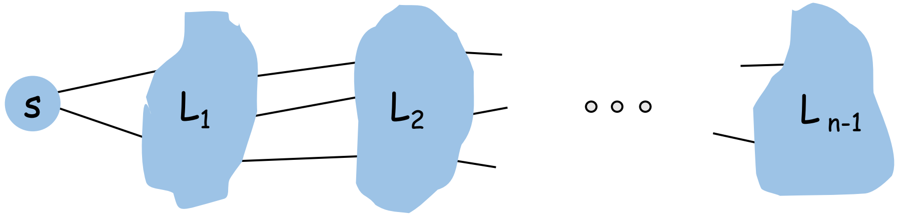
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.



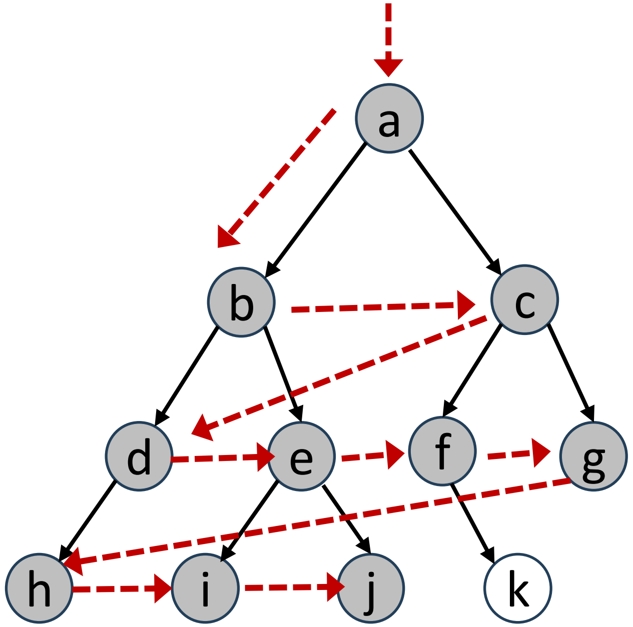
Breath First Search

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.

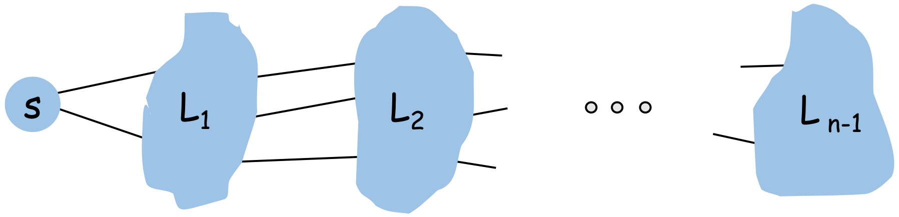


Breath First Search

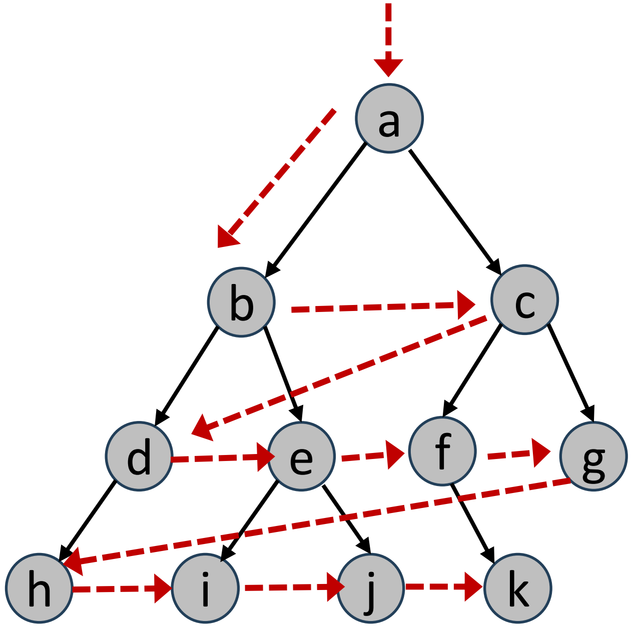
Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

Breadth First Search

- BFS intuition: Explore outward from ***s*** in all possible directions, adding nodes **one "layer" at a time**.



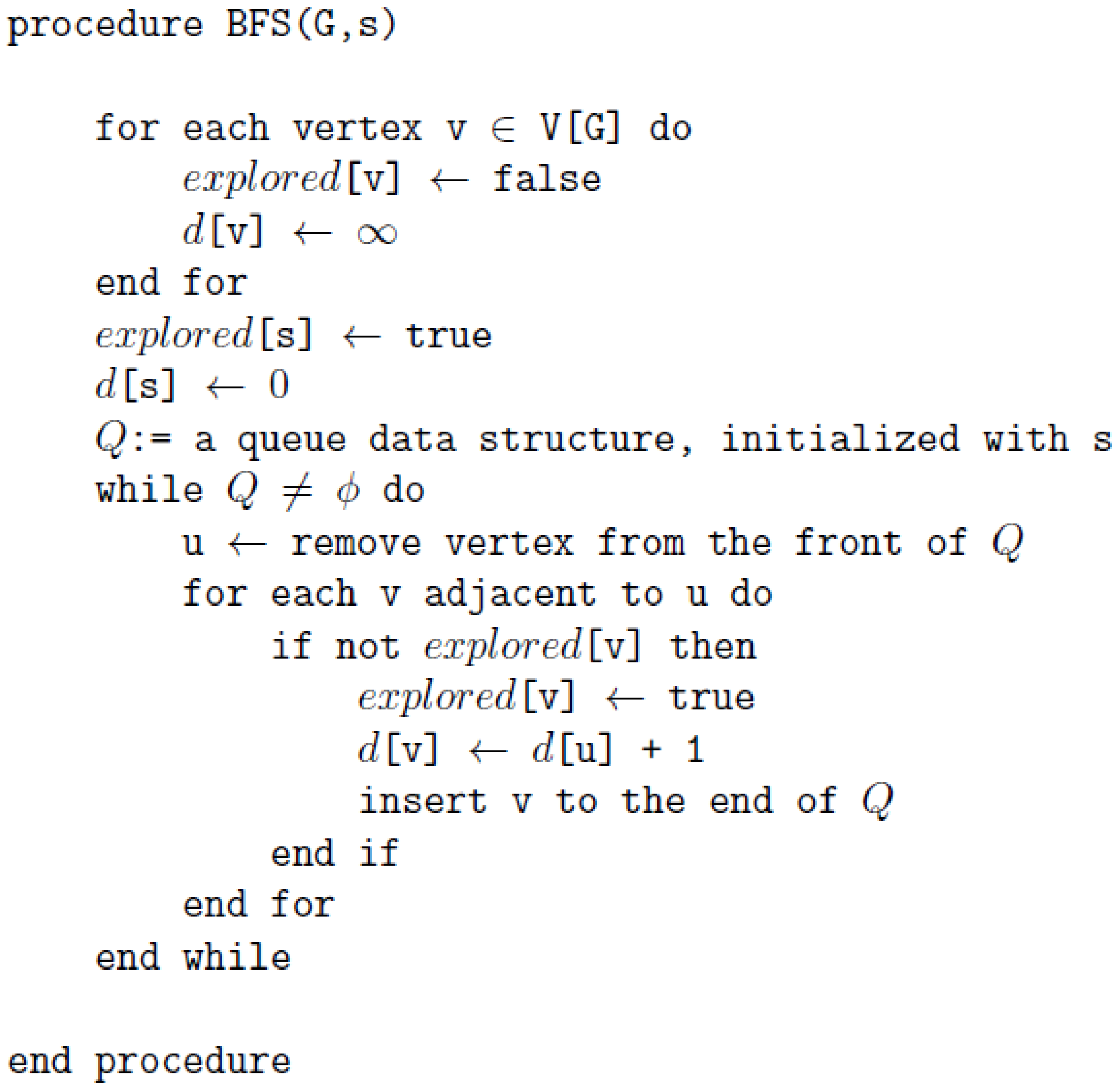
- BFS algorithm.
  - L_0 = { s }.
  - L_1 = all neighbors of L_0.
  - L_2 = all nodes that do not belong to L_0 or L_1, and that have an edge to a node in L_1.
  - L_{i+1} = all nodes that do not belong to an earlier layer, and that have an edge to a node in L_i.



Breath First Search

Theorem: For each ***i***, $\mathbf{L_i}$ consists of all nodes at distance exactly ***l*** from **s**. There is a path from **s** to **t** iff **t** appears in some layer.

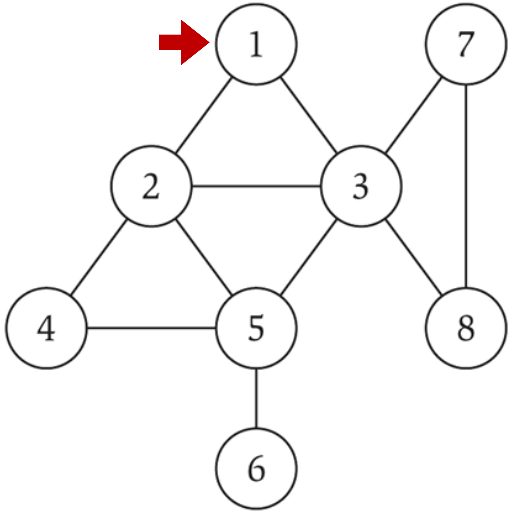
Breadth First Search



```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ $0$
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```

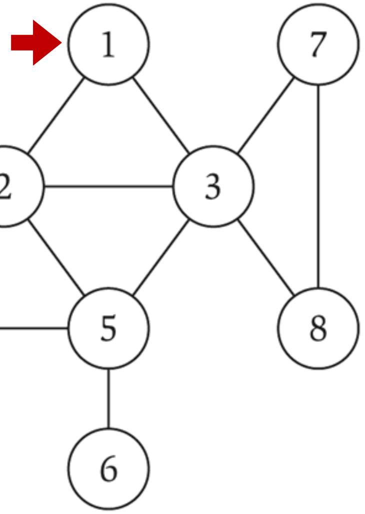
Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ $0$
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```

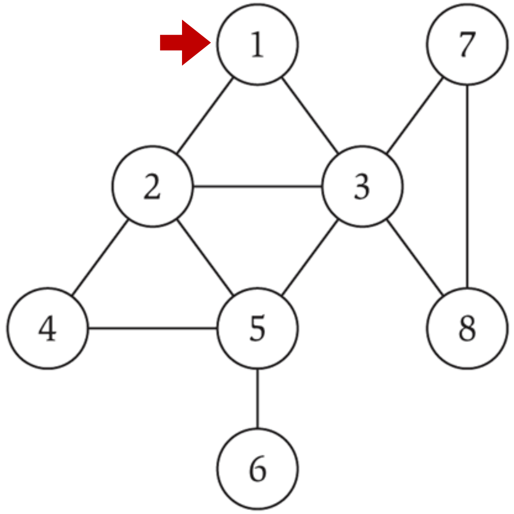


Explored

Vertex	Value
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

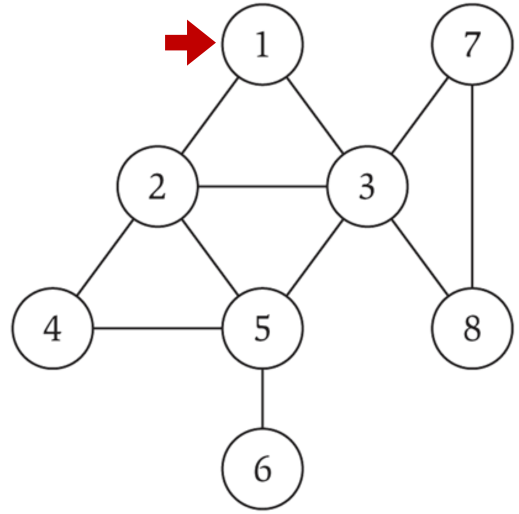
Vertex	Value
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	∞
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← $\infty$
➡   end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

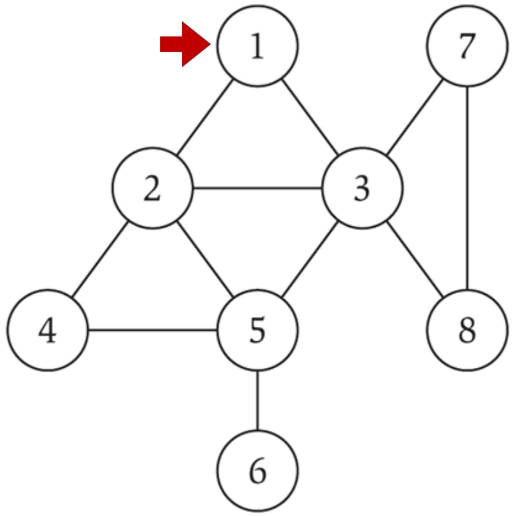
Vertex	Value
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	∞
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
  ➡ $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

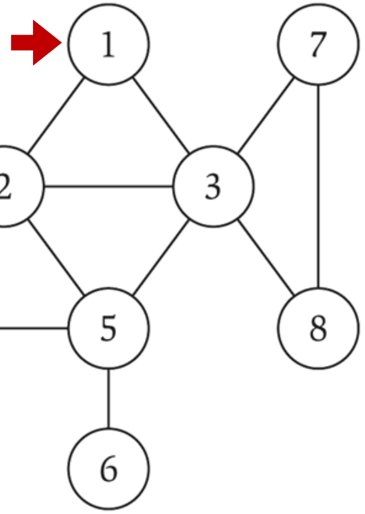
Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	∞
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
➡   $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

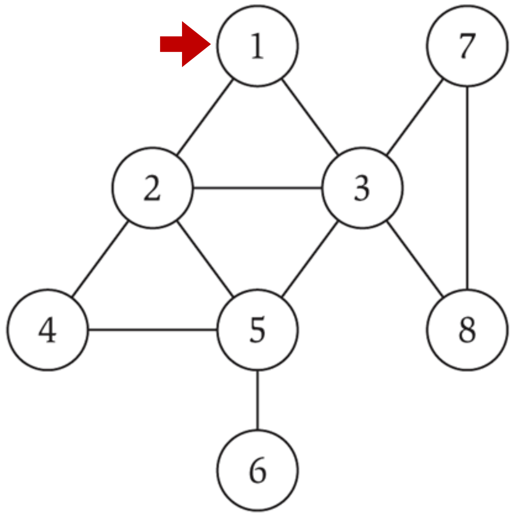
Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
  ➡ $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

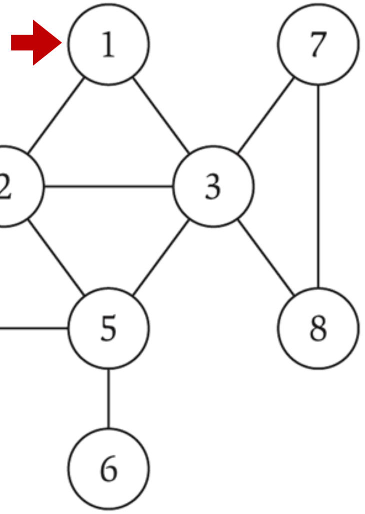
Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

Vertex
1

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
➡   while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

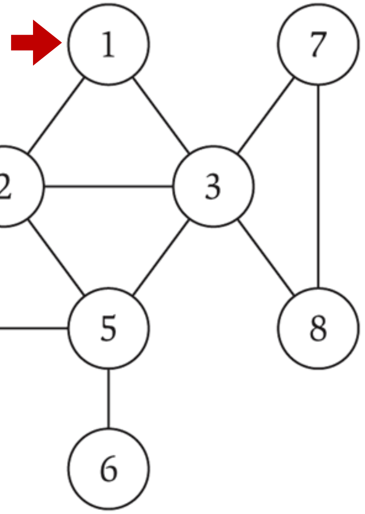
Queue

Vertex
1

Breadth First Search

U
1

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
➡       u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

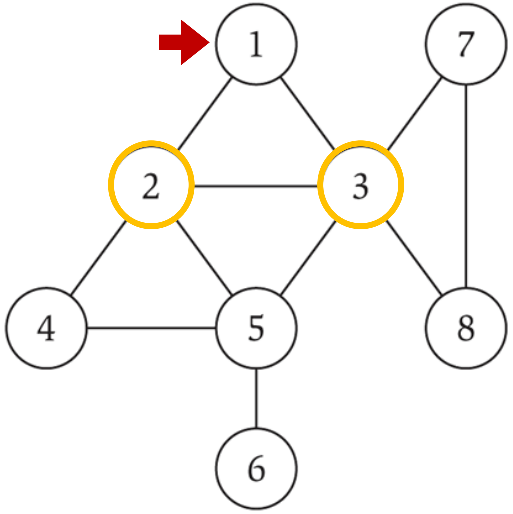
Queue

Vertex

Breadth First Search

U	V
1	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

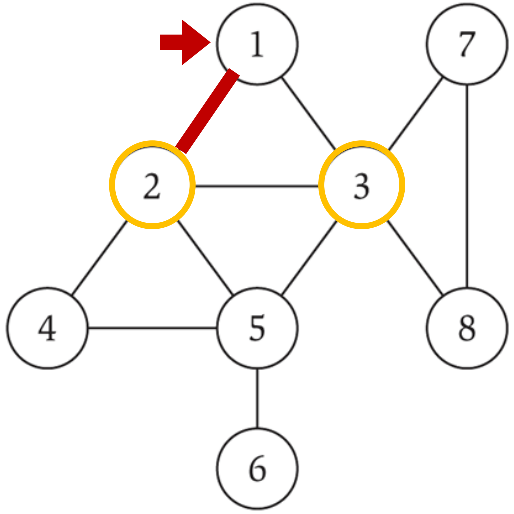
Queue

Vertex

Breadth First Search

U	V
1	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	False
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

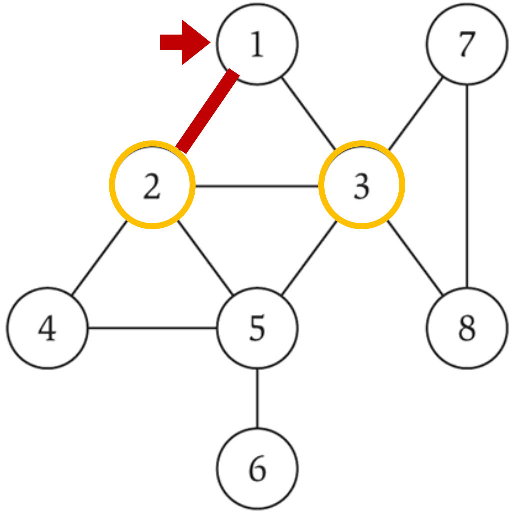
Vertex

Breadth First Search

U
1

V
2

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← $\infty$
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
➡               $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

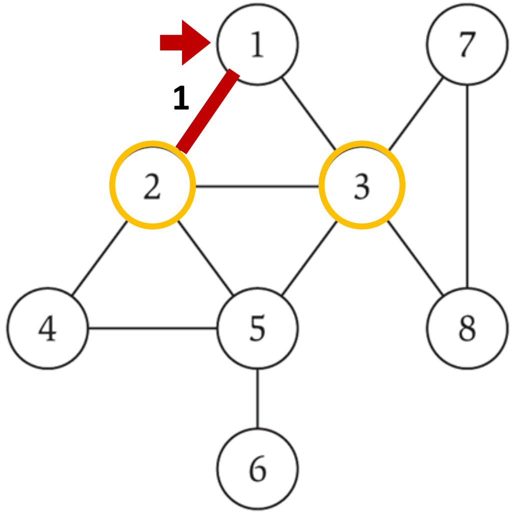
Queue

Vertex

Breadth First Search

U	V
1	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

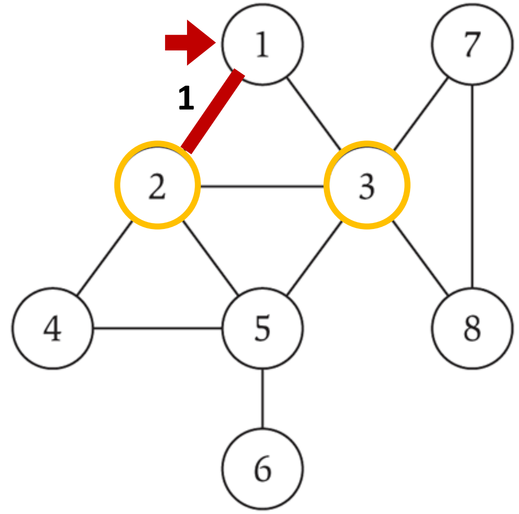
Queue

Vertex

Breadth First Search

U	V
1	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

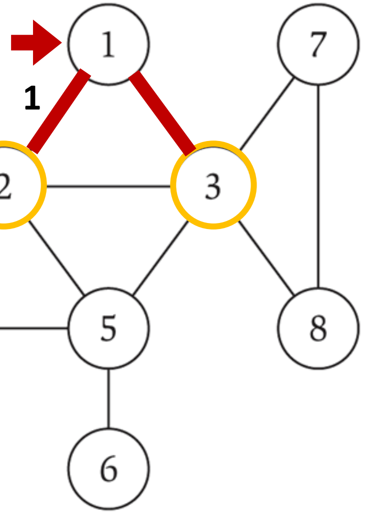
Vertex
2

Breadth First Search

U
1

V
3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	False
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

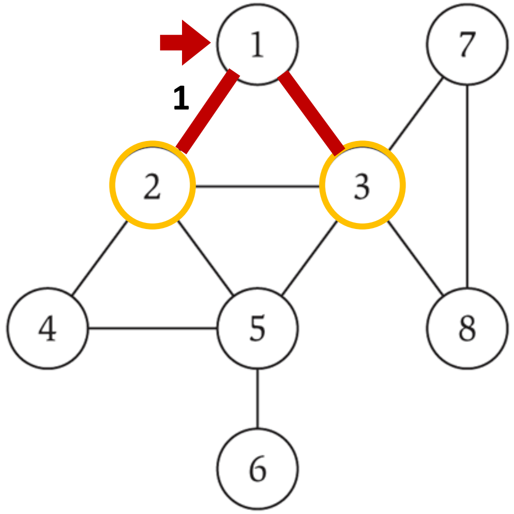
Queue

Vertex
2

Breadth First Search

U	V
1	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
➡               $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞

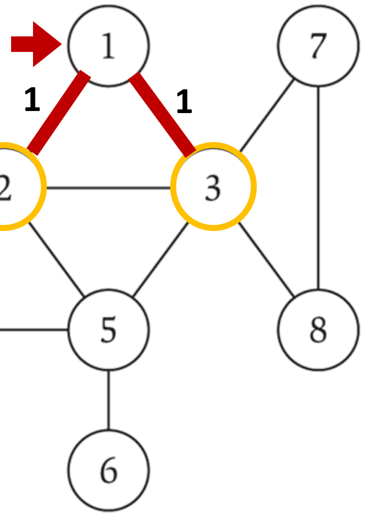
Queue

Vertex
2

Breadth First Search

U	V
1	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

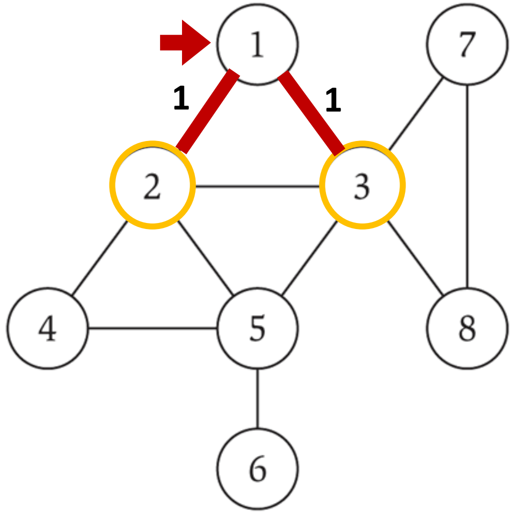
Queue

Vertex
2

Breadth First Search

U	V
1	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

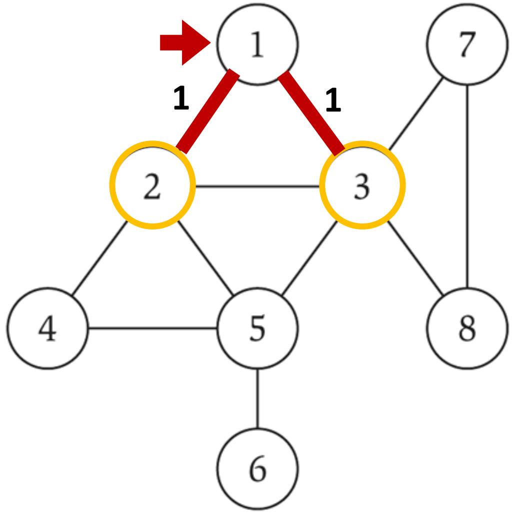
Queue

Vertex
2
3

Breadth First Search

U	V
1	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

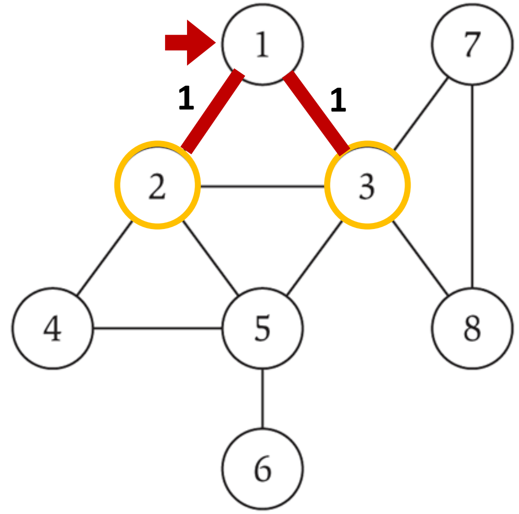
Queue

Vertex
2
3

Breadth First Search

U	V
1	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

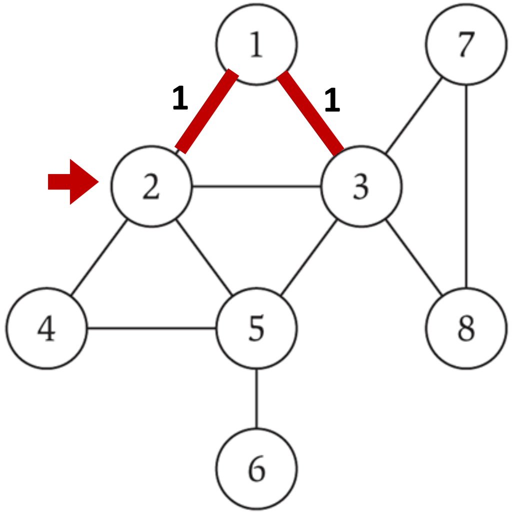
Queue

Vertex
2
3

Breadth First Search

U	V
2	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
 ➡      u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

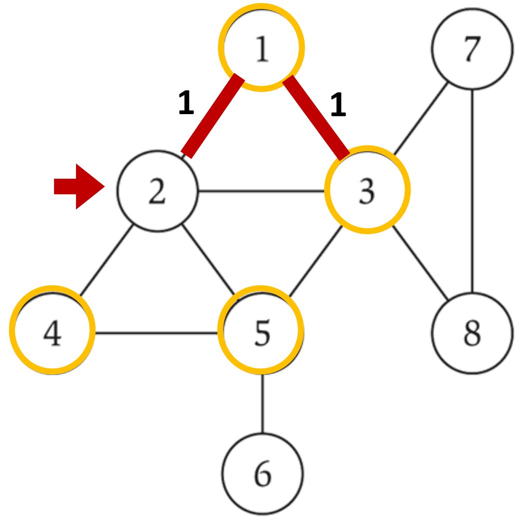
Queue

Vertex
3

Breadth First Search

U	V
2	

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

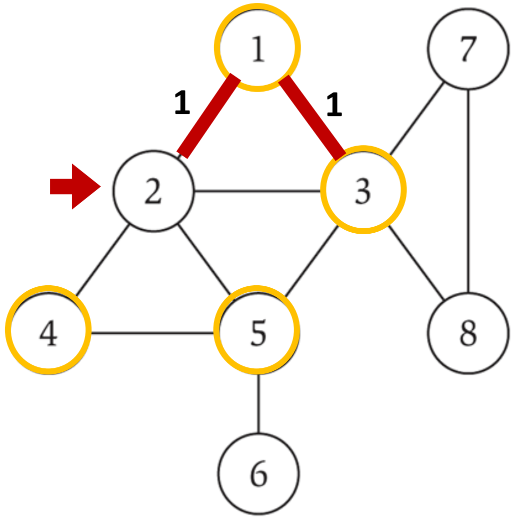
Queue

Vertex
3

Breadth First Search

U	V
2	1

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

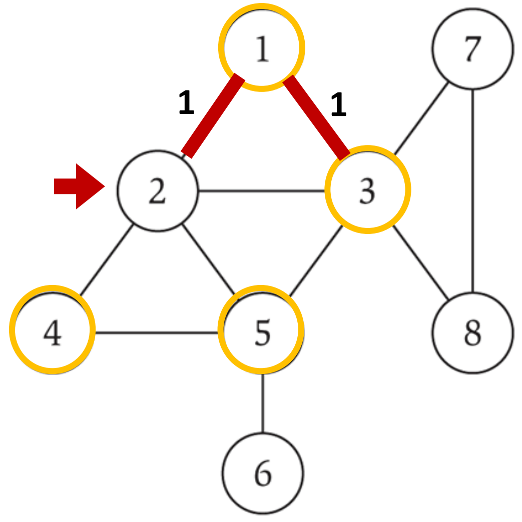
Vertex
3

Breadth First Search

U
2

V
1

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
➡           end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

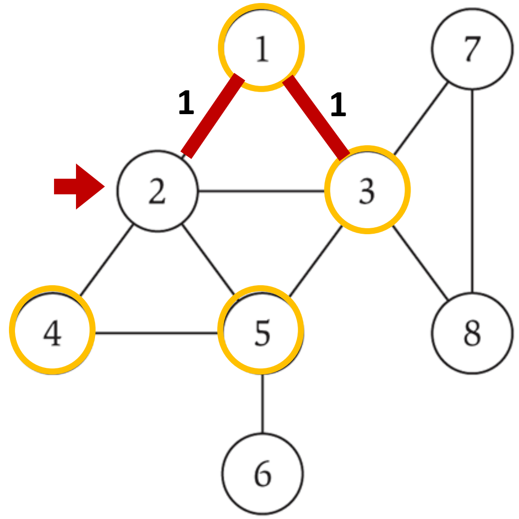
Vertex
3

Breadth First Search

U
2

V
3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

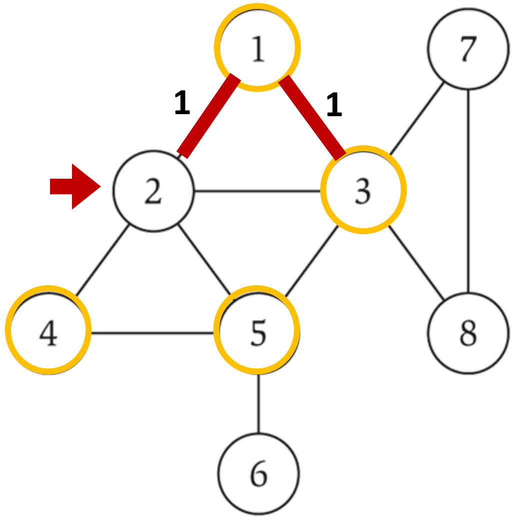
Queue

Vertex
3

Breadth First Search

U	V
2	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

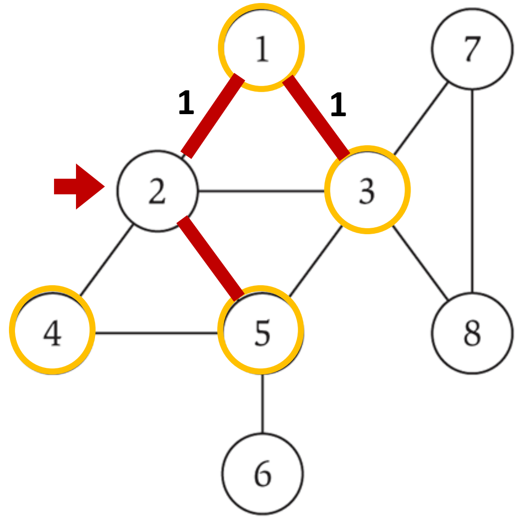
Queue

Vertex
3

Breadth First Search

U	V
2	5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	False
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

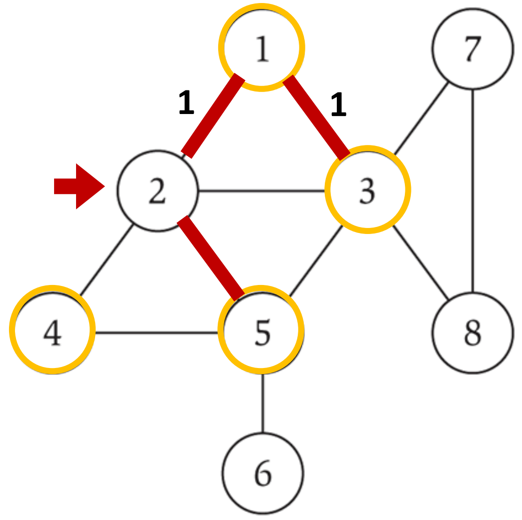
Queue

Vertex
3

Breadth First Search

U	V
2	5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	∞
6	∞
7	∞
8	∞

Queue

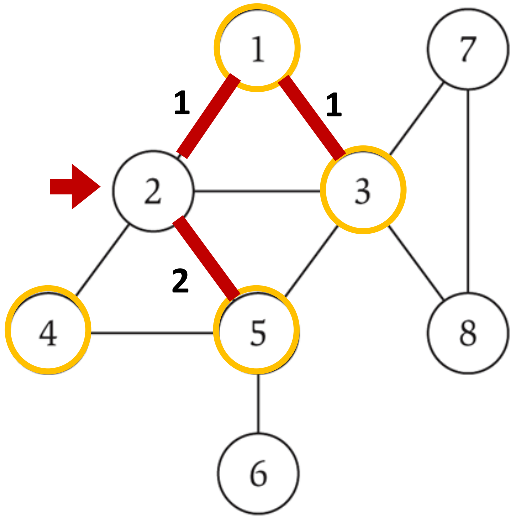
Vertex
3

Breadth First Search

U
2

V
5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	2
6	∞
7	∞
8	∞

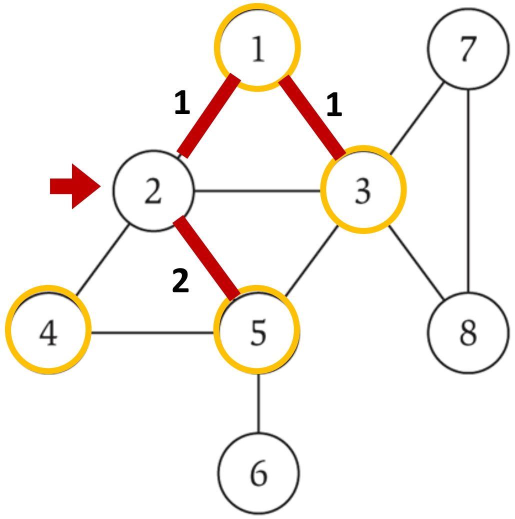
Queue

Vertex
3

Breadth First Search

U	V
2	5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	2
6	∞
7	∞
8	∞

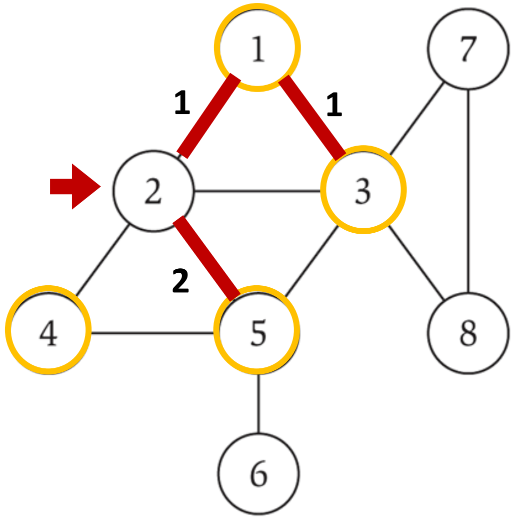
Queue

Vertex
3
5

Breadth First Search

U	V
2	5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
➡           end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	2
6	∞
7	∞
8	∞

Queue

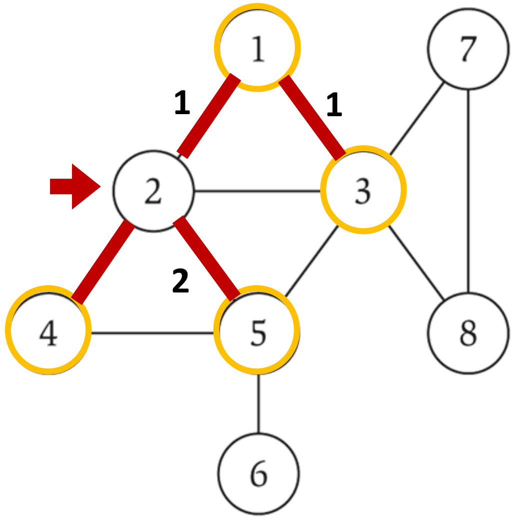
Vertex
3
5

Breadth First Search

U
2

V
4

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
➡           if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	False
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	2
6	∞
7	∞
8	∞

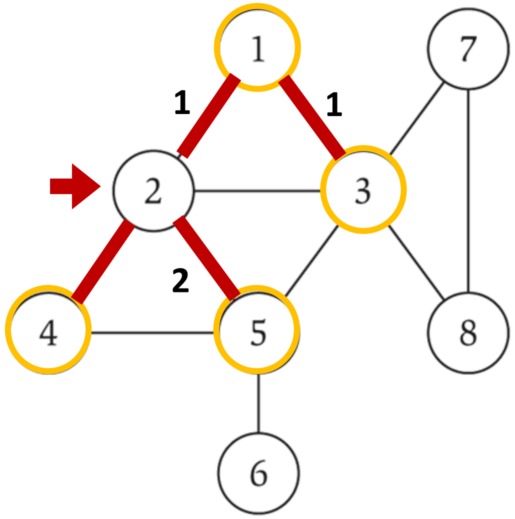
Queue

Vertex
3
5

Breadth First Search

U	V
2	4

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	∞
5	2
6	∞
7	∞
8	∞

Queue

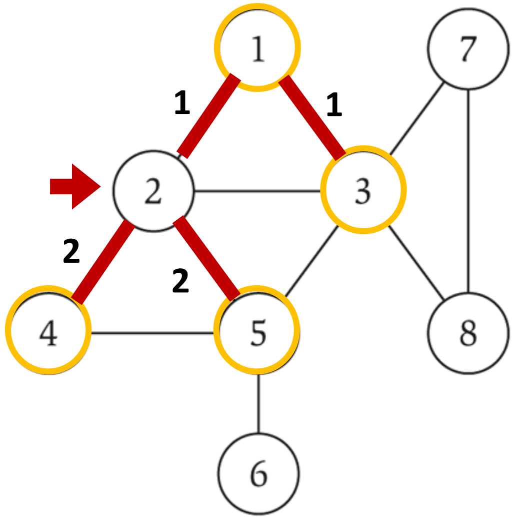
Vertex
3
5

Breadth First Search

U
2

V
4

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
➡               $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

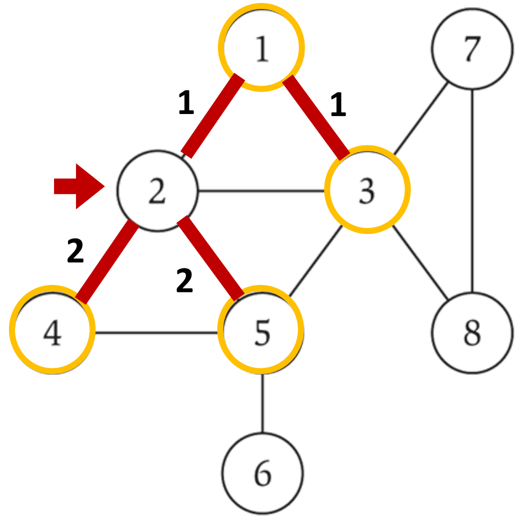
Queue

Vertex
3
5

Breadth First Search

U	V
2	4

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

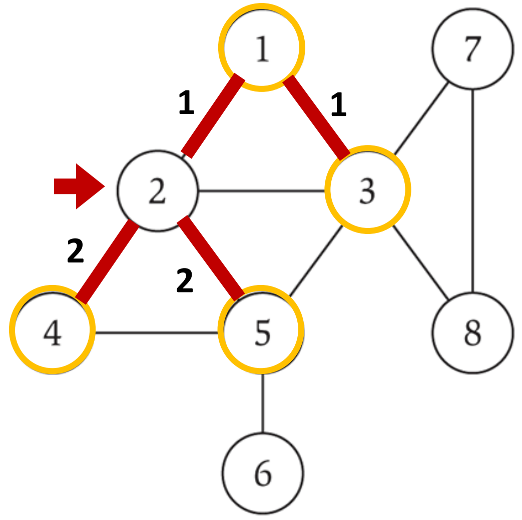
Queue

Vertex
3
5
4

Breadth First Search

U	V
2	4

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

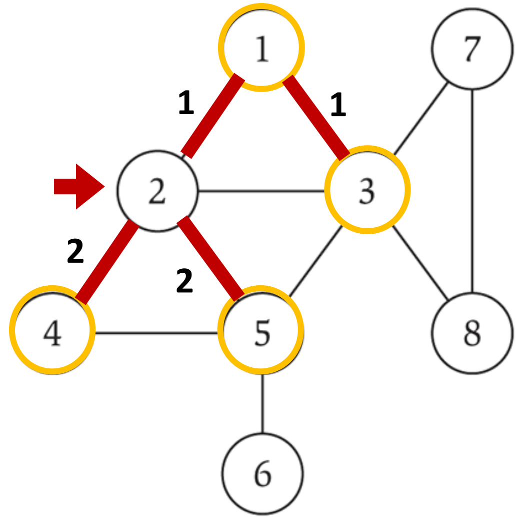
Queue

Vertex
3
5
4

Breadth First Search

U	V
2	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

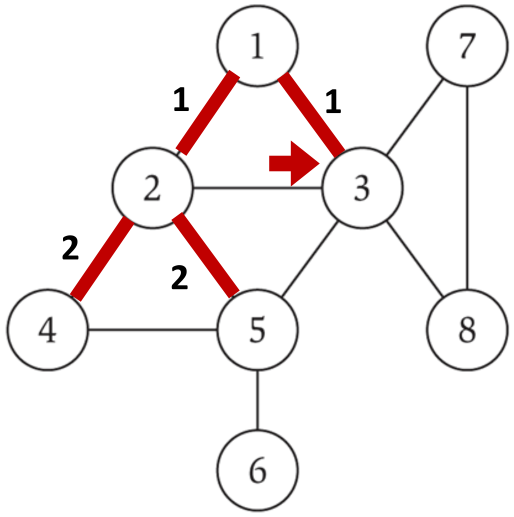
Queue

Vertex
3
5
4

Breadth First Search

U	V
3	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

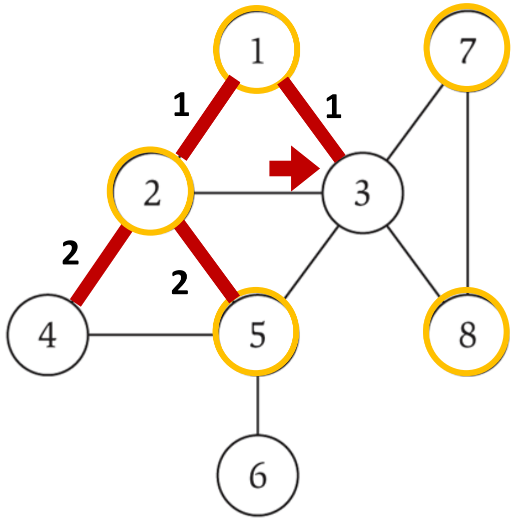
Queue

Vertex
5
4

Breadth First Search

U	V
3	

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        explored[v] ← false
        d[v] ← ∞
    end for
    explored[s] ← true
    d[s] ← 0
    Q:= a queue data structure, initialized with s
    while Q ≠ φ do
        u ← remove vertex from the front of Q
➡       for each v adjacent to u do
            if not explored[v] then
                explored[v] ← true
                d[v] ← d[u] + 1
                insert v to the end of Q
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

Queue

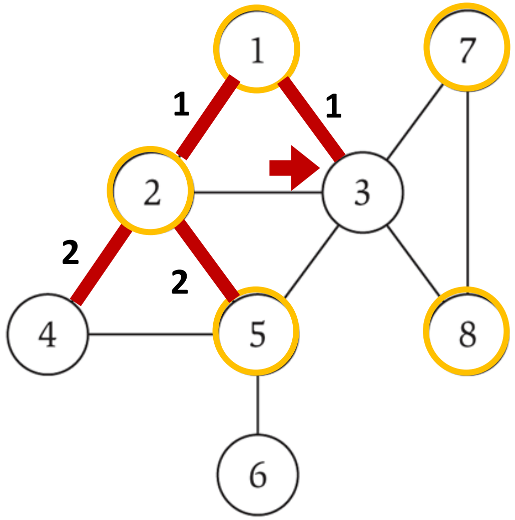
Vertex
5
4

Breadth First Search

U
3

V
1

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
➡           if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

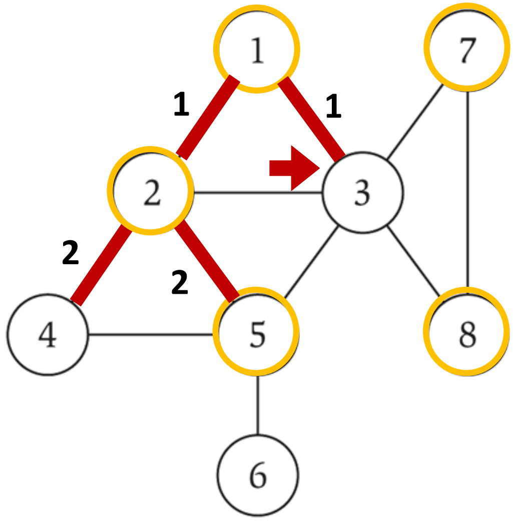
Queue

Vertex
5
4

Breadth First Search

U	V
3	1

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

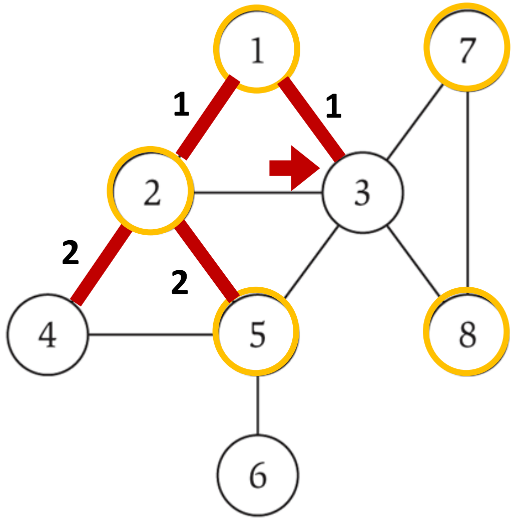
Queue

Vertex
5
4

Breadth First Search

U	V
3	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

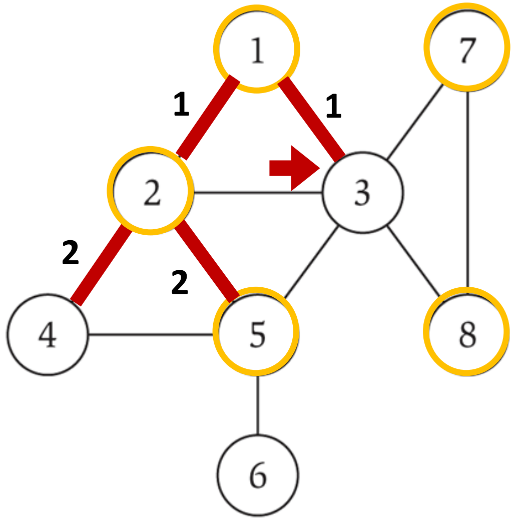
Queue

Vertex
5
4

Breadth First Search

U	V
3	2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

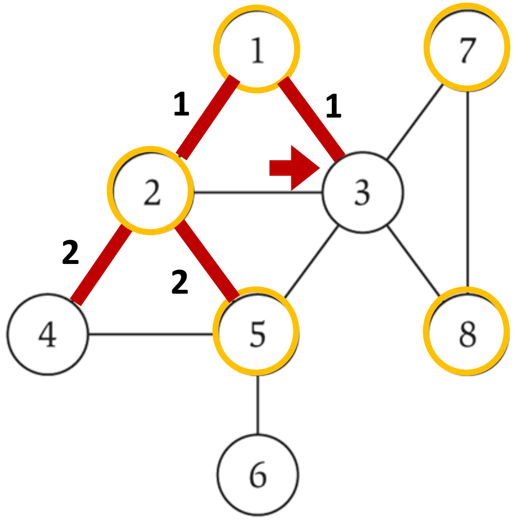
Queue

Vertex
5
4

Breadth First Search

U	V
3	5

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

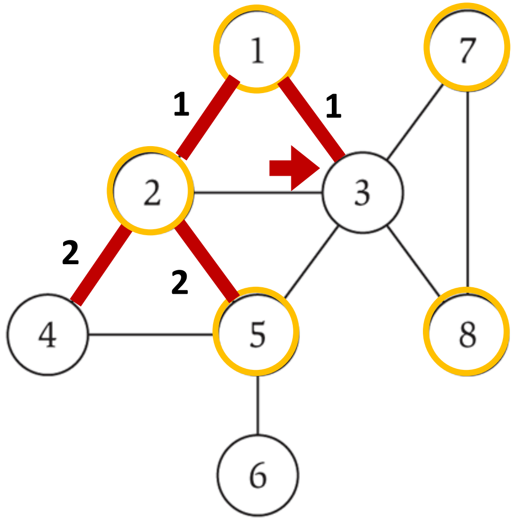
Queue

Vertex
5
4

Breadth First Search

U	V
3	5

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

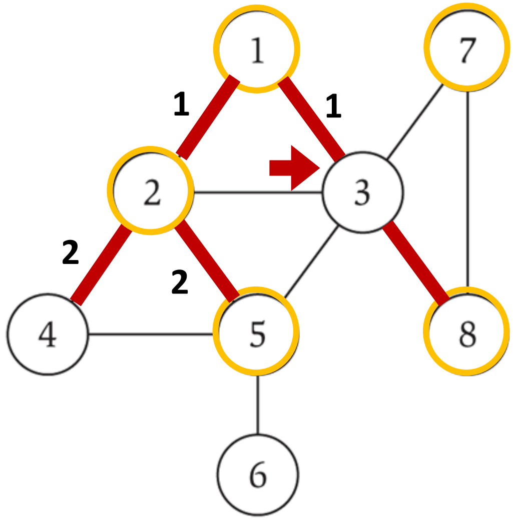
Queue

Vertex
5
4

Breadth First Search

U	V
3	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	False

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

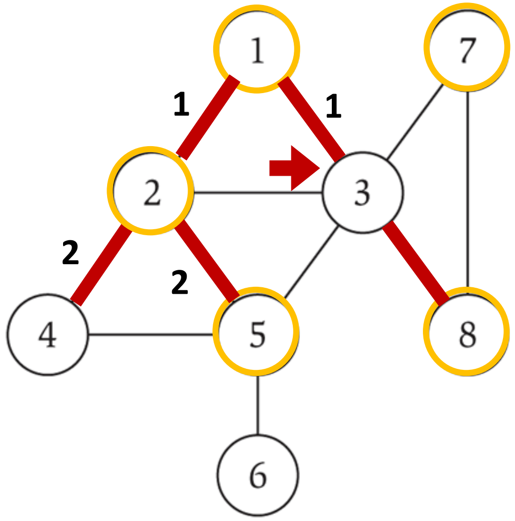
Queue

Vertex
5
4

Breadth First Search

U	V
3	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	∞

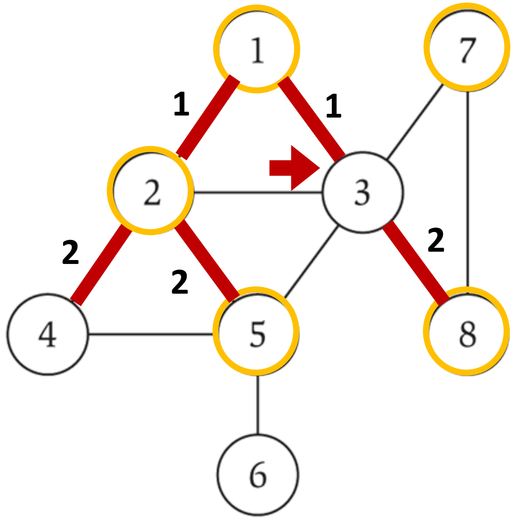
Queue

Vertex
5
4

Breadth First Search

U	V
3	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	2

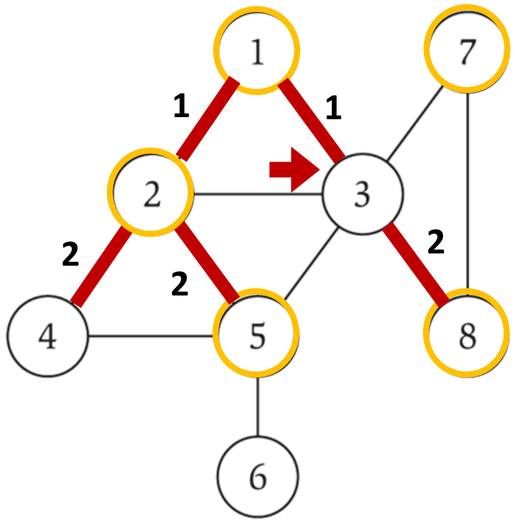
Queue

Vertex
5
4

Breadth First Search

U	V
3	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	2

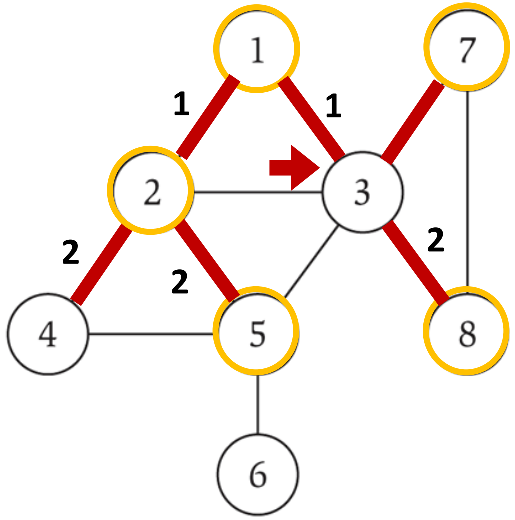
Queue

Vertex
5
4
8

Breadth First Search

U	V
3	7

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
➡           if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	False
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	2

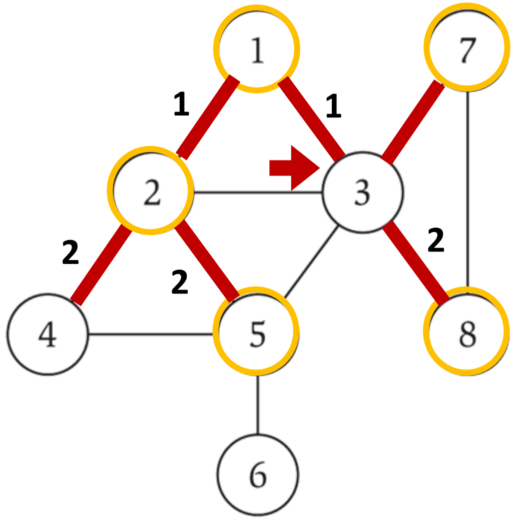
Queue

Vertex
5
4
8

Breadth First Search

U	V
3	7

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	∞
8	2

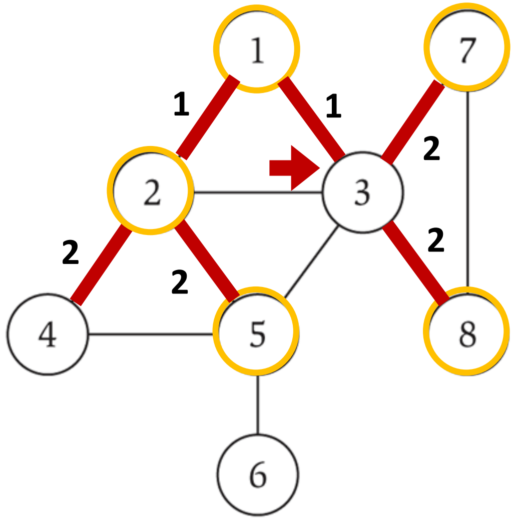
Queue

Vertex
5
4
8

Breadth First Search

U	V
3	7

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

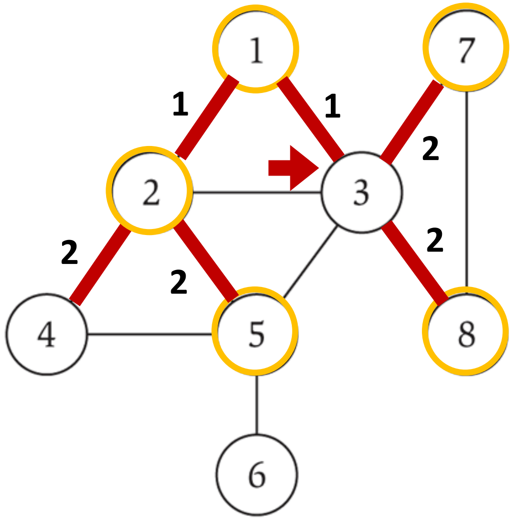
Queue

Vertex
5
4
8

Breadth First Search

U	V
3	7

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

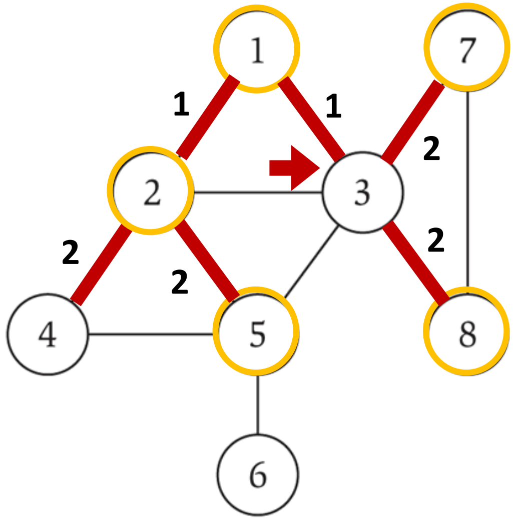
Queue

Vertex
5
4
8
7

Breadth First Search

U	V
3	7

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
➡           end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

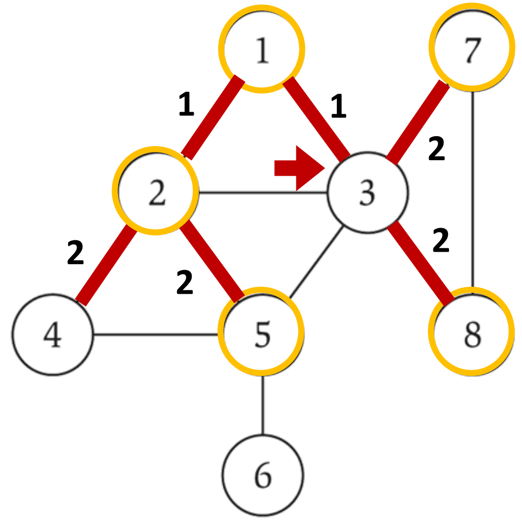
Queue

Vertex
5
4
8
7

Breadth First Search

U	V
3	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

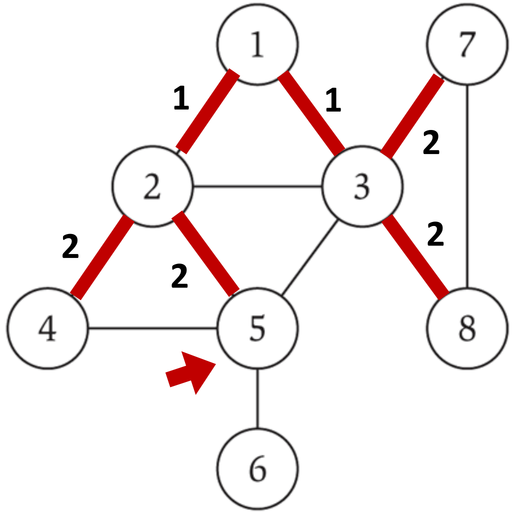
Queue

Vertex
5
4
8
7

Breadth First Search

U	V
5	

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

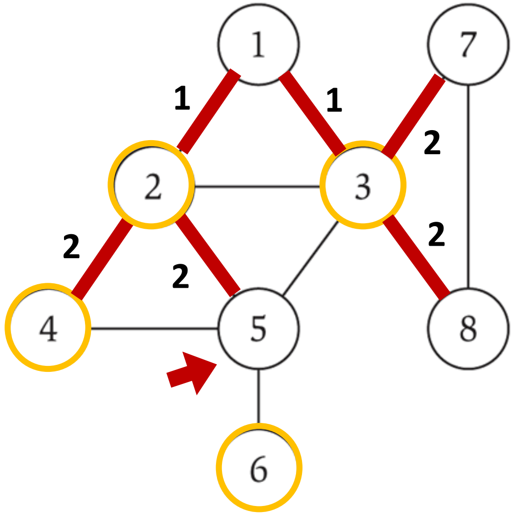
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        explored[v] ← false
        d[v] ← ∞
    end for
    explored[s] ← true
    d[s] ← 0
    Q:= a queue data structure, initialized with s
    while Q ≠ ϕ do
        u ← remove vertex from the front of Q
        for each v adjacent to u do
            if not explored[v] then
                explored[v] ← true
                d[v] ← d[u] + 1
                insert v to the end of Q
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

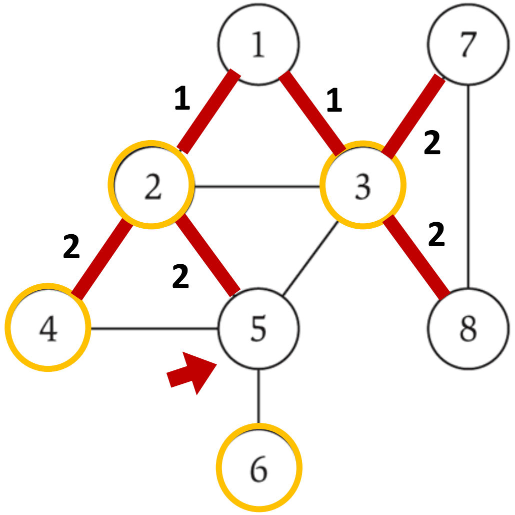
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	4

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        explored[v] ← false
        d[v] ← ∞
    end for
    explored[s] ← true
    d[s] ← 0
    Q:= a queue data structure, initialized with s
    while Q ≠ ϕ do
        u ← remove vertex from the front of Q
        for each v adjacent to u do
➡           if not explored[v] then
                explored[v] ← true
                d[v] ← d[u] + 1
                insert v to the end of Q
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

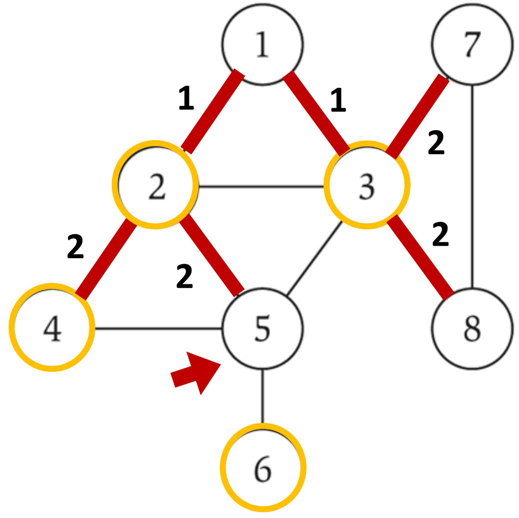
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	4

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

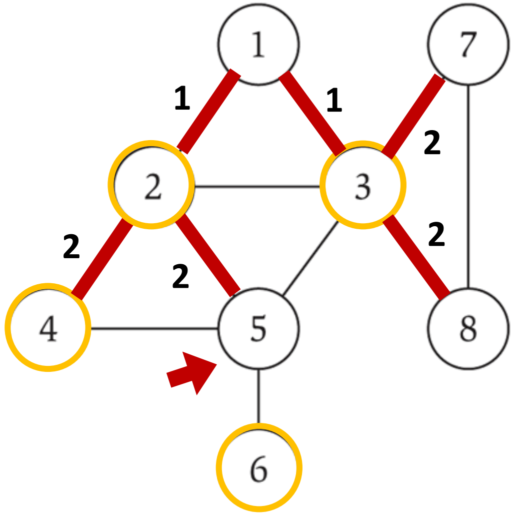
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	2

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

Queue

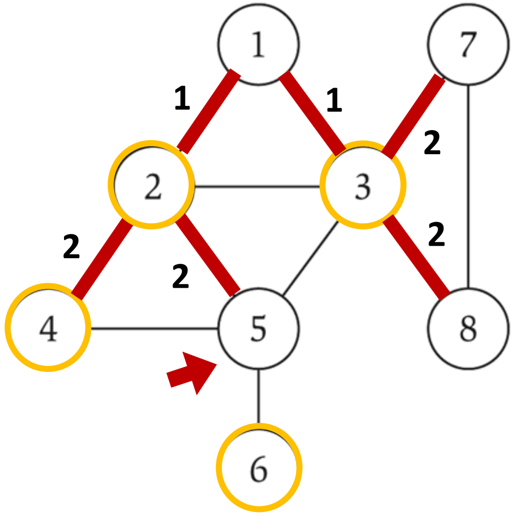
Vertex
4
8
7

Breadth First Search

U
5

V
2

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

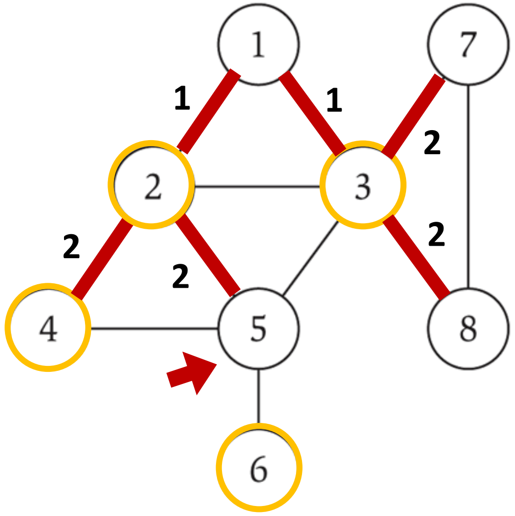
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

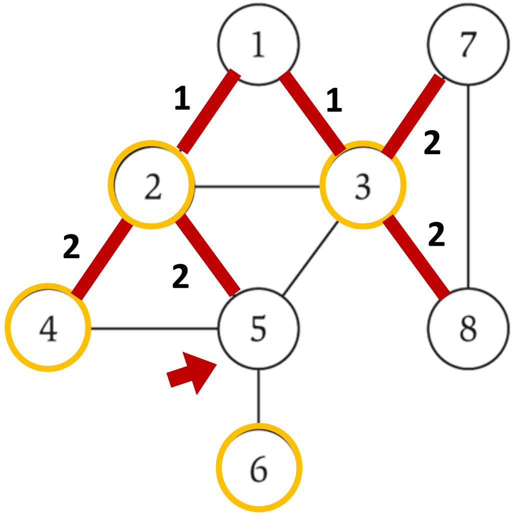
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	3

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

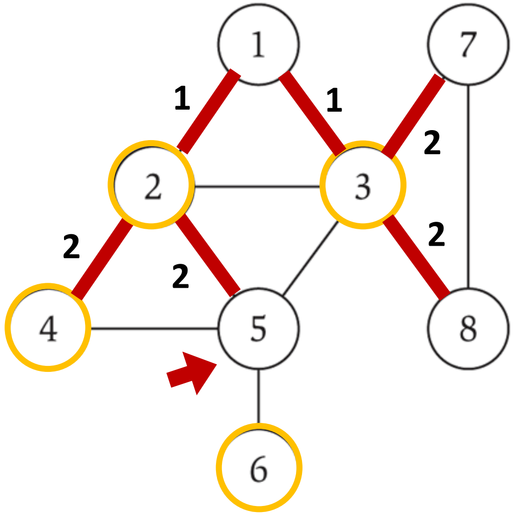
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

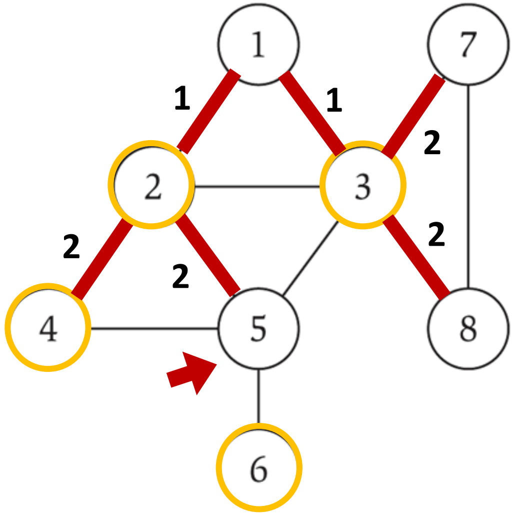
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	8

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

Queue

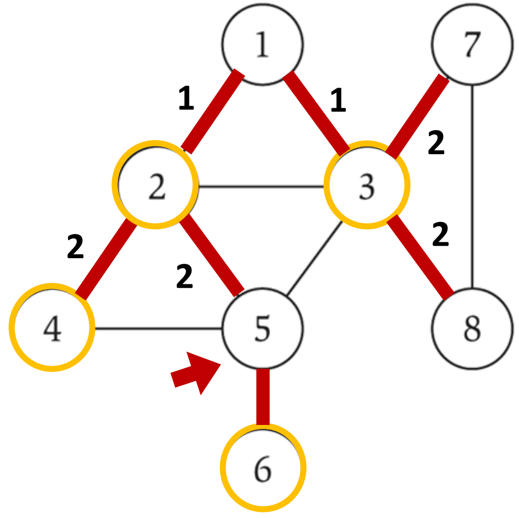
Vertex
4
8
7

Breadth First Search

U
5

V
6

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	False
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

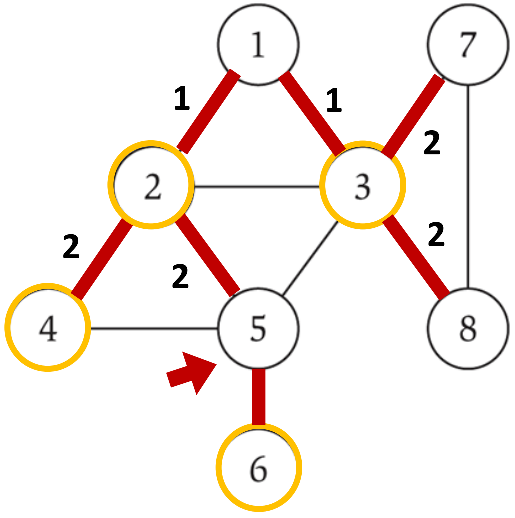
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	6

```
procedure BFS(G,s)

    for each vertex v ∈ V[G] do
        $explored$[v] ← false
        $d$[v] ← ∞
    end for
    $explored$[s] ← true
    $d$[s] ← 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u ← remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
➡               $explored$[v] ← true
                $d$[v] ← $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	∞
7	2
8	2

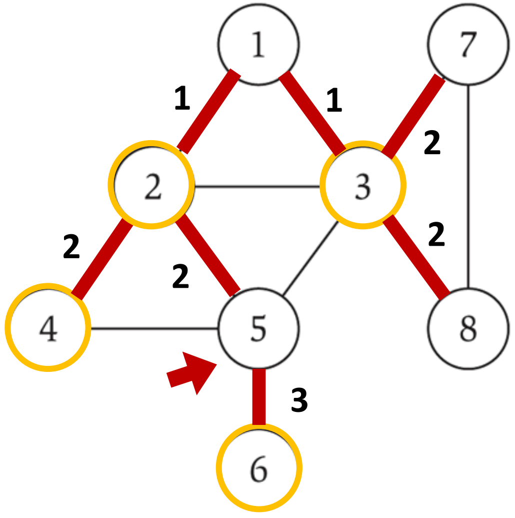
Queue

Vertex
4
8
7

Breadth First Search

U	V
5	6

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

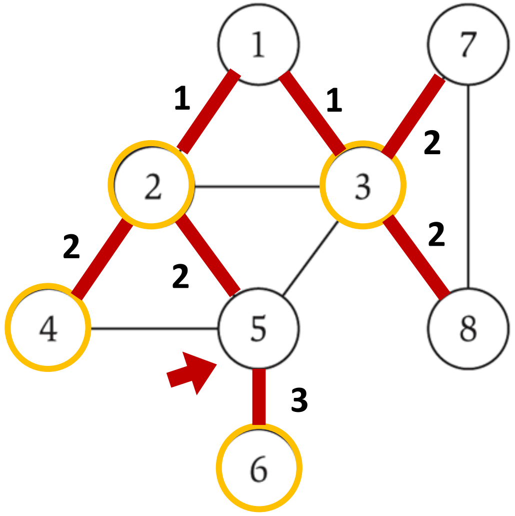
Vertex
4
8
7

Breadth First Search

U
5

V
6

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

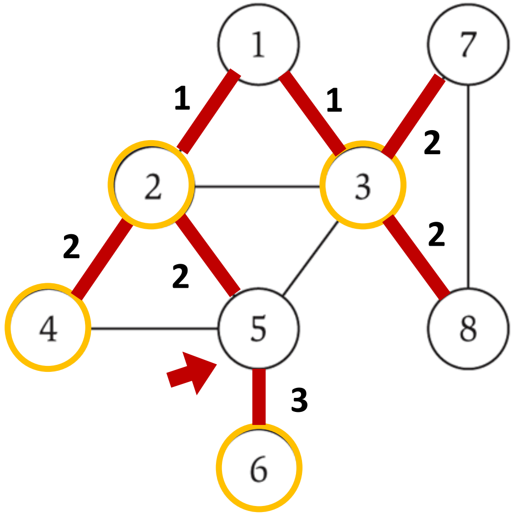
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex
4
8
7
6

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

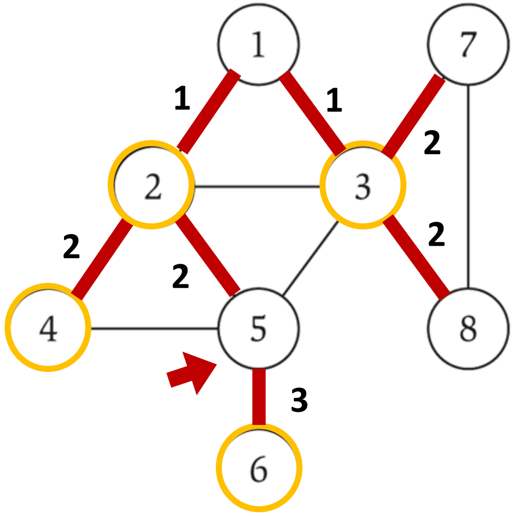
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex
4
8
7

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

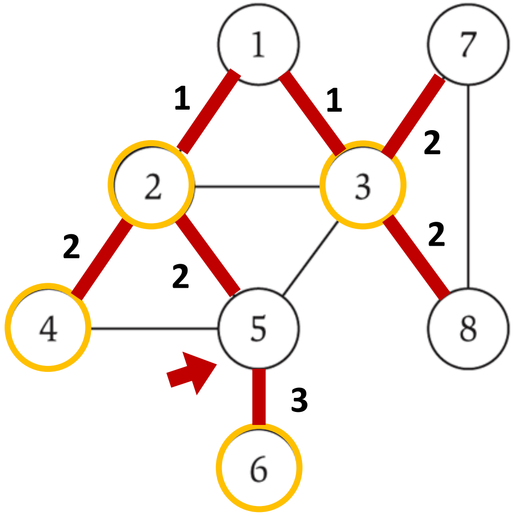
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex
4
8

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

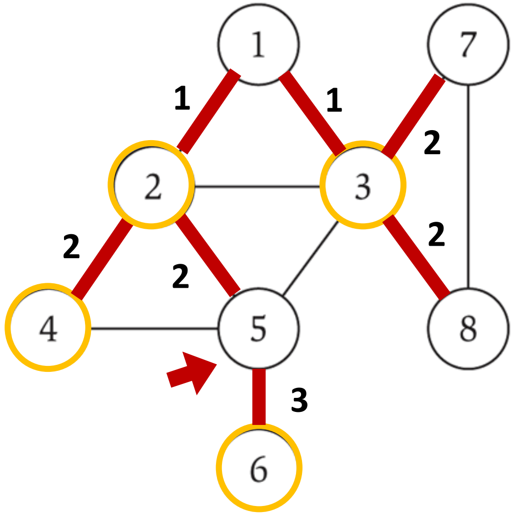
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex
4

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

Distance

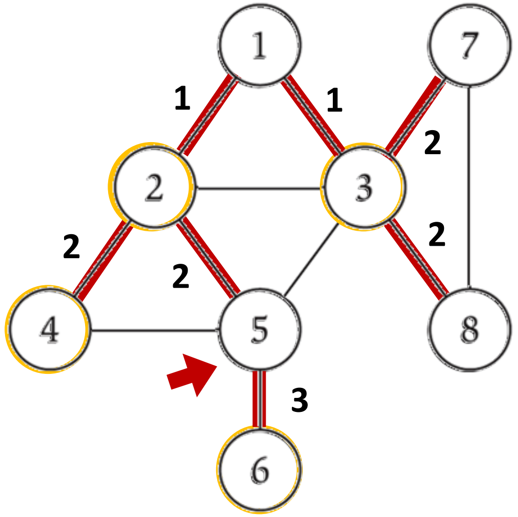
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex

Breadth First Search

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false
        $d$[v] $\leftarrow$ $\infty$
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Explored

Vertex	Value
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True

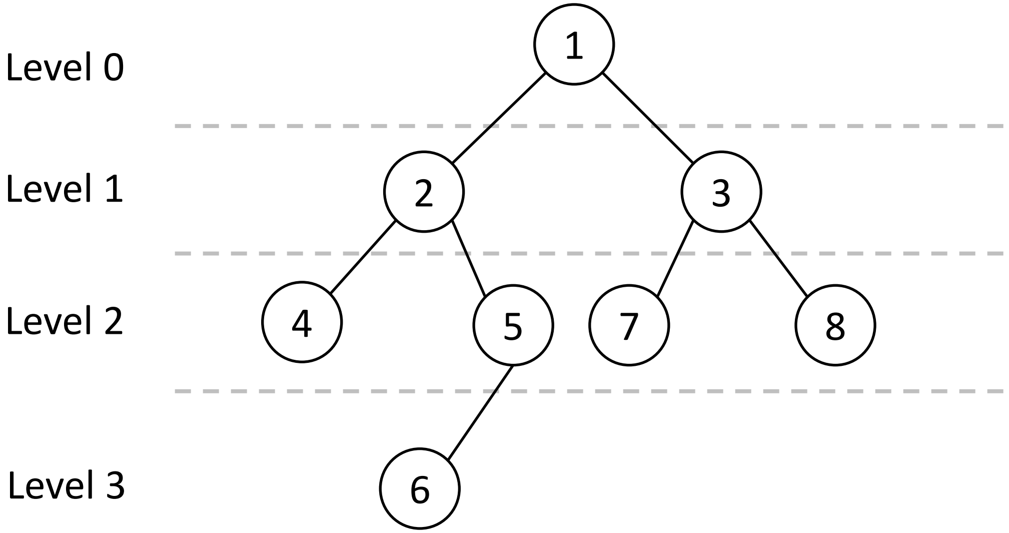
Distance

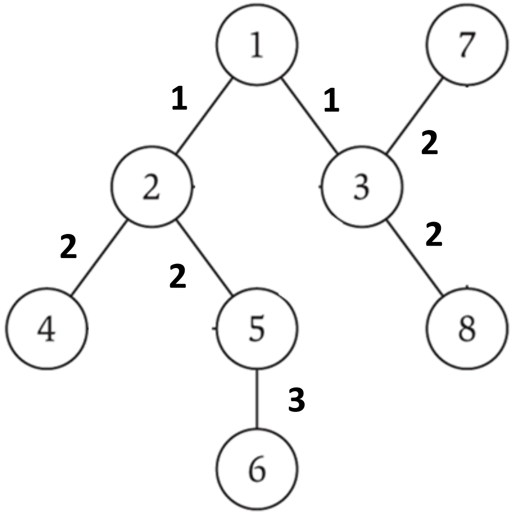
Vertex	Value
1	0
2	1
3	1
4	2
5	2
6	3
7	2
8	2

Queue

Vertex

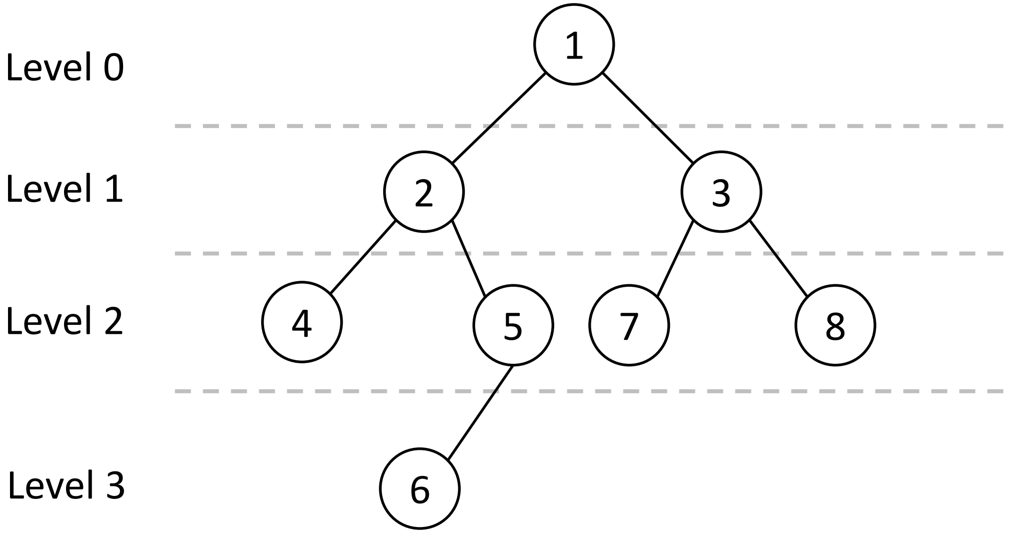
Breadth First Search

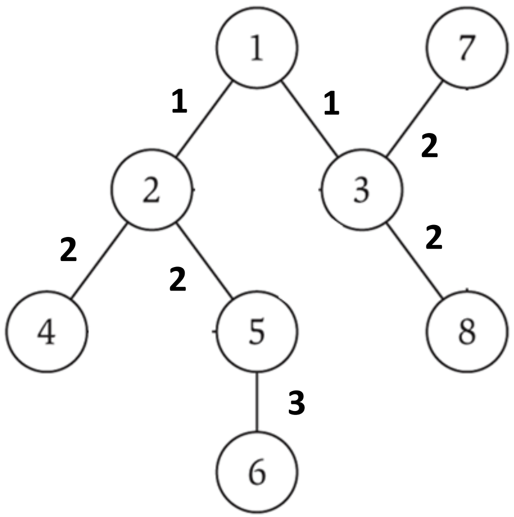




Breadth First Search

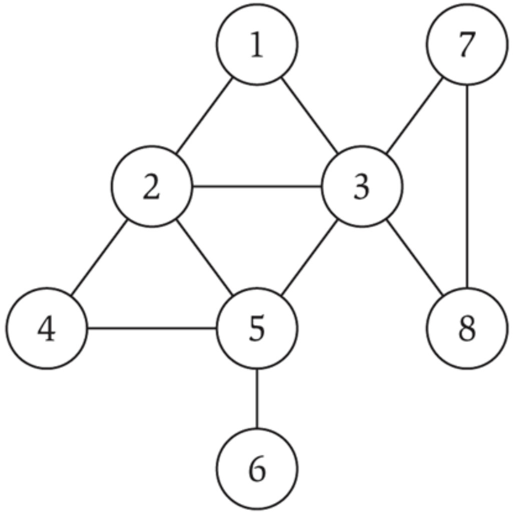
Property: Let ***T*** be a BFS tree of **G = (V, E)**, and let **(x, y)** be an edge of **G**. Then the level of ***x*** and ***y*** differ by at most **1**.





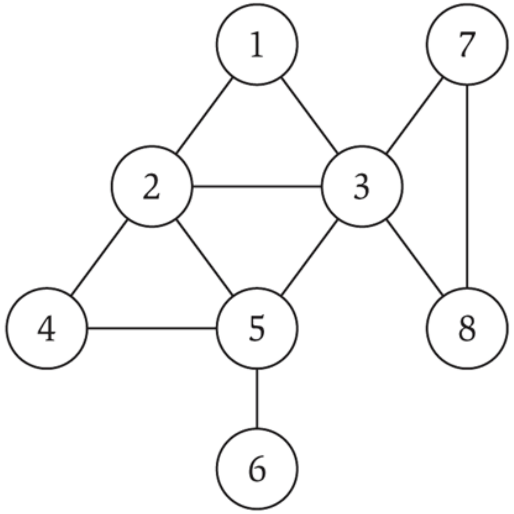
Breadth First Search - Analysis

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false    O(1)    O(V)
        $d$[v] $\leftarrow$ $\infty$                O(1)
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true              O(V)
                $d$[v] $\leftarrow$ $d$[u] + 1         O(deg(v))
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Breadth First Search - Analysis

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false   O(1)
        $d$[v] $\leftarrow$ $\infty$            O(1)     O(V)
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true                  O(2E)
                $d$[v] $\leftarrow$ $d$[u] + 1
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Time Complexity: **O(V) + O(2E)**

O(V + 2E)

O(V + E)

BFS: Live Poll 1

Consider a complete undirected graph where every vertex ***V*** has an edge with every other vertex. You are going to perform a Breadth-First Search (BFS) on this graph.

Which of the following expressions give equivalent time complexity in terms of the Big O notation of the BFS for this graph?

A. V^2

B. 2E

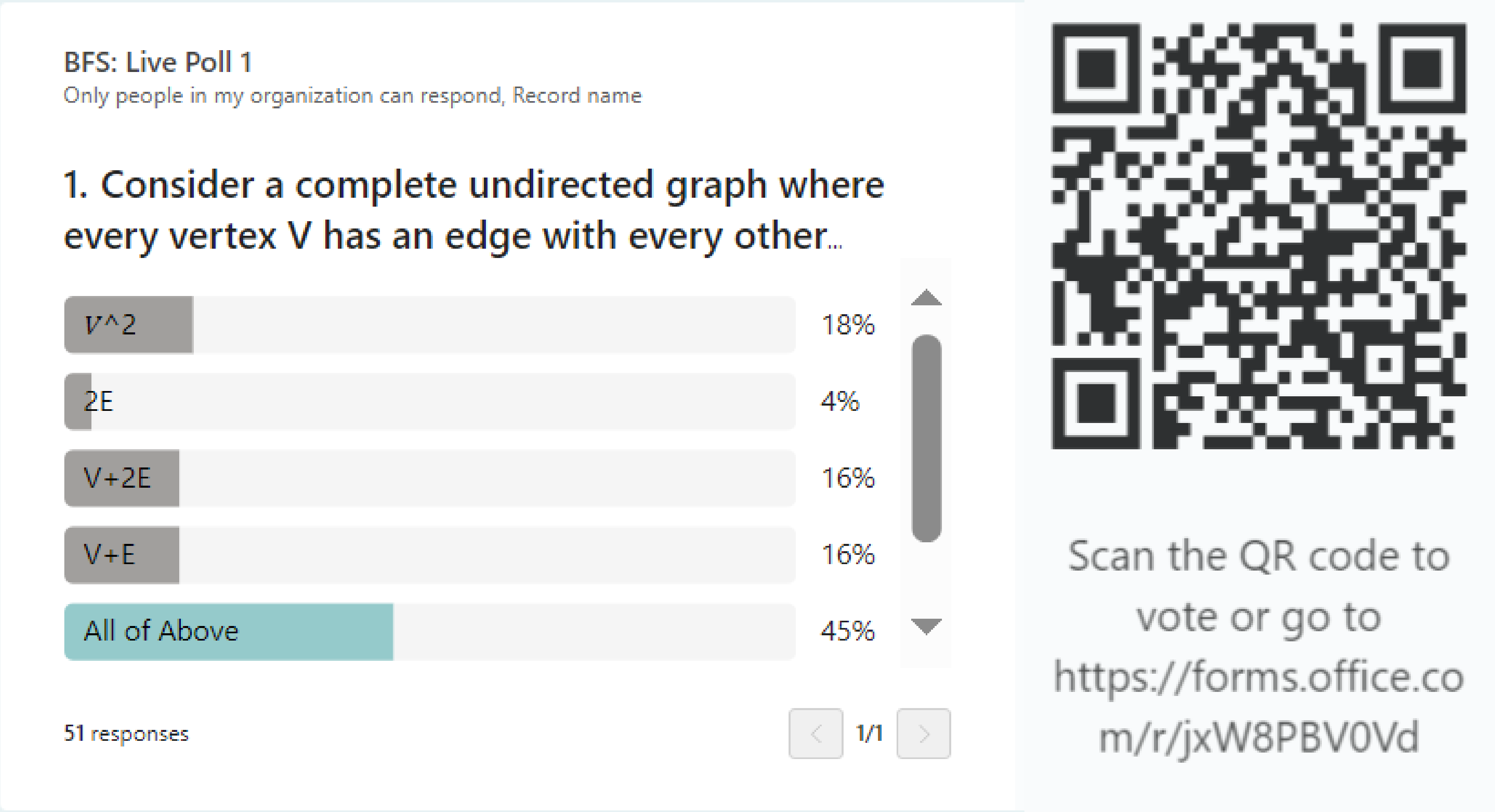
C. V+2E

D. V+E

E. All of Above

F. None of Above

Scan the QR code to vote or go to https://forms.office.com/r/jxW8PBV0Vd



BFS: Live Poll 1

Only people in my organization can respond, Record name

1. Consider a complete undirected graph where every vertex V has an edge with every other...

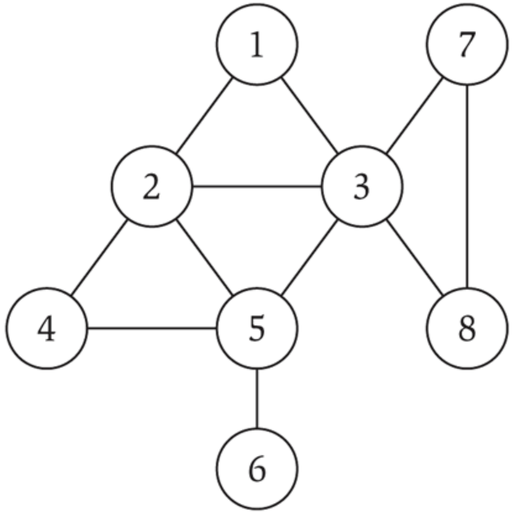
V^2	18%
2E	4%
V+2E	16%
V+E	16%
All of Above	45%

51 responses

Scan the QR code to vote or go to https://forms.office.com/r/jxW8PBV0Vd

Breadth First Search - Analysis

```
procedure BFS(G,s)

    for each vertex v $\in$ V[G] do
        $explored$[v] $\leftarrow$ false   O(1)   O(V)
        $d$[v] $\leftarrow$ $\infty$              O(1)
    end for
    $explored$[s] $\leftarrow$ true
    $d$[s] $\leftarrow$ 0
    $Q$:= a queue data structure, initialized with s
    while $Q \neq \phi$ do
        u $\leftarrow$ remove vertex from the front of $Q$
        for each v adjacent to u do
            if not $explored$[v] then
                $explored$[v] $\leftarrow$ true                      O(V)
                $d$[v] $\leftarrow$ $d$[u] + 1     (V-1) = O(V)
                insert v to the end of $Q$
            end if
        end for
    end while

end procedure
```



Time Complexity: O(V) + O(V^2)

O(V^2)

BFS: Live Poll 1

Consider a complete undirected graph where every vertex ***V*** has an edge with every other vertex. You are going to perform a Breadth-First Search (BFS) on this graph.

Which of the following expressions give equivalent time complexity in terms of the Big O notation of the BFS for this graph?

$A.$ $V^2 = \mathbf{O}(\boldsymbol{V^2})$ ✓

B. 2E = $V(V-1) = V^2 - V = \mathbf{O}(\boldsymbol{V^2})$ ✓ 😈

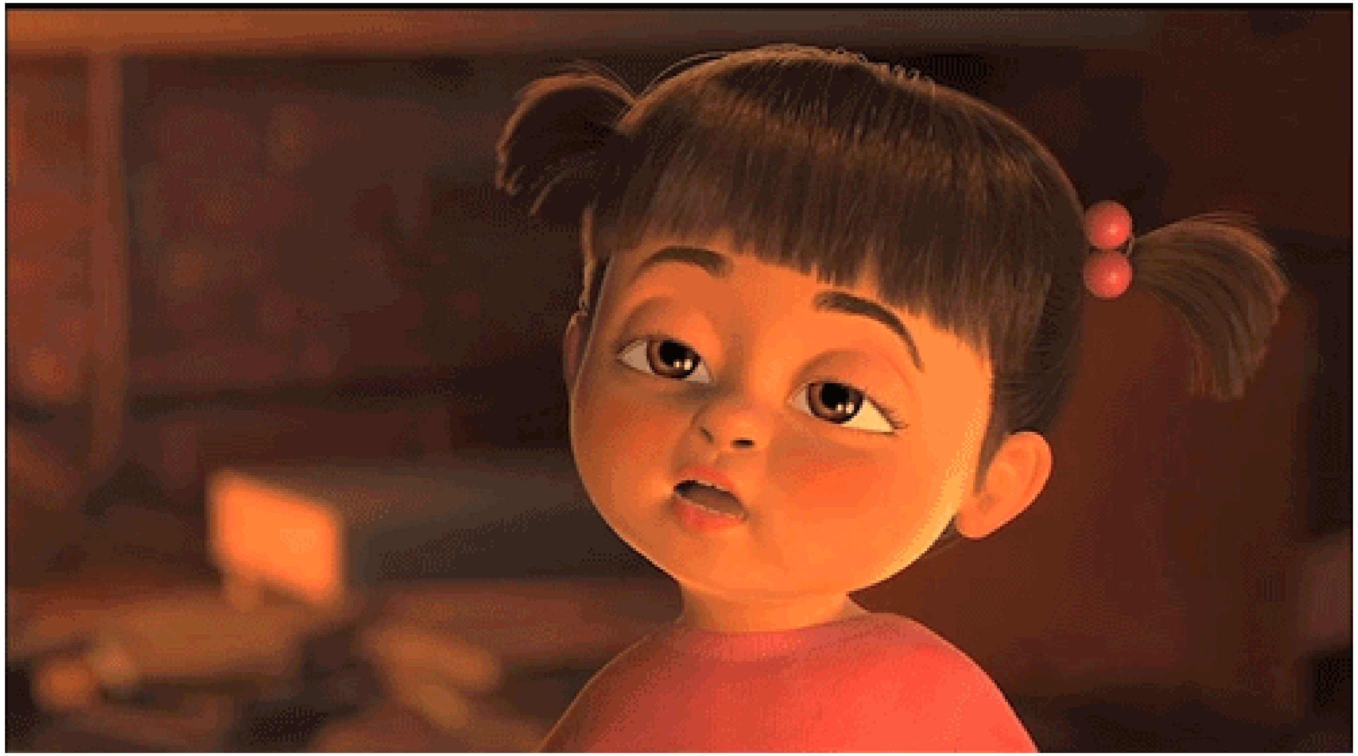
C. V+2E $= V + V(V-1) = V^2 = \mathbf{O}(\boldsymbol{V^2})$ ✓ 😈

D. V+E $= V + (V(V-1))/2 = V + (V^2 - V)/2 = \mathbf{O}(\boldsymbol{V^2})$ ✓ 😈

E. All of Above

F. None of Above

Thanks a lot



If you are taking a Nap, **wake up**........Lecture Over