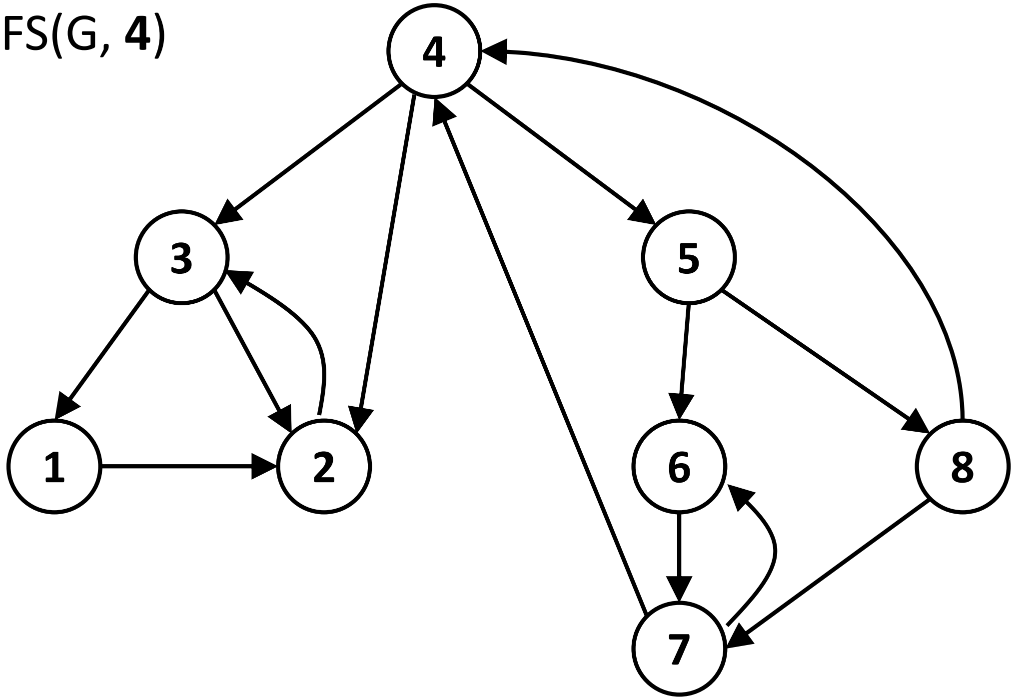
Graphs Algorithms I

Breadth First Search

```
BFS(G, s)                                                    /*  G = (V, E) and s ∈ V  */
    color[s] ← grey ;  d[s] ← 0 ;  p[s] ← NIL
    For each v ∈ V – {s}   do
            color[v] ← white
            d[v] ← ∞
            p[v] ← NIL
    Q ← empty ; ENQ(Q, s)                      /*  Q: nodes that are discovered but not yet explored  */
    While Q is not empty   do
            u ← DEQ(Q)                                       /*  Explore u  */
            For each (u, v) ∈ E   do                         /*  Explore edge (u,v)  */
                    If color[v] = white   then do            /*  If v is first discovered  */
                            color[v] ← grey
                            d[v] ← d[u] + 1
                            p[v] ← u
                            ENQ(Q, v)
                    End If
            End For
            color[u] ← black                                 /*  Done exploring u  */
    End While
End BFS
```

BFS(G, **4**)

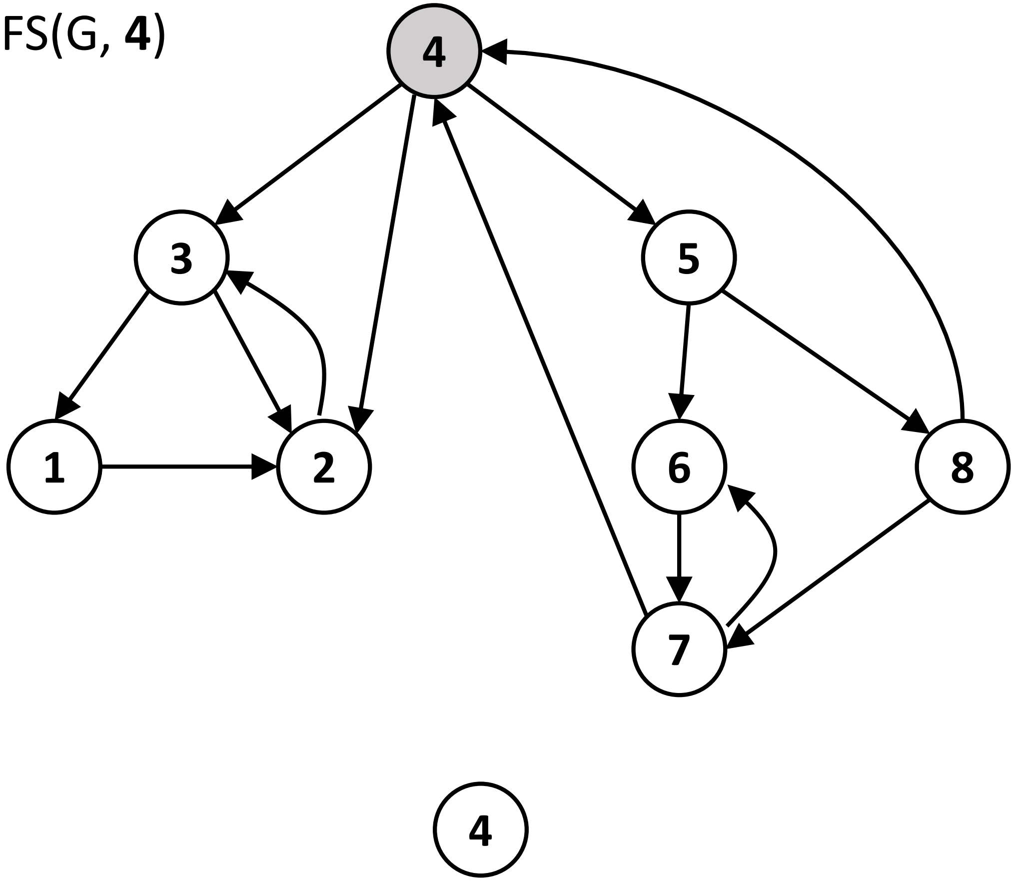


Adj List of G :

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

Contents of Q :

BFS(G, **4**)



Adj List of G :

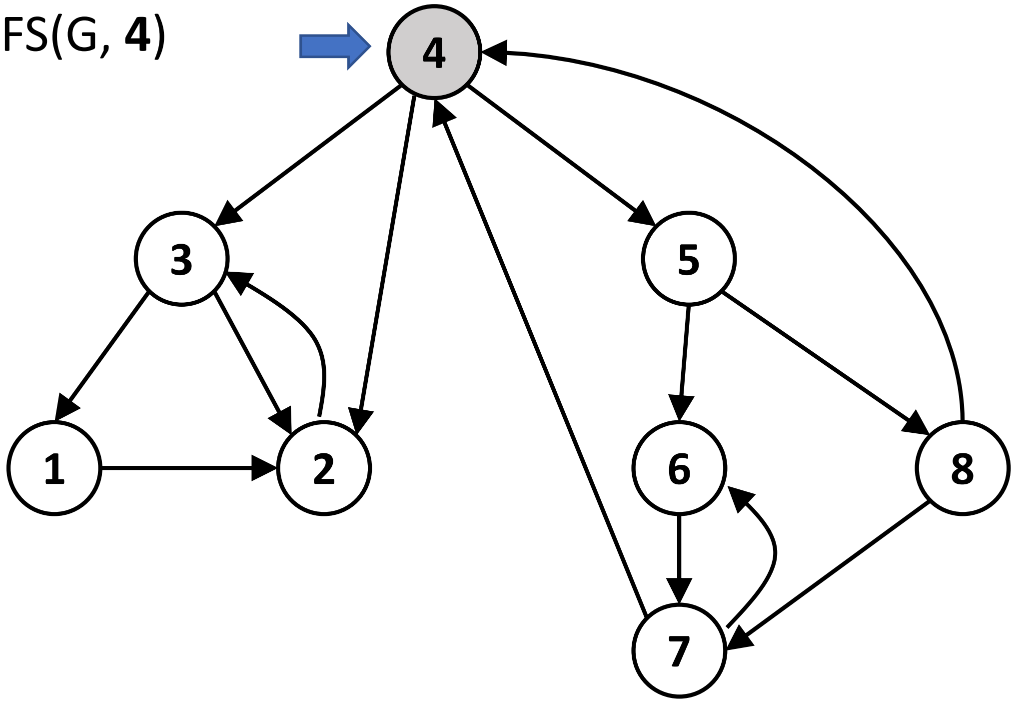
1	→ **2**
2	→ **3**
3	→ **1** → **2**
4	→ **3** → **2** → **5**
5	→ **6** → **8**
6	→ **7**
7	→ **6** → **4**
8	→ **7** → **4**

Contents of Q :

d = 0

$\boxed{\mathbf{4}}$

BFS(G, **4**)



Adj List of G :

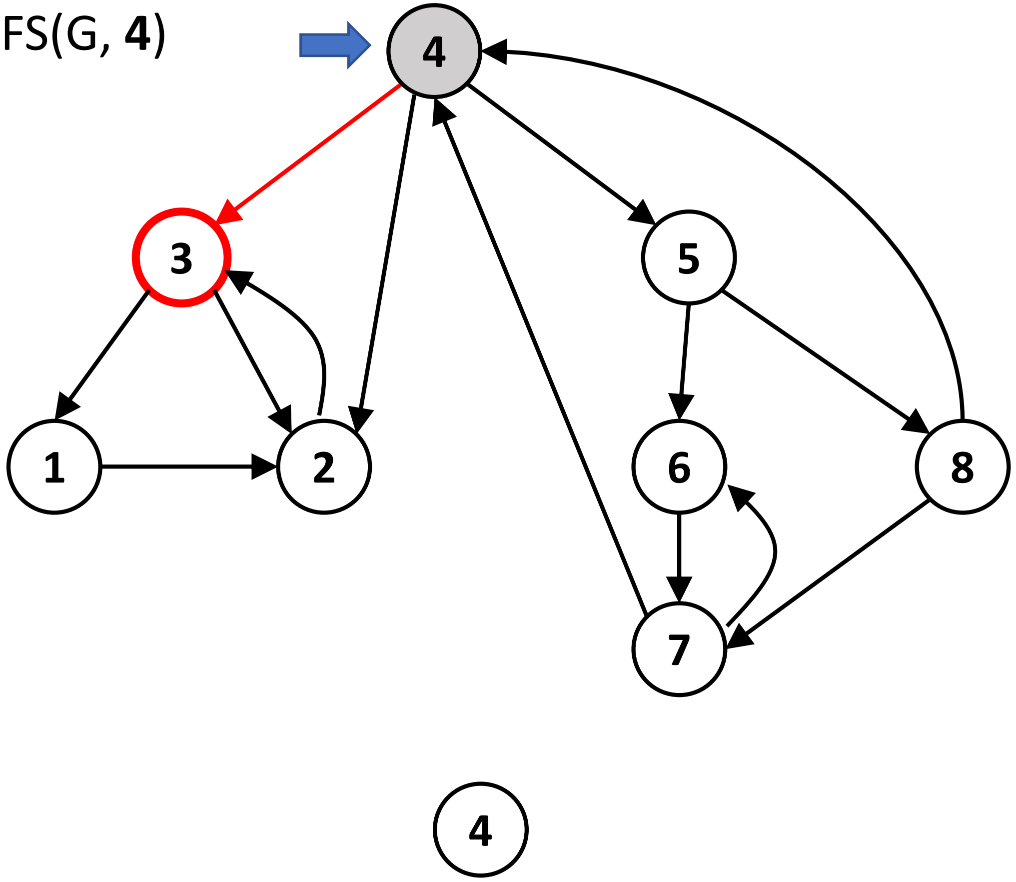
1 → **2**
2 → **3**
3 → **1** → **2**
4 → **3** → **2** → **5**
5 → **6** → **8**
6 → **7**
7 → **6** → **4**
8 → **7** → **4**

Contents of Q :

d = 0

$\boxed{\mathbf{4}}$

BFS(G, **4**)



Adj List of G :

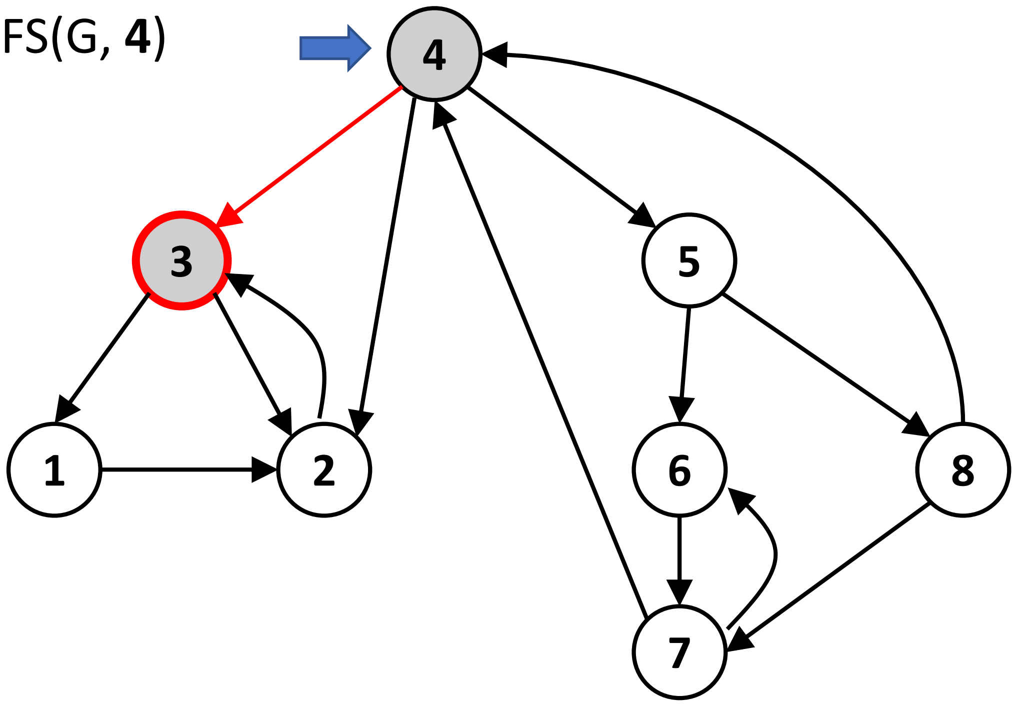
1 → 2
2 → 3
3 → 1 → 2
4 → 3 → 2 → 5
5 → 6 → 8
6 → 7
7 → 6 → 4
8 → 7 → 4

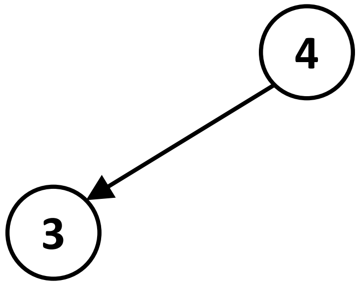
Contents of Q :

d = 0

$\boxed{4}$

BFS(G, **4**)





Adj List of G :

1	**2**
2	**3**
3	**1** → **2**
4	**3** → **2** → **5**
5	**6** → **8**
6	**7**
7	**6** → **4**
8	**7** → **4**

Contents of Q :

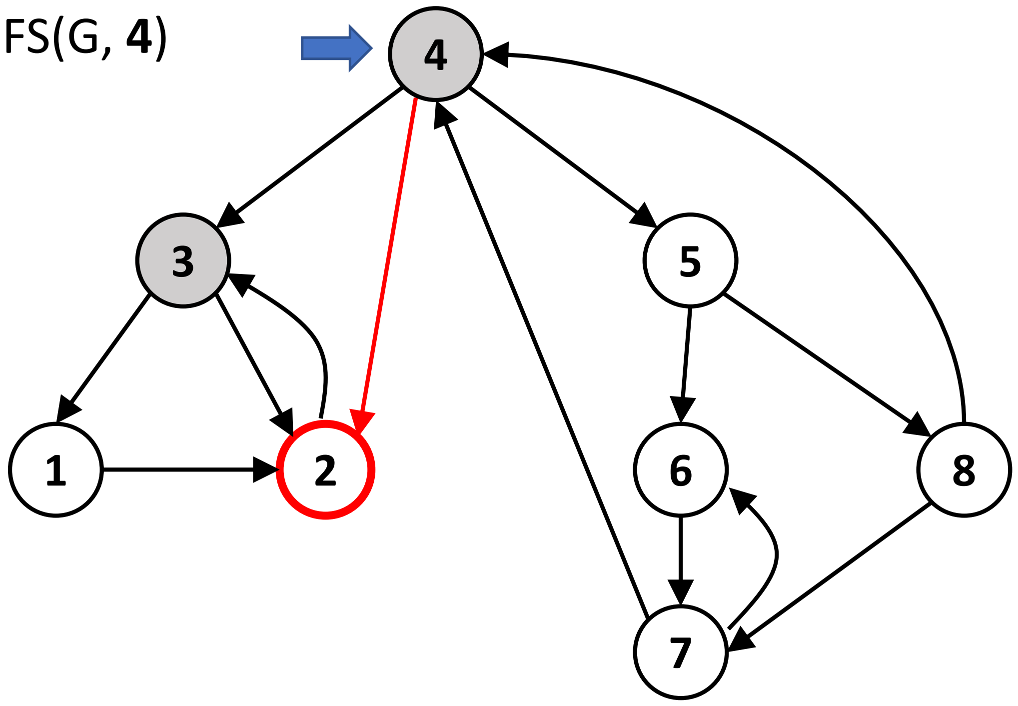
d = 0

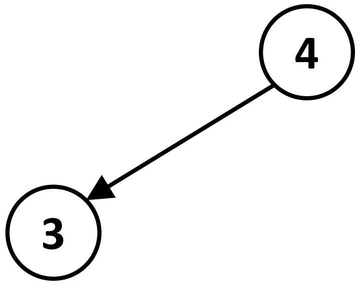
$\boxed{4}$

d = 1

$\boxed{3}$

BFS(G, **4**)





Adj List of G :

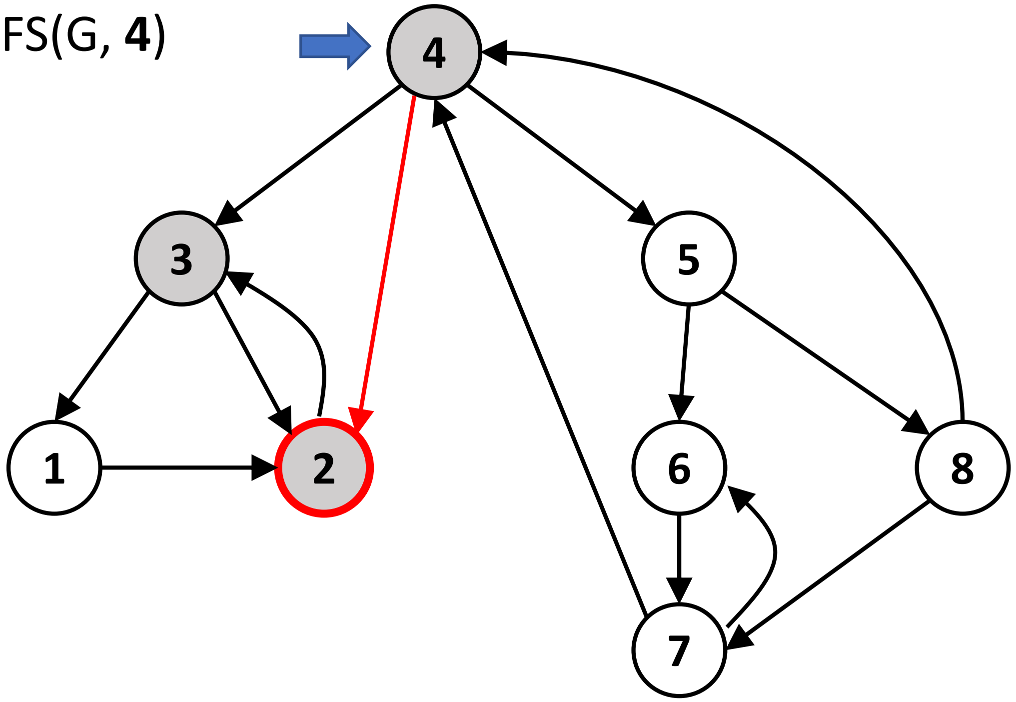
1 → **2**
2 → **3**
3 → **1** → **2**
4 → **3** → **2** → **5**
5 → **6** → **8**
6 → **7**
7 → **6** → **4**
8 → **7** → **4**

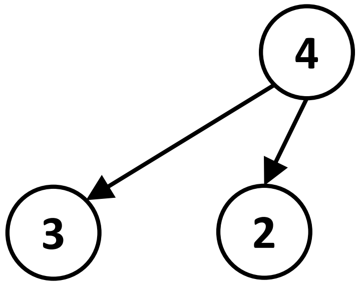
Contents of Q :

d = 0
$\boxed{4}$

d = 1
$\boxed{3}$

BFS(G, **4**)





Adj List of G :

1 → 2

2 → 3

3 → 1 → 2

4 → 3 → 2 → 5

5 → 6 → 8

6 → 7

7 → 6 → 4

8 → 7 → 4

Contents of Q :

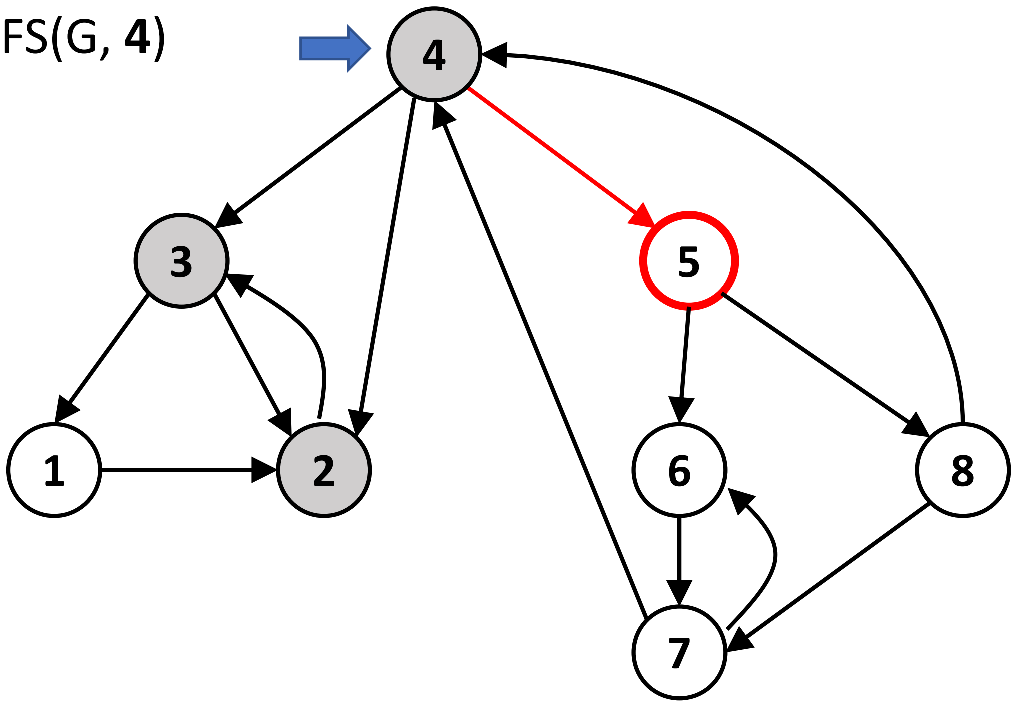
d = 0

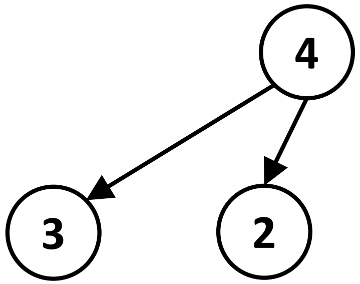
4

d = 1 1

3	**2**

BFS(G, **4**)





Adj List of G :

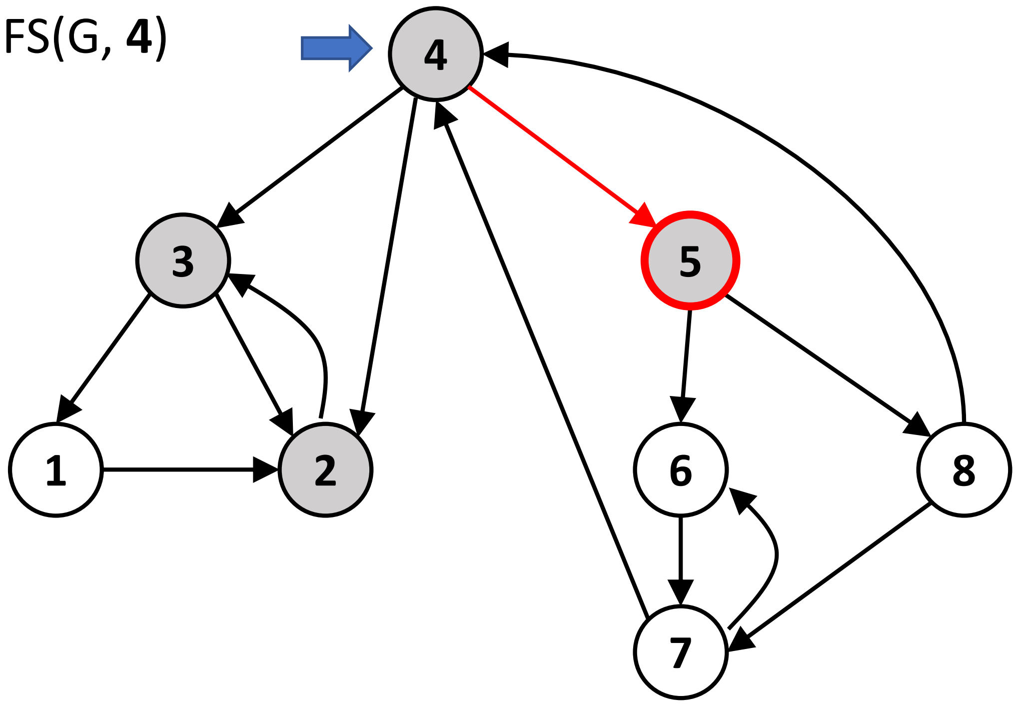
1 → **2**
2 → **3**
3 → **1** → **2**
4 → **3** → **2** → **5**
5 → **6** → **8**
6 → **7**
7 → **6** → **4**
8 → **7** → **4**

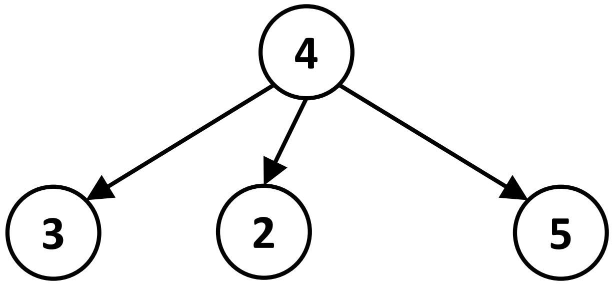
Contents of Q :

d =	0
	4

d =	1	1
	3	**2**

BFS(G, **4**)





Adj List of G :

1 → **2**

2 → **3**

3 → **1** → **2**

4 → **3** → **2** → **5**

5 → **6** → **8**

6 → **7**

7 → **6** → **4**

8 → **7** → **4**

Contents of Q :

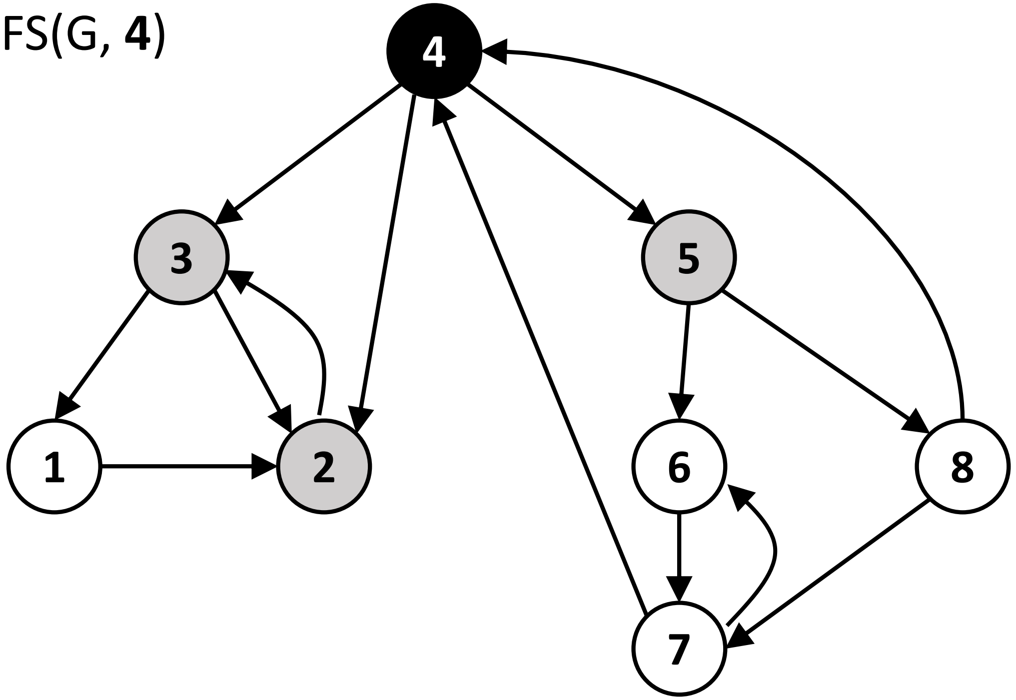
d = 0

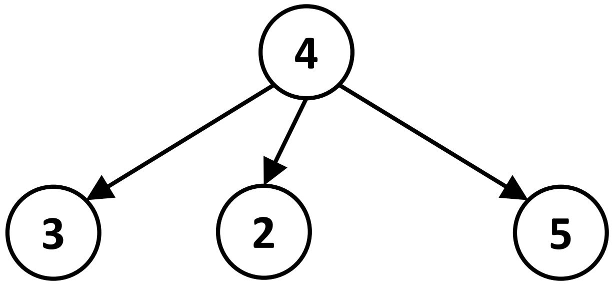
4

d = 1 1 1

3	**2**	**5**

BFS(G, **4**)





Adj List of G :

1		→ 2
2		→ 3
3		→ 1 → 2
4	✓	→ 3 → 2 → 5
5		→ 6 → 8
6		→ 7
7		→ 6 → 4
8		→ 7 → 4

Contents of Q :

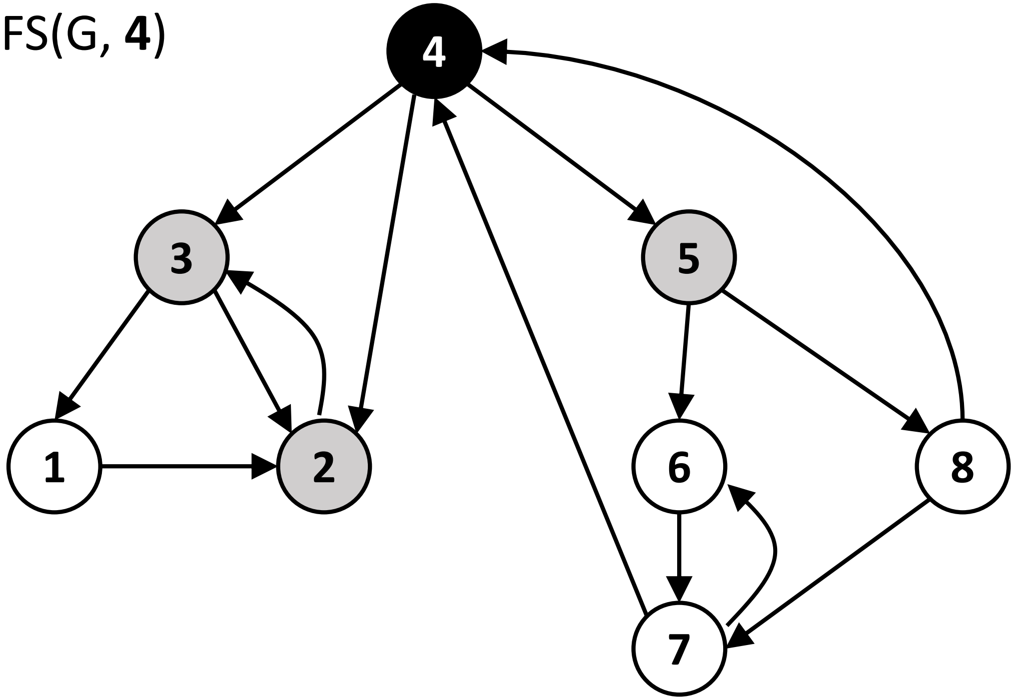
d = 0

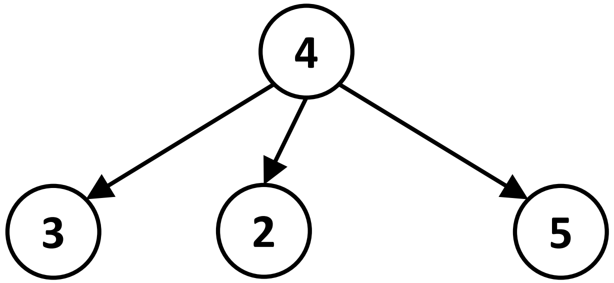
4

d = 1 1 1

3	2	5

BFS(G, **4**)





Adj List of G :

1	→ **2**
2	→ **3**
3	→ **1** → **2**
✓ 4	→ **3** → **2** → **5**
5	→ **6** → **8**
6	→ **7**
7	→ **6** → **4**
8	→ **7** → **4**

Contents of Q :

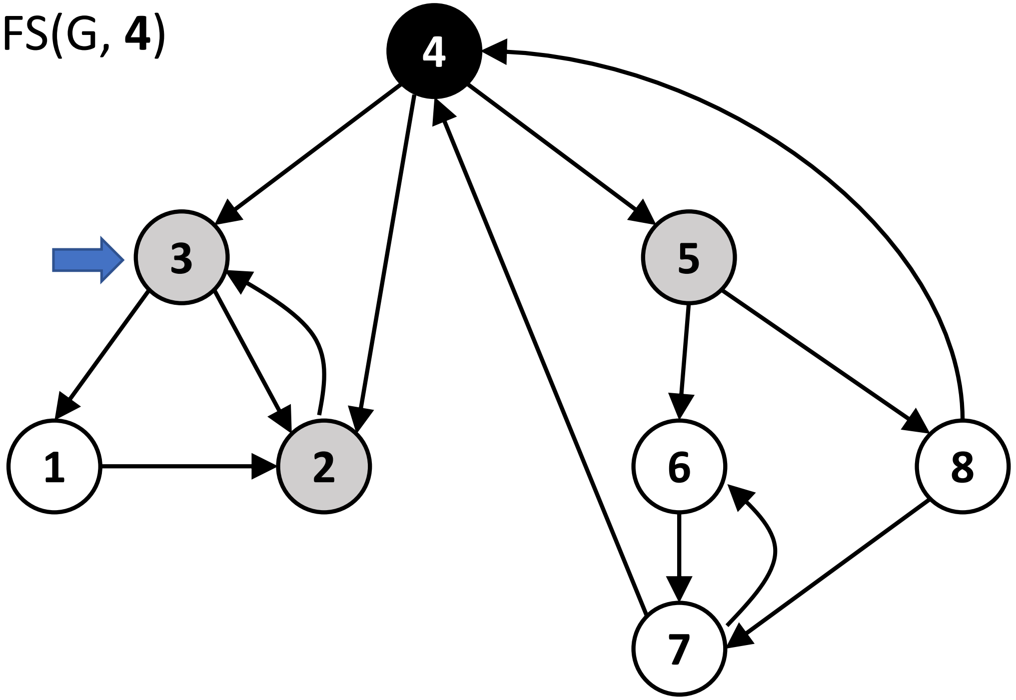
d = 0

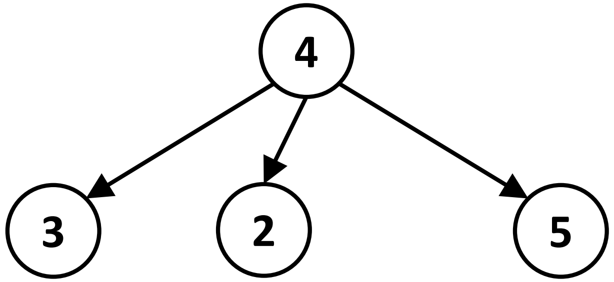
4

d = 1 1 1

3	**2**	**5**

BFS(G, **4**)





Adj List of G :

1 → 2
2 → 3
3 → 1 → 2
✓ 4 → 3 → 2 → 5
5 → 6 → 8
6 → 7
7 → 6 → 4
8 → 7 → 4

Contents of Q :

d = 0

4

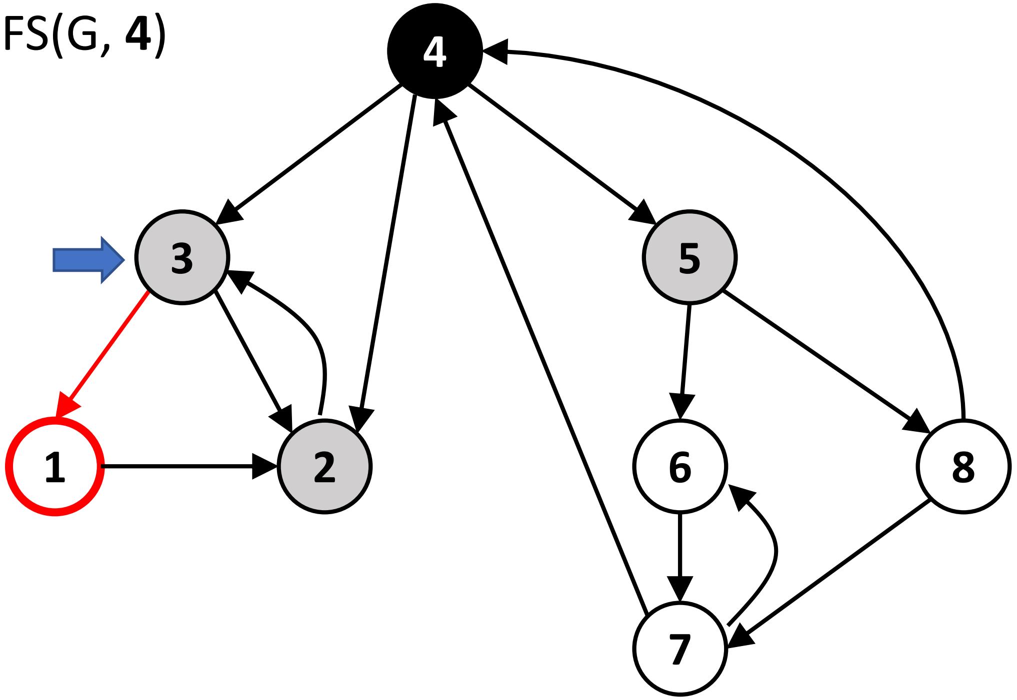
d = 1 1 1

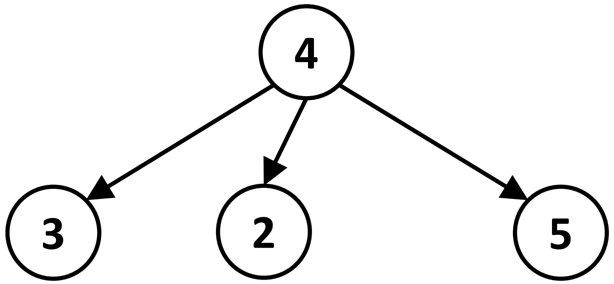
3	2	5

1 1

2	5

BFS(G, **4**)





Adj List of G :

	1	→ 2
	2	→ 3
	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

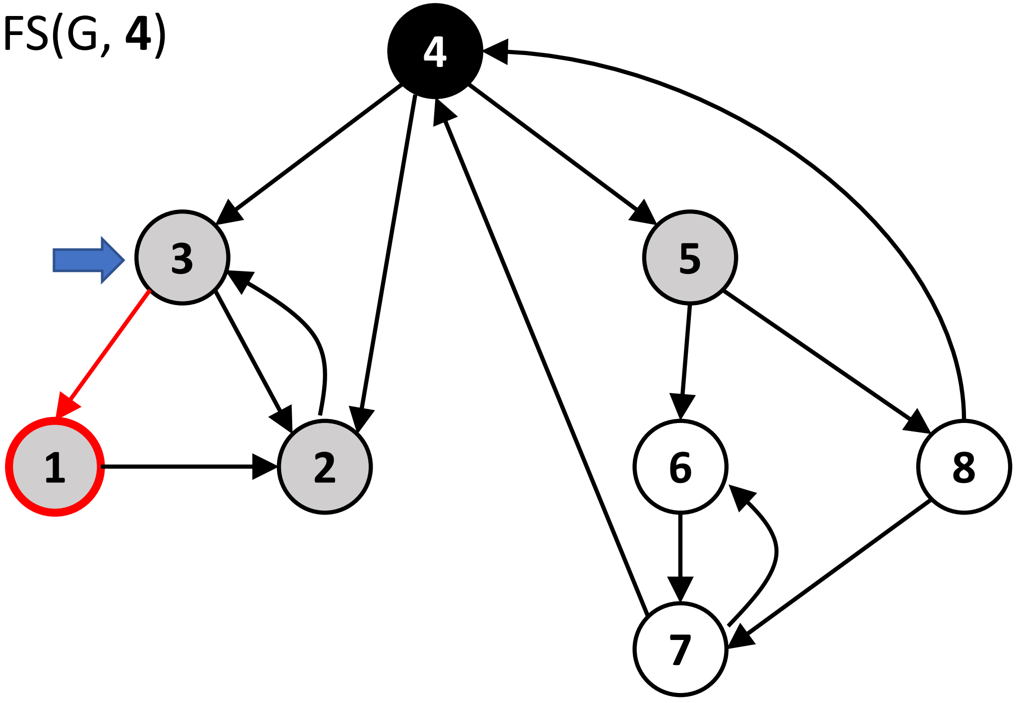
Contents of Q :

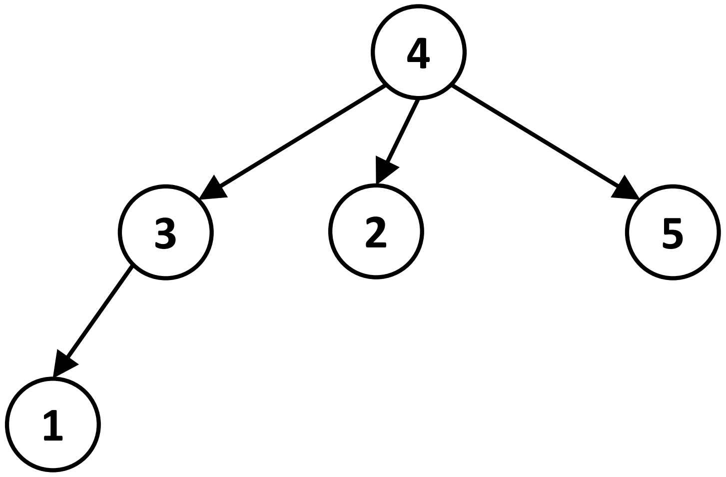
d =	0
	4

d =	1	1	1
	3	2	5

1	1
2	5

BFS(G, **4**)





Adj List of G :

1	→ 2
2	→ 3
3	→ 1 → 2
✓ 4	→ 3 → 2 → 5
5	→ 6 → 8
6	→ 7
7	→ 6 → 4
8	→ 7 → 4

Contents of Q :

d = 0

4

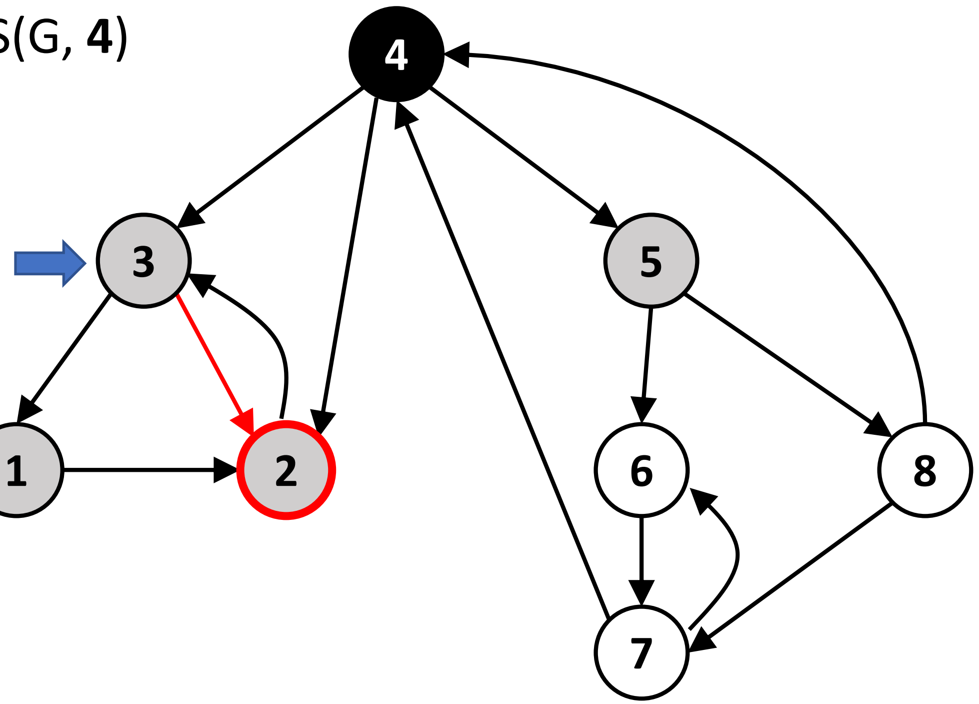
d = 1 1 1

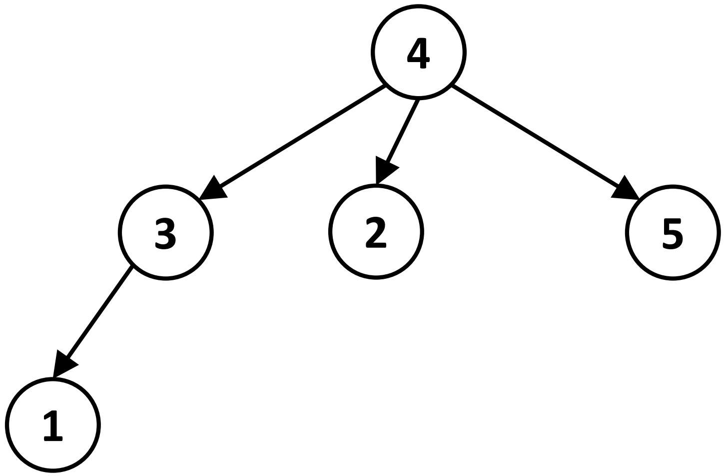
3	2	5

1 1 2

2	5	1

BFS(G, **4**)





Adj List of G :

1	→ **2**
2	→ **3**
3	→ **1** → **2**
✓ 4	→ **3** → **2** → **5**
5	→ **6** → **8**
6	→ **7**
7	→ **6** → **4**
8	→ **7** → **4**

Contents of Q :

d = 0

4

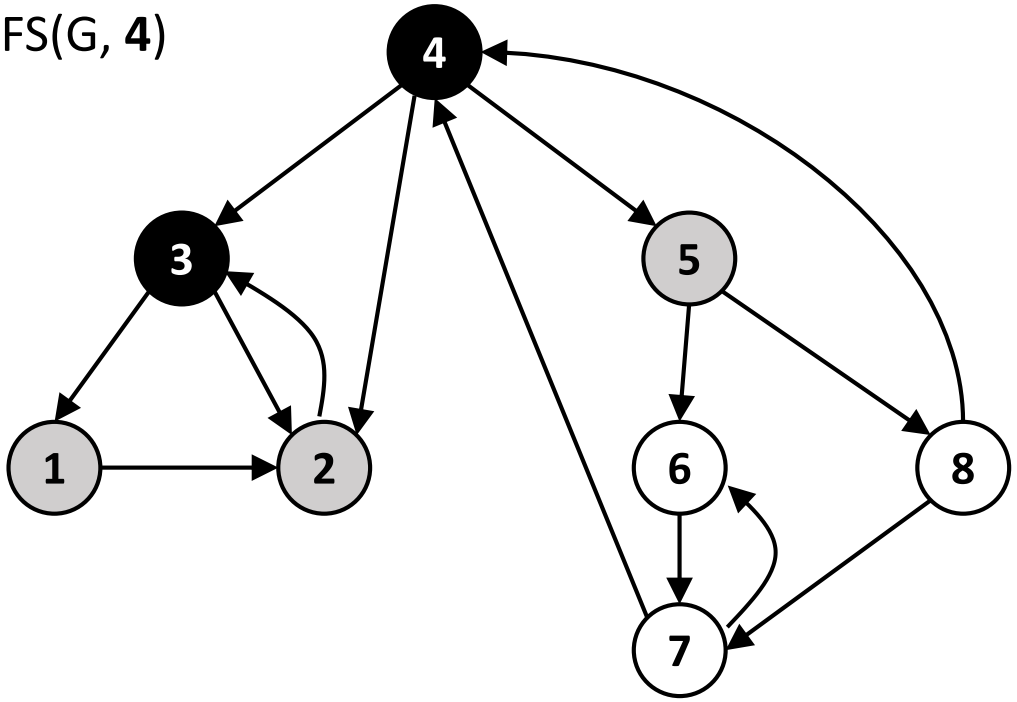
d = 1 1 1

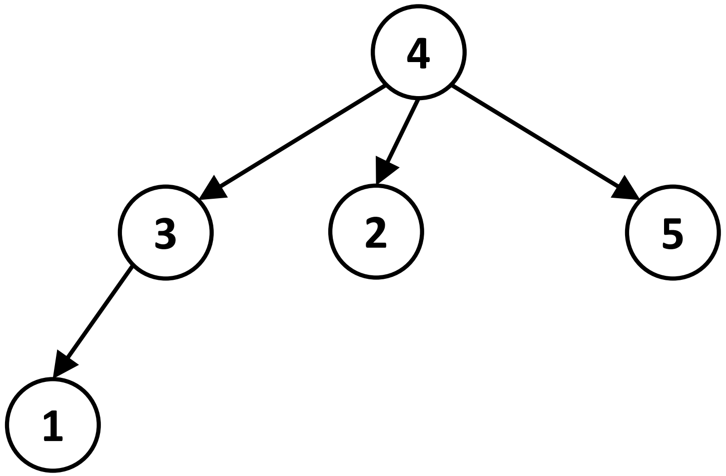
3	**2**	**5**

1 1 2

2	**5**	**1**

BFS(G, **4**)





Adj List of G :

1 → **2**

2 → **3**

✓ 3 → **1** → **2**

✓ 4 → **3** → **2** → **5**

5 → **6** → **8**

6 → **7**

7 → **6** → **4**

8 → **7** → **4**

Contents of Q :

d = 0

4

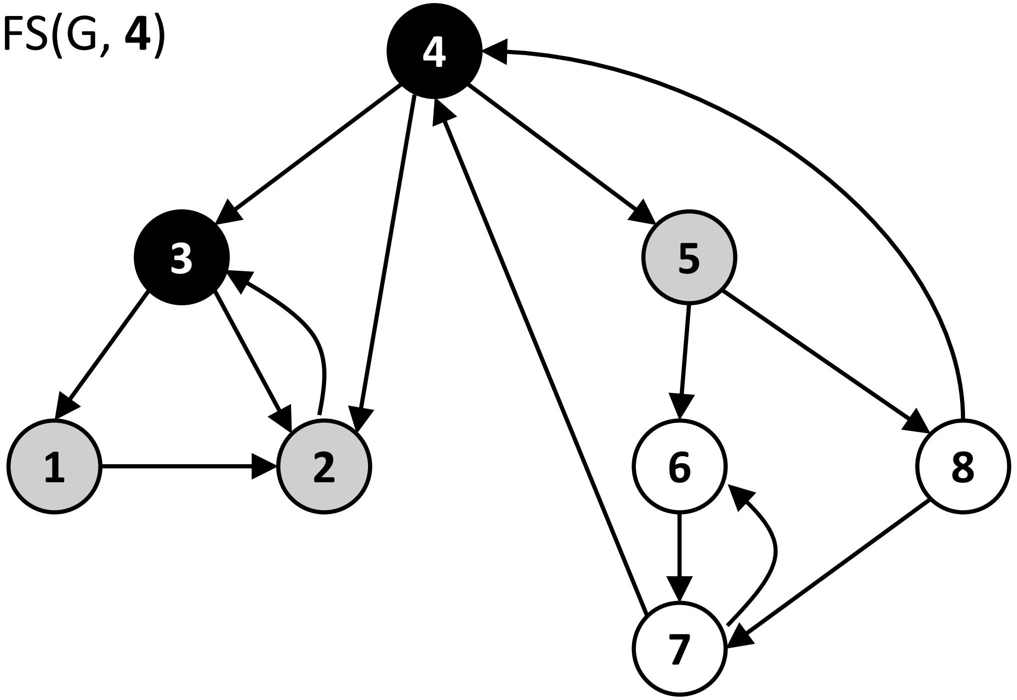
d = 1 1 1

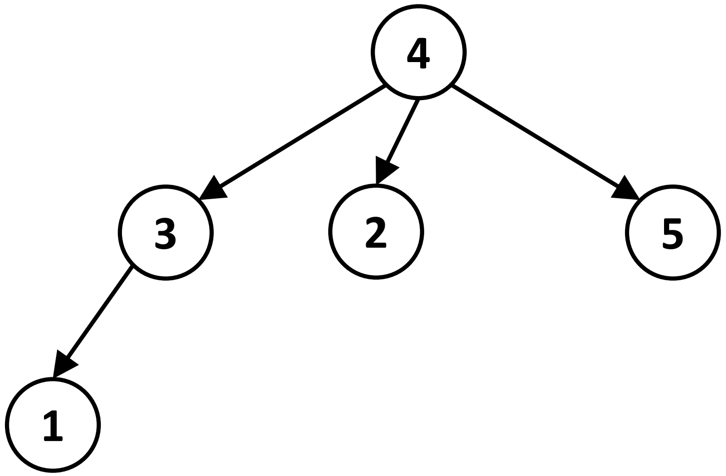
3	2	5

1 1 2

2	5	1

BFS(G, **4**)





Adj List of G :

	1	→ 2
	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

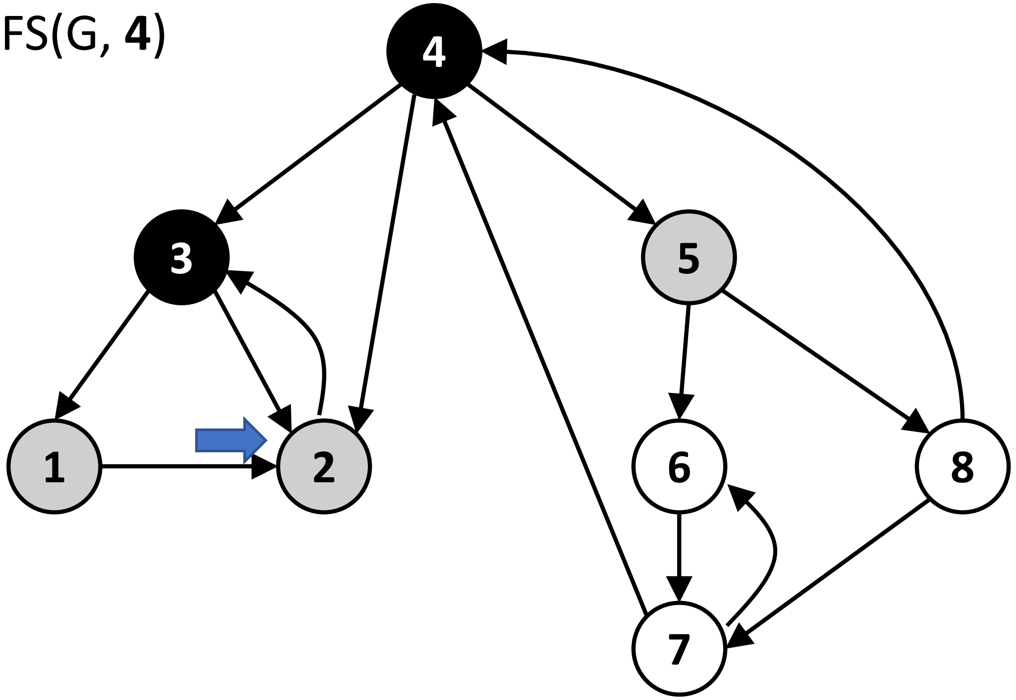
d = 1 1 1

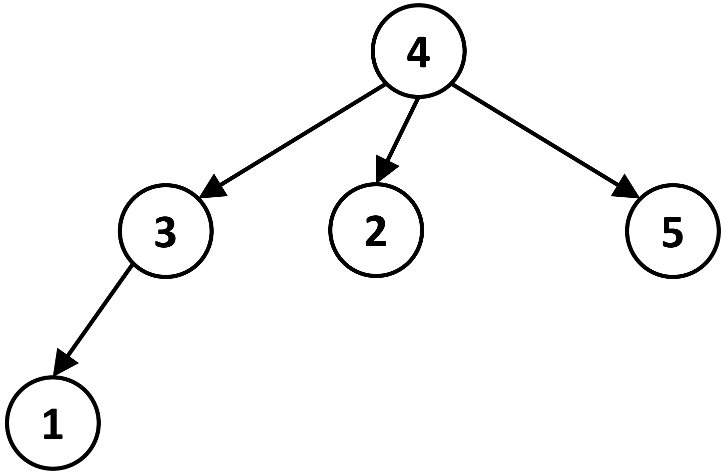
3	2	5

1 1 2

2	5	1

BFS(G, **4**)





Adj List of G :

	1	→ **2**
	2	→ **3**
✓	**3**	→ **1** → **2**
✓	**4**	→ **3** → **2** → **5**
	5	→ **6** → **8**
	6	→ **7**
	7	→ **6** → **4**
	8	→ **7** → **4**

Contents of Q :

d = 0

4

d = 1 1 1

3	**2**	**5**

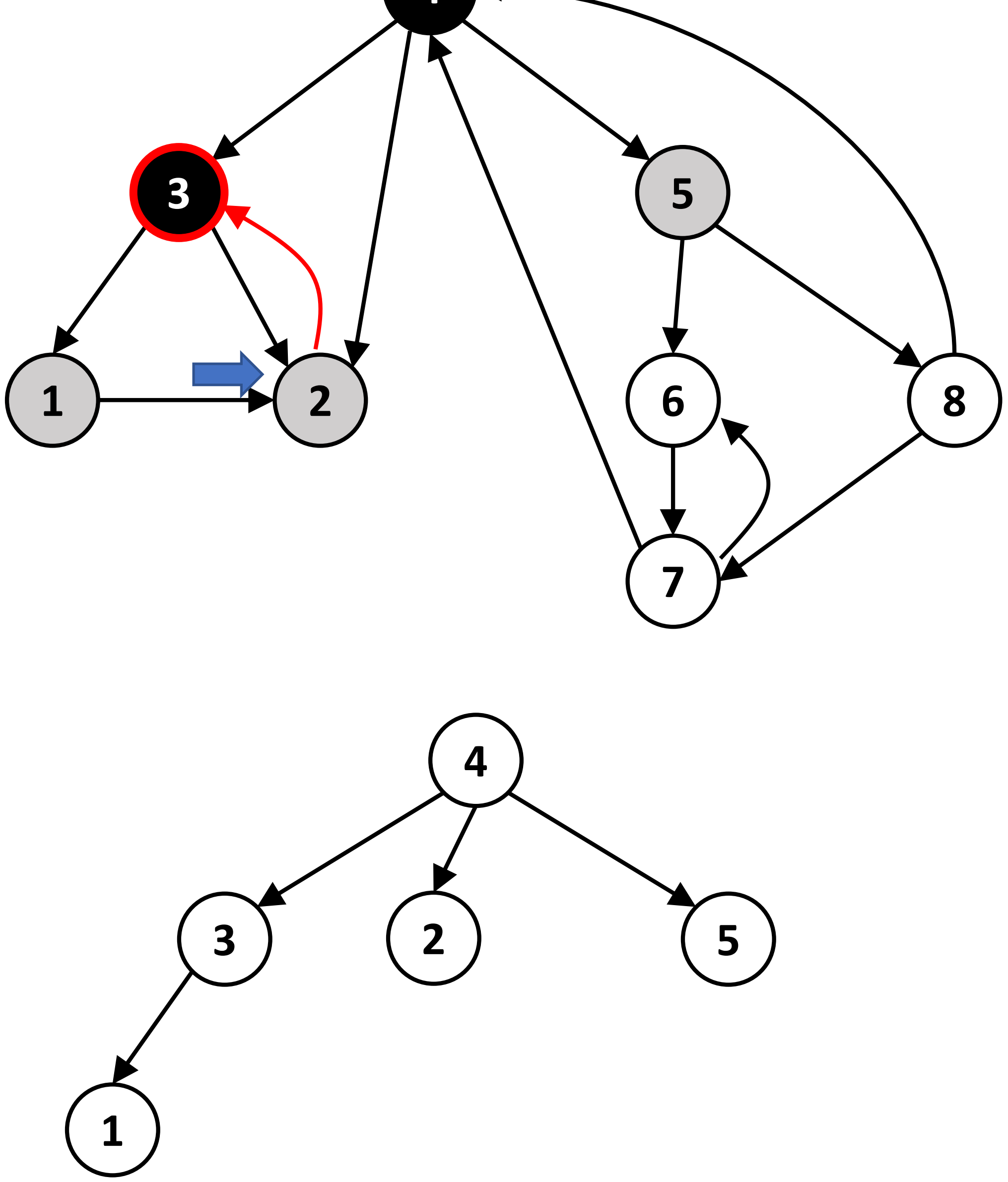
1 1 2

2	**5**	**1**

1 2

5	**1**

BFS(G, **4**)



Adj List of G :

1	→ **2**	
➡ 2	→ **3**	
✓ 3	→ **1** → **2**	
✓ 4	→ **3** → **2** → **5**	
5	→ **6** → **8**	
6	→ **7**	
7	→ **6** → **4**	
8	→ **7** → **4**	

Contents of Q :

d = 0

4

d = 1 1 1

3	**2**	**5**

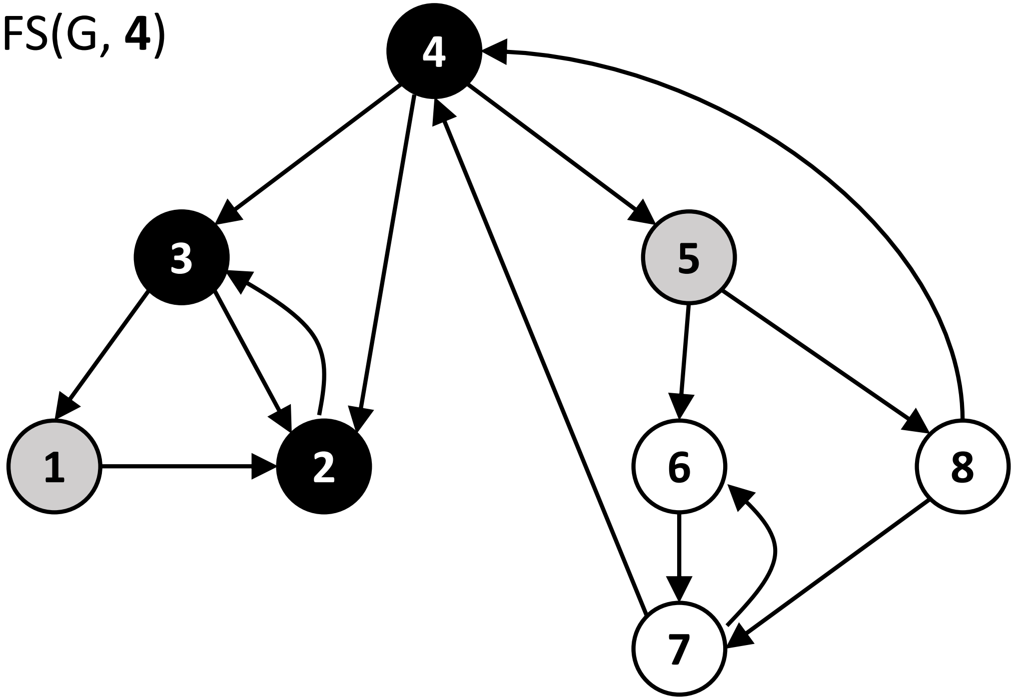
1 1 2

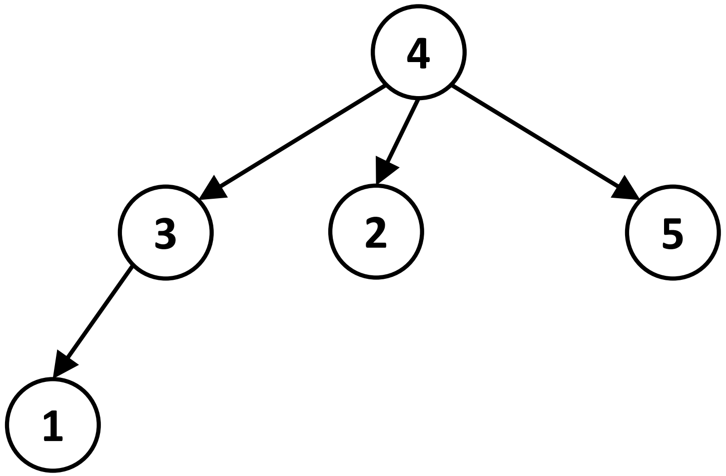
2	**5**	**1**

1 2

5	**1**

BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

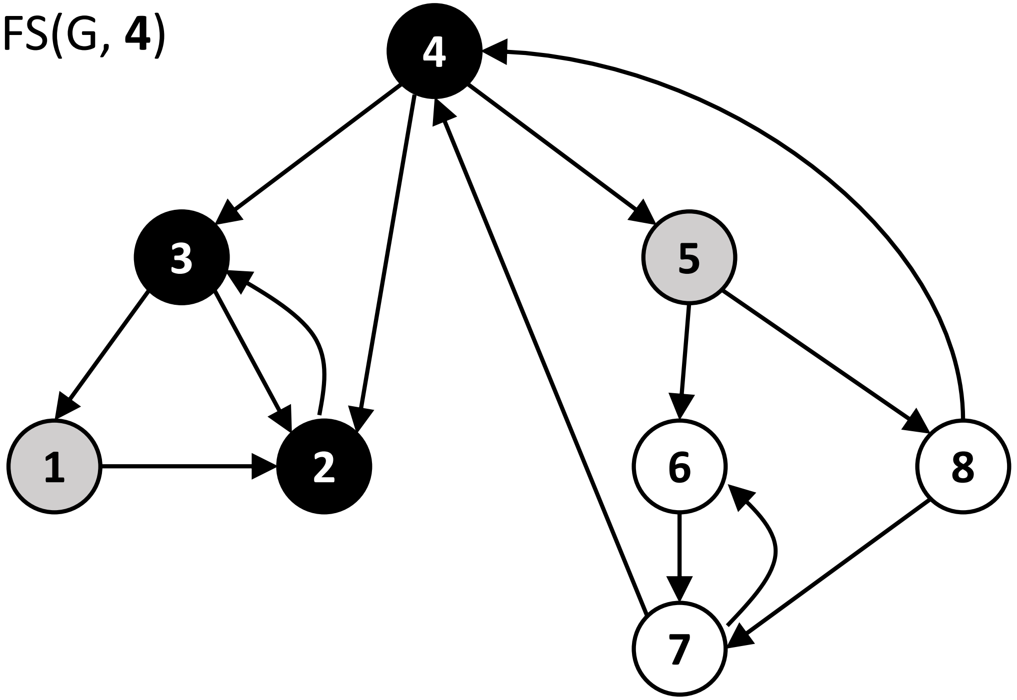
1 1 2

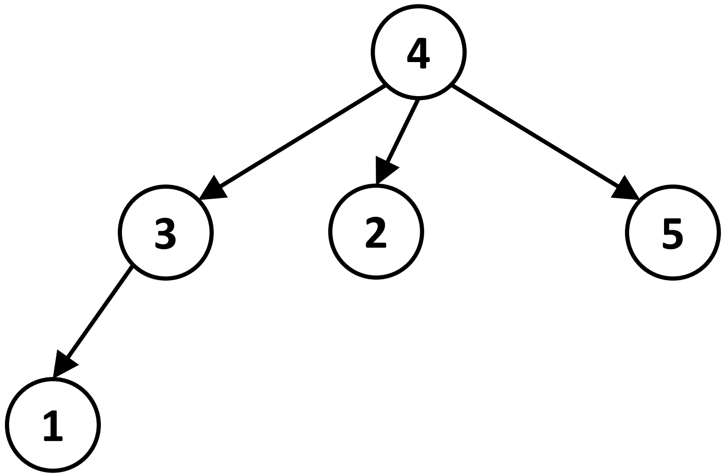
2	5	1

1 2

5	1

BFS(G, **4**)





Adj List of G :

	1	→ **2**
✓	**2**	→ **3**
✓	**3**	→ **1** → **2**
✓	**4**	→ **3** → **2** → **5**
	5	→ **6** → **8**
	6	→ **7**
	7	→ **6** → **4**
	8	→ **7** → **4**

Contents of Q :

d = 0

4

d = 1 1 1

3	**2**	**5**

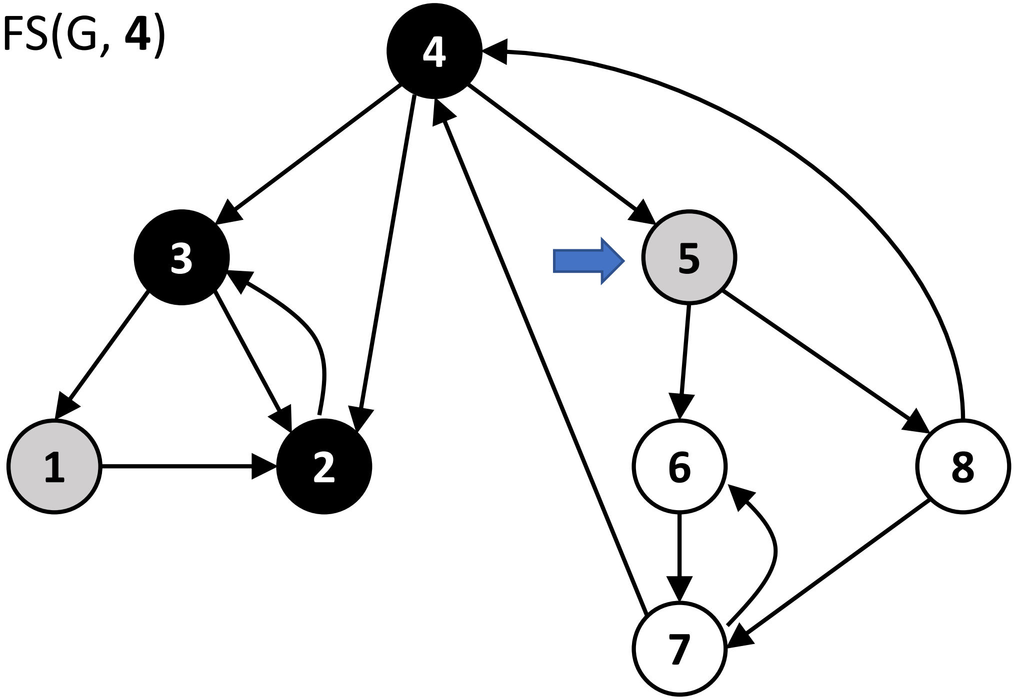
1 1 2

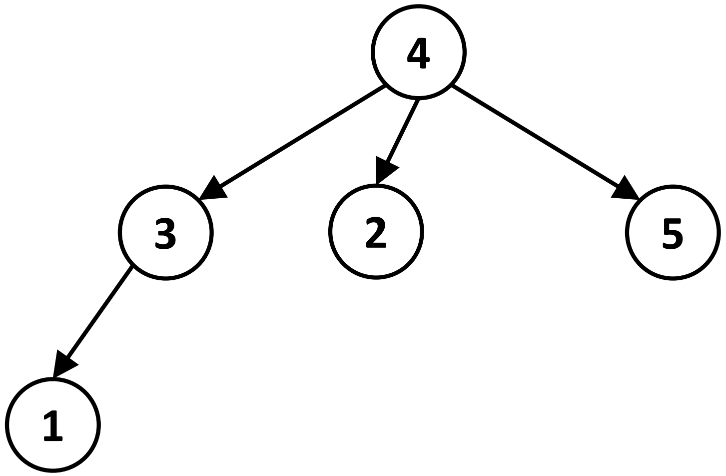
2	**5**	**1**

1 2

5	**1**

BFS(G, **4**)





Adj List of G :

	1	2
✓	2	3
✓	3	1 → 2
✓	4	3 → 2 → 5
➡	5	6 → 8
	6	7
	7	6 → 4
	8	7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

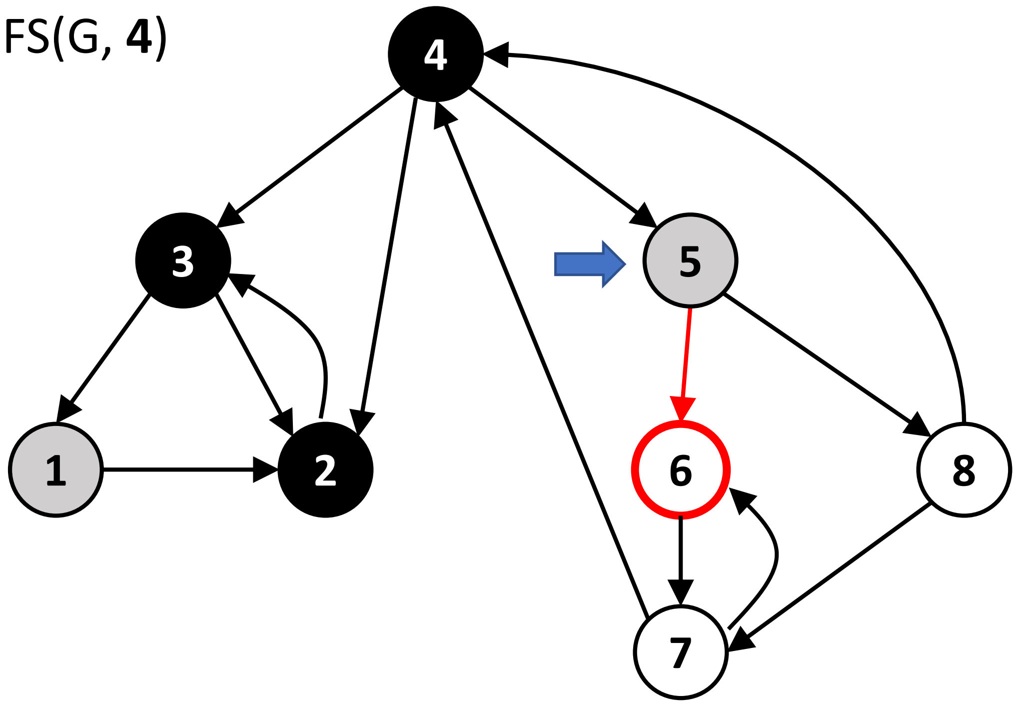
1 2

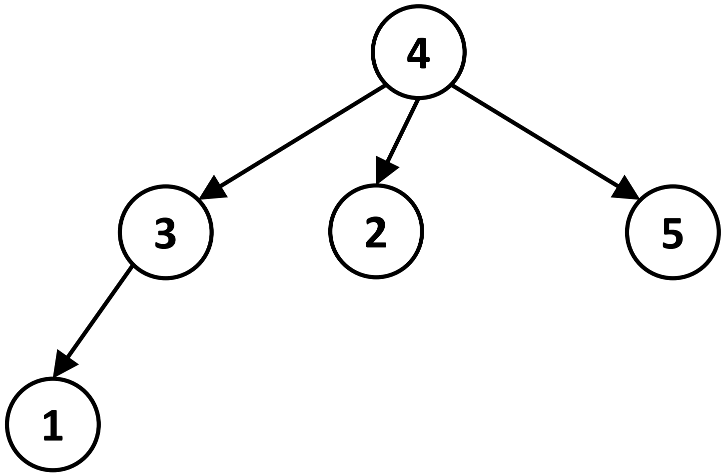
5	1

2

1

BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

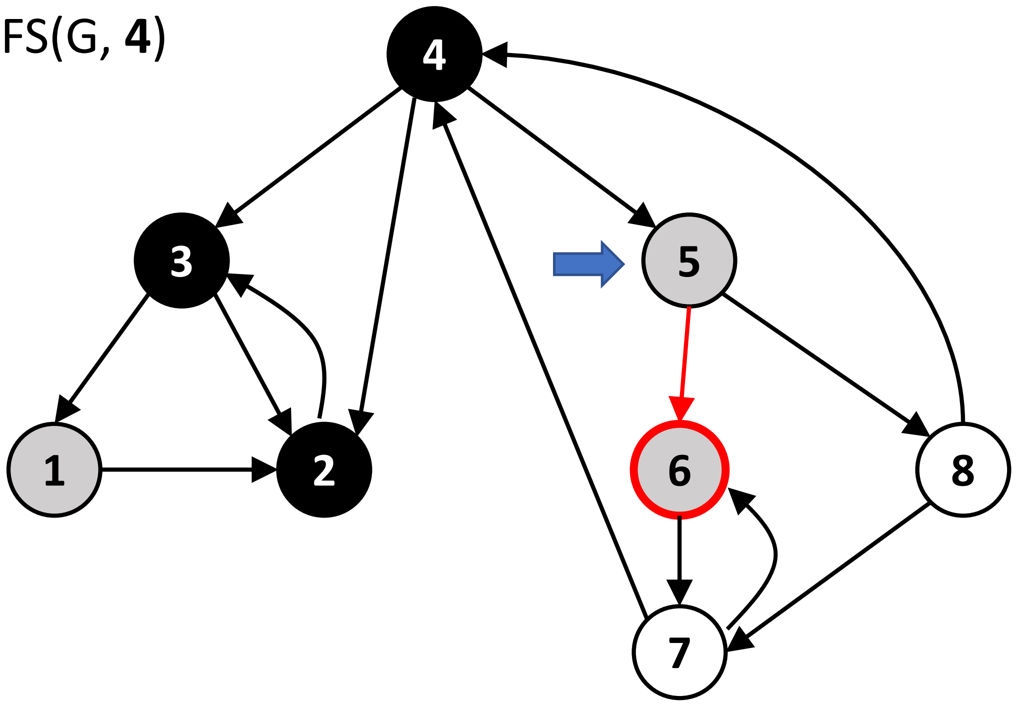
1 2

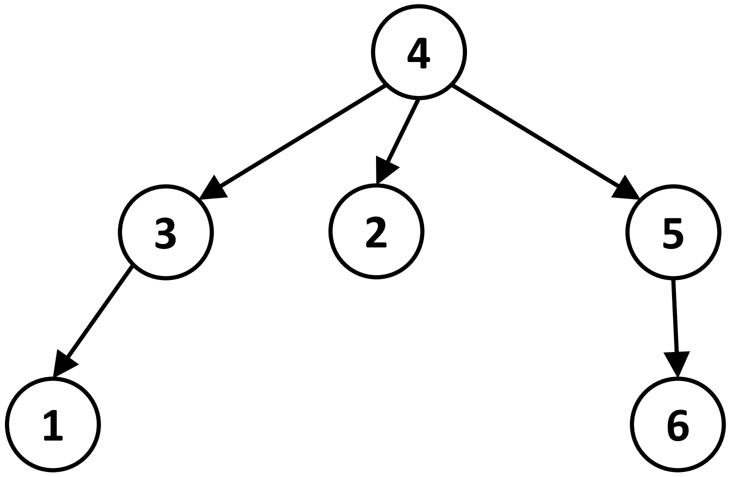
5	1

2

1

BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
➡	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d =

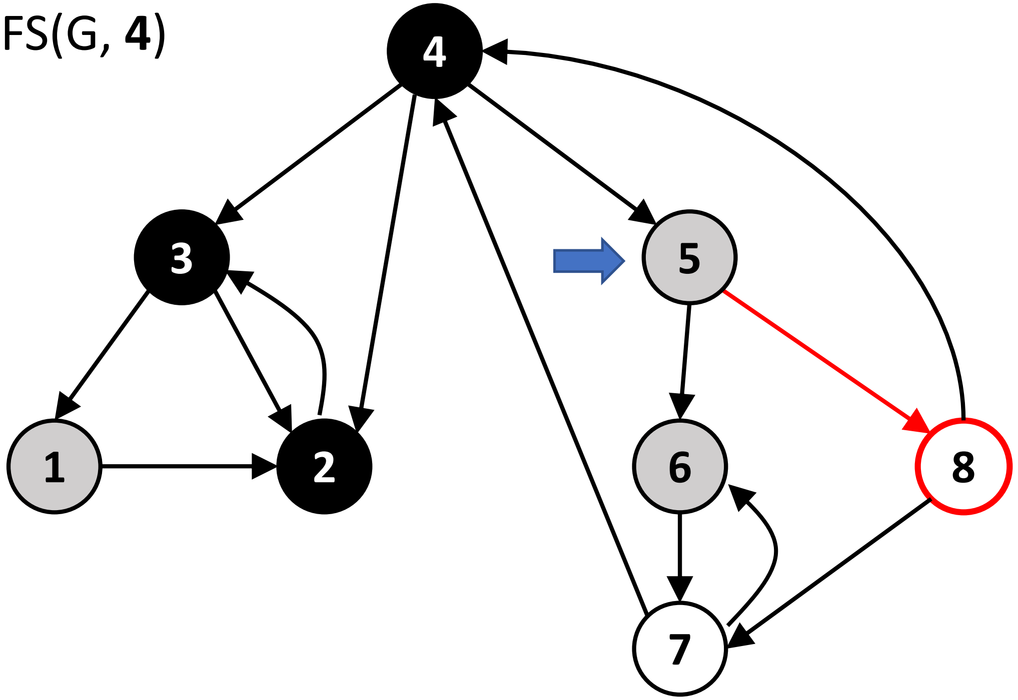
1	1	1
3	2	5

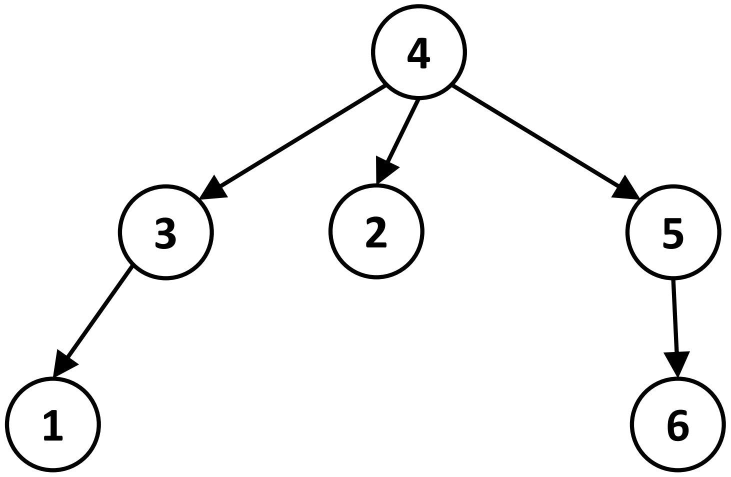
1	1	2
2	5	1

1	2
5	1

2	2
1	6

BFS(G, **4**)





Adj List of G :

	1	→ **2**
✓	**2**	→ **3**
✓	**3**	→ **1** → **2**
✓	**4**	→ **3** → **2** → **5**
➡	**5**	→ **6** → **8**
	6	→ **7**
	7	→ **6** → **4**
	8	→ **7** → **4**

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

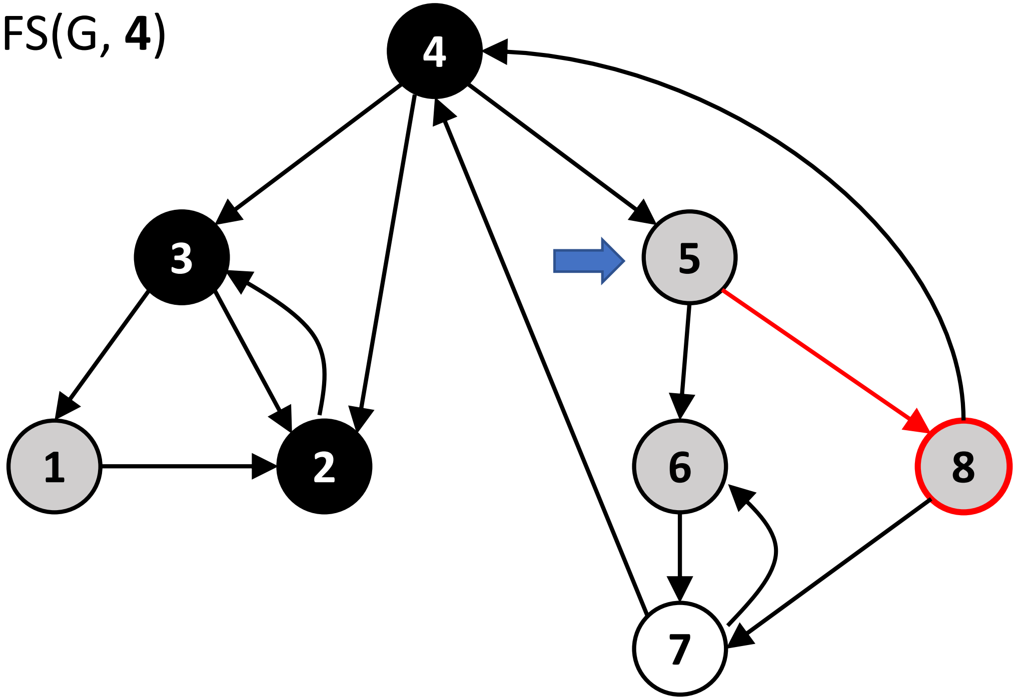
1 2

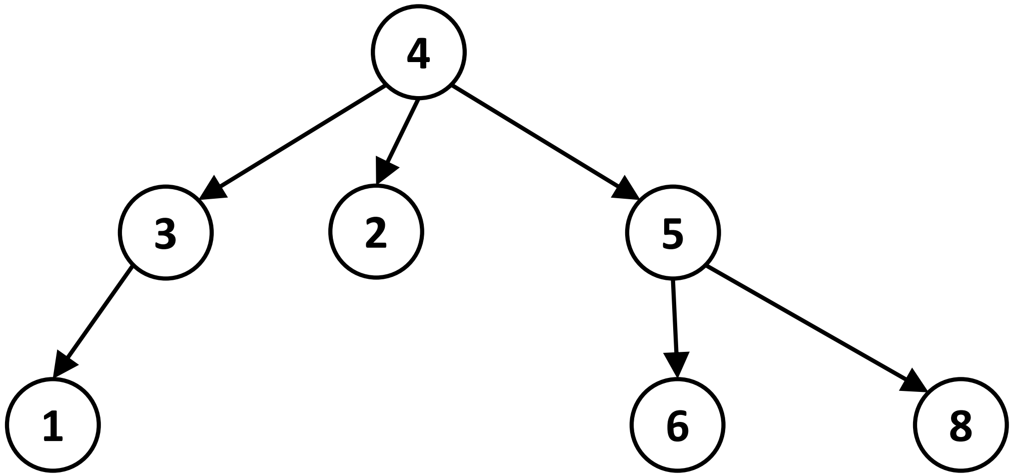
5	1

2 2

1	6

BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
➡	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

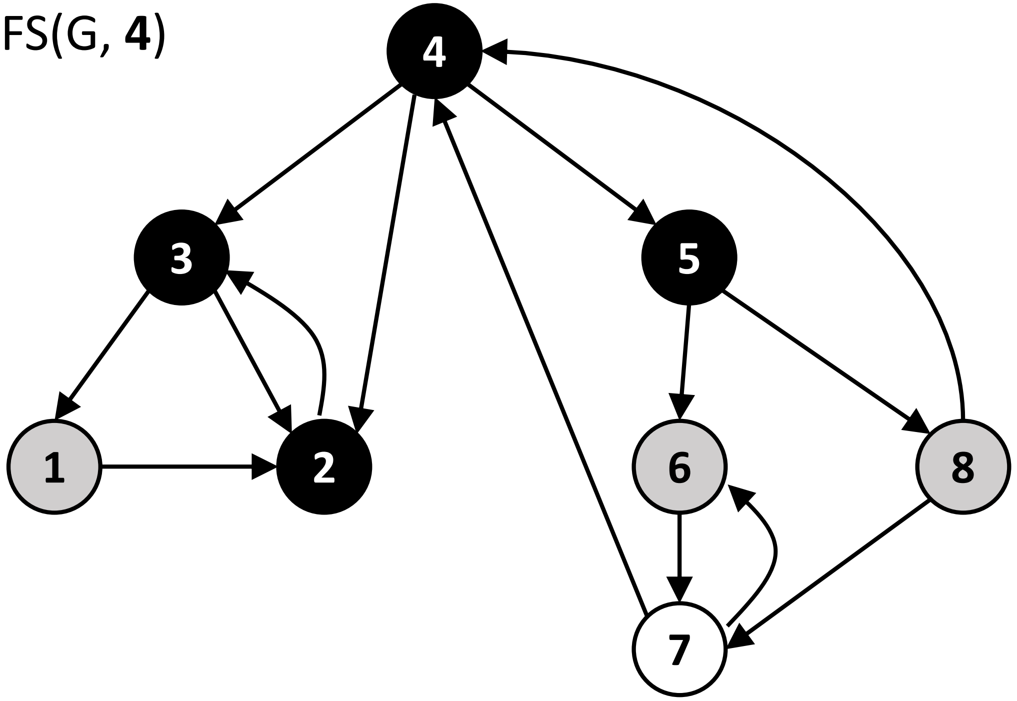
1 2

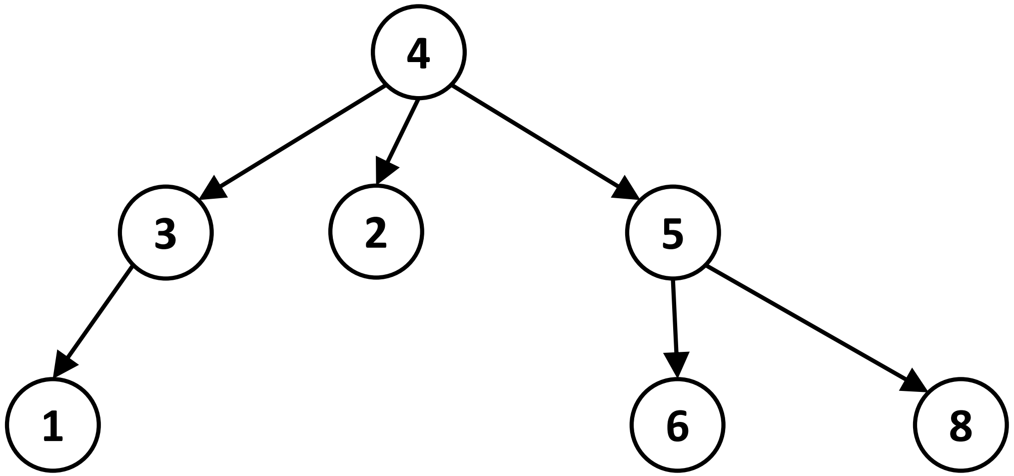
5	1

2 2 2

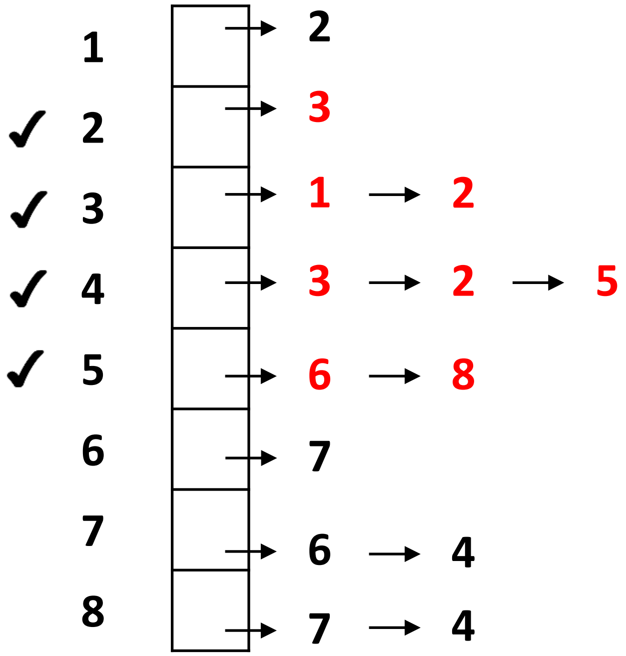
1	6	8

BFS(G, **4**)

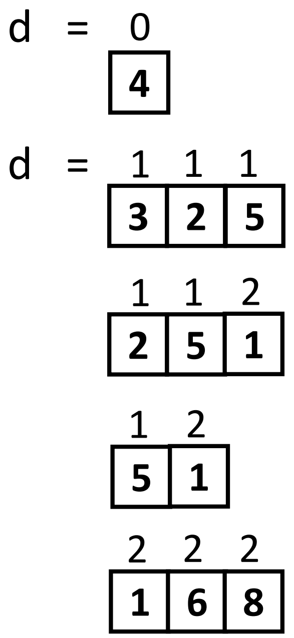




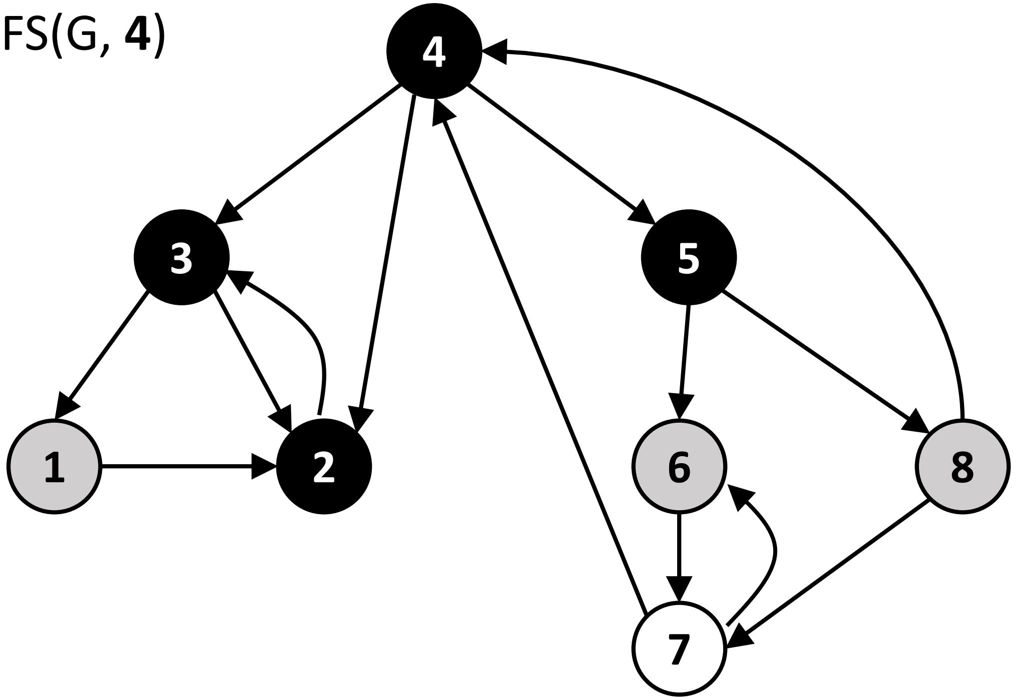
Adj List of G :

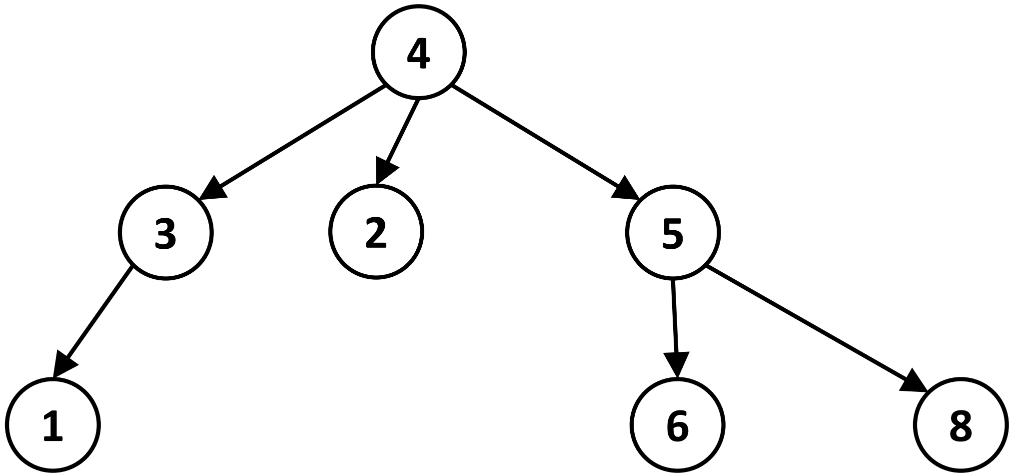


Contents of Q :



BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

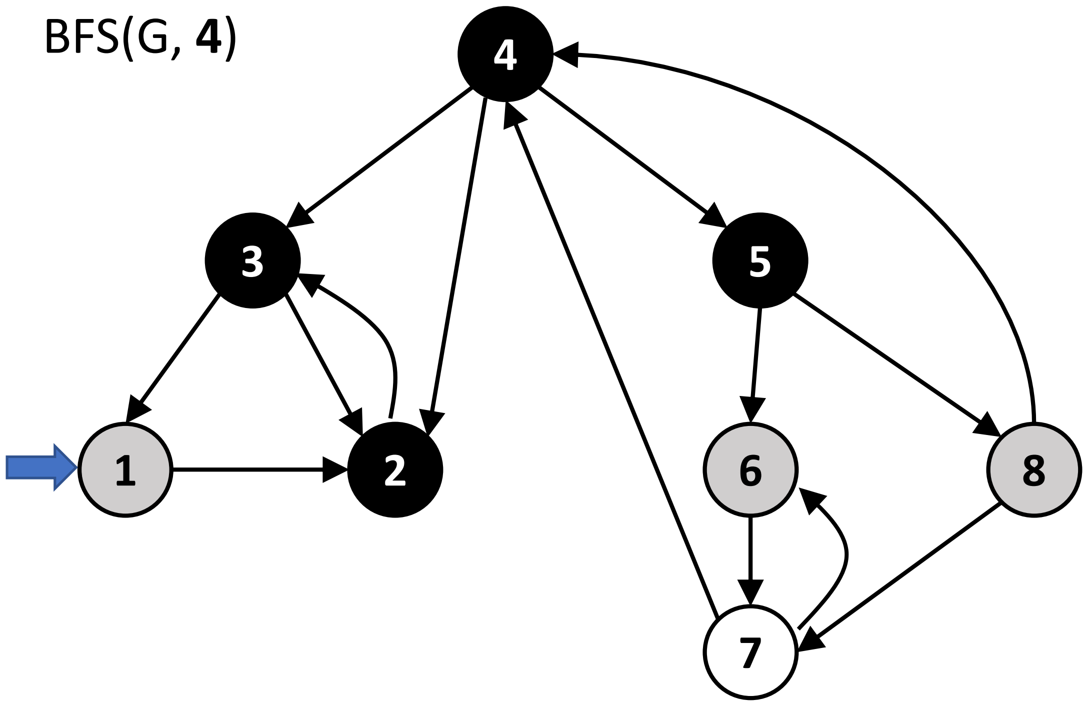
1 2

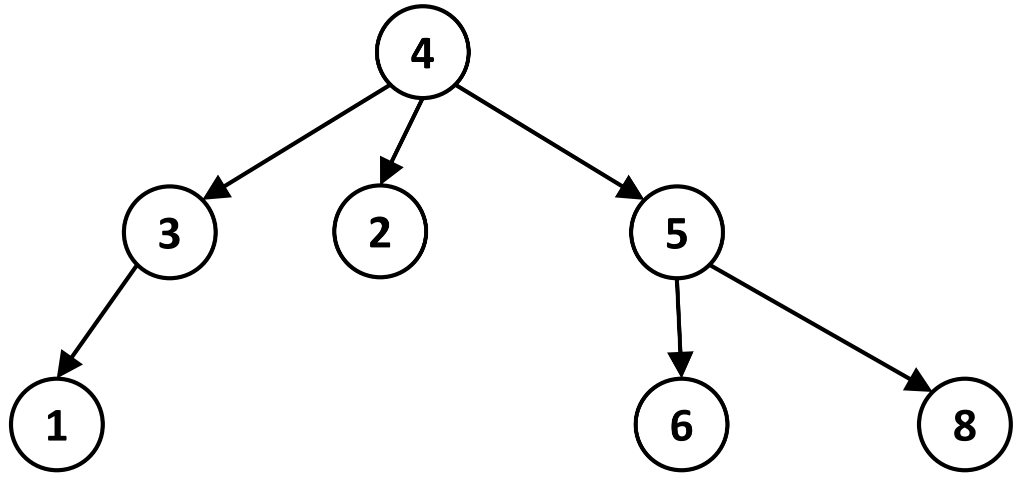
5	1

2 2 2

1	6	8

BFS(G, **4**)





Adj List of G :

	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

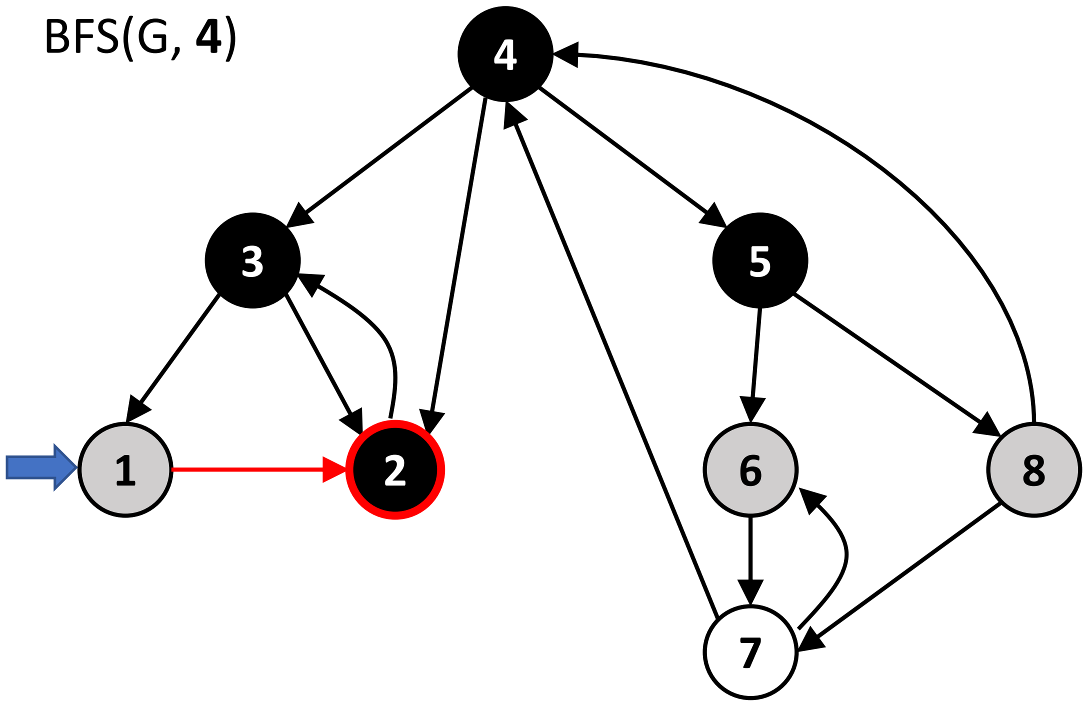
2 2 2

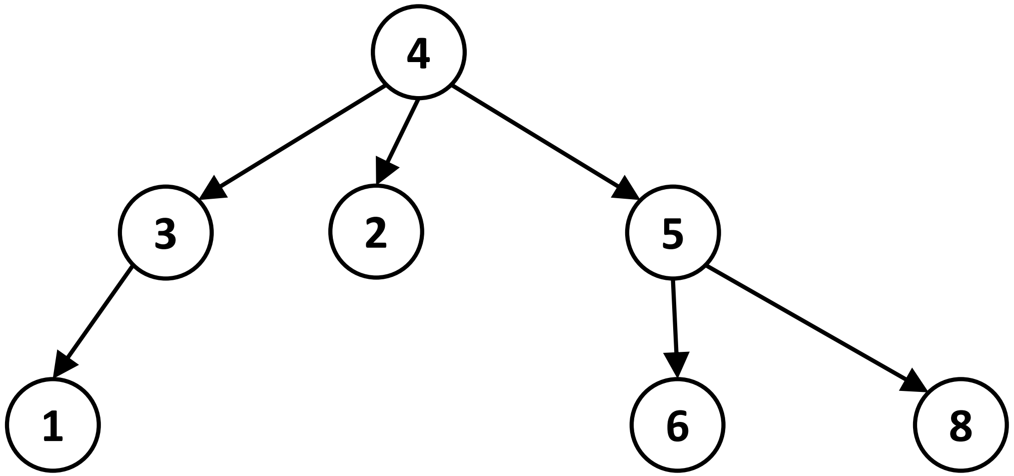
1	6	8

2 2

6	8

BFS(G, **4**)





Adj List of G :

1	→ 2
✓ 2	→ 3
✓ 3	→ 1 → 2
✓ 4	→ 3 → 2 → 5
✓ 5	→ 6 → 8
6	→ 7
7	→ 6 → 4
8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

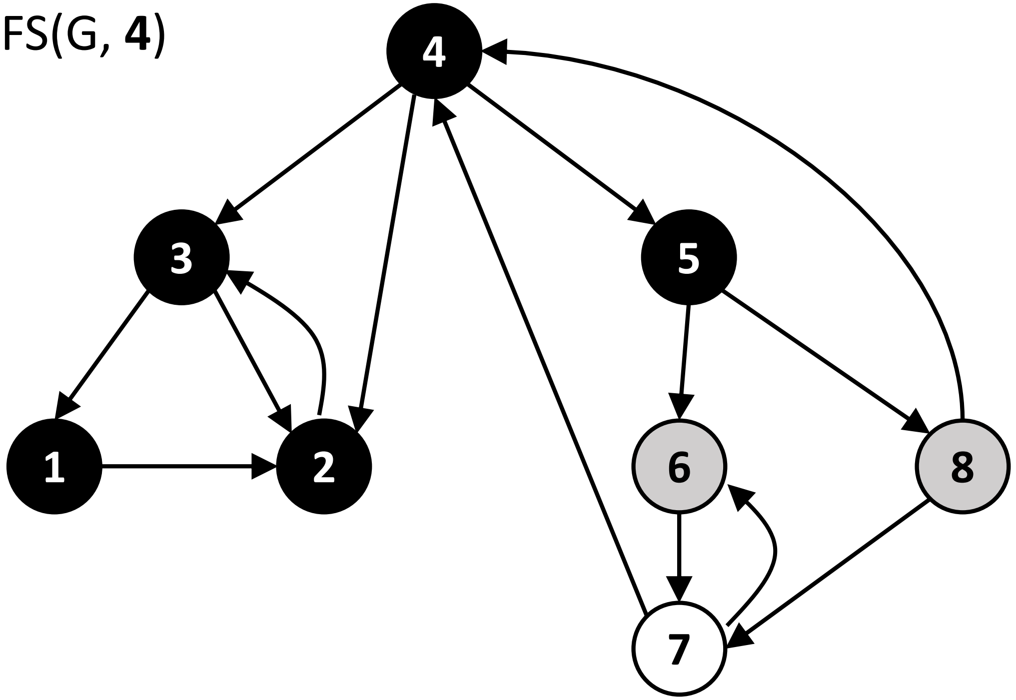
2 2 2

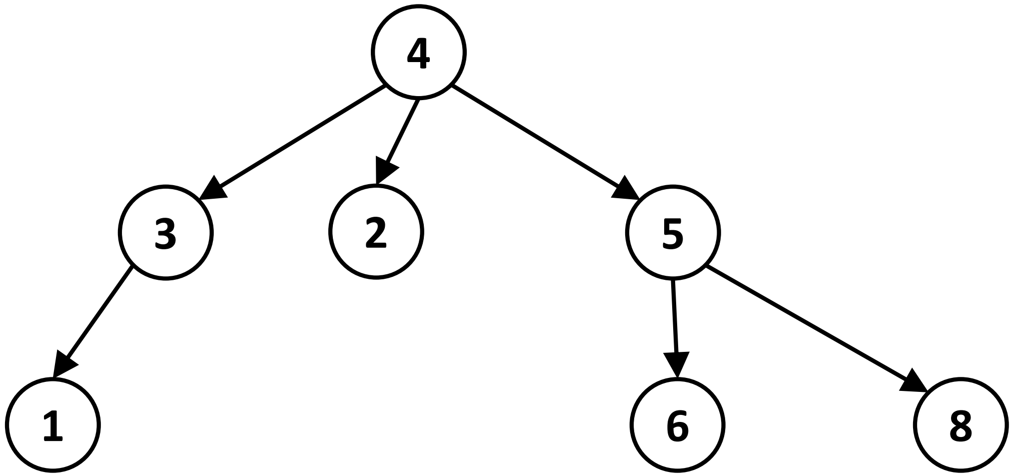
1	6	8

2 2

6	8

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

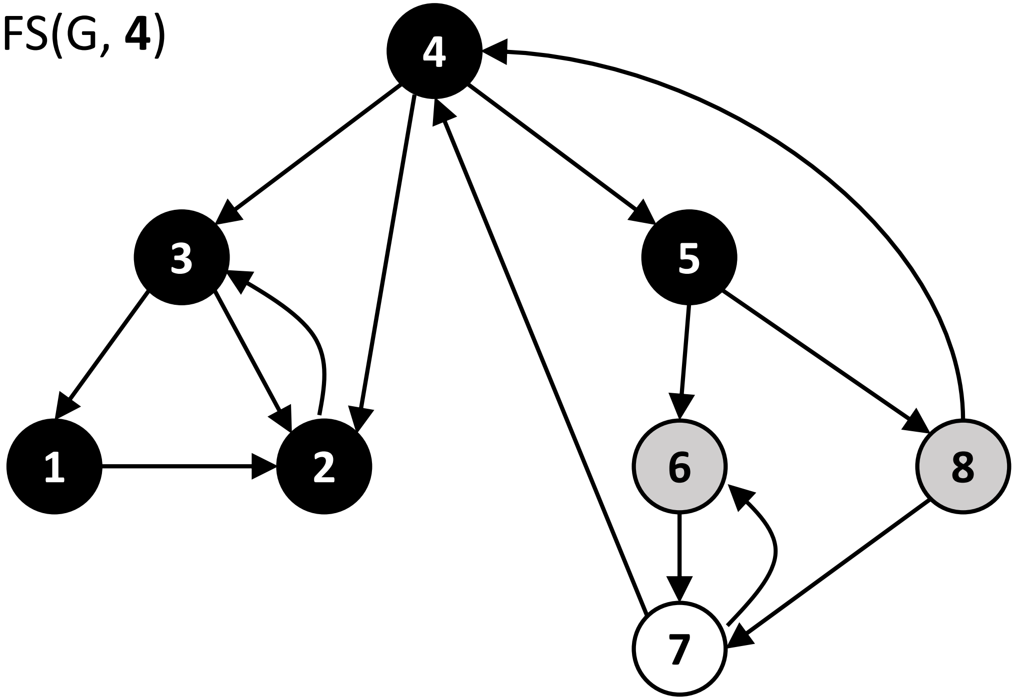
2 2 2

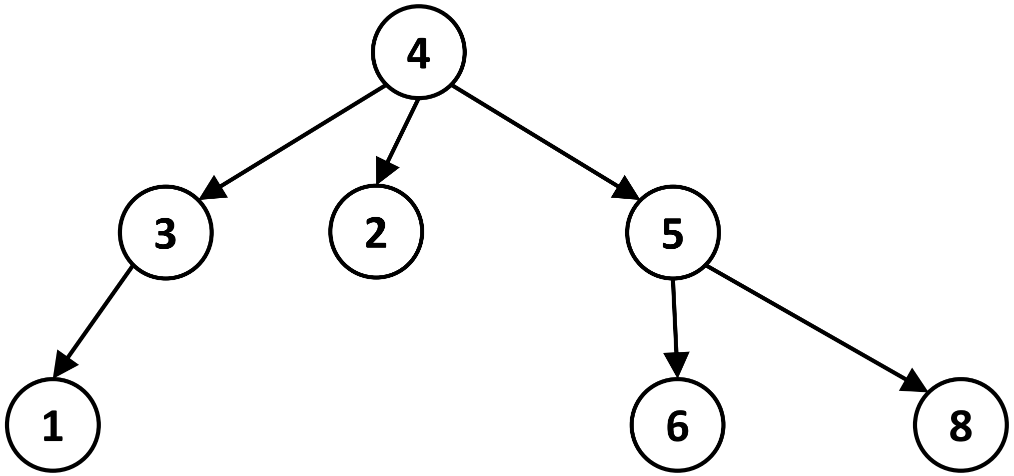
1	6	8

2 2

6	8

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

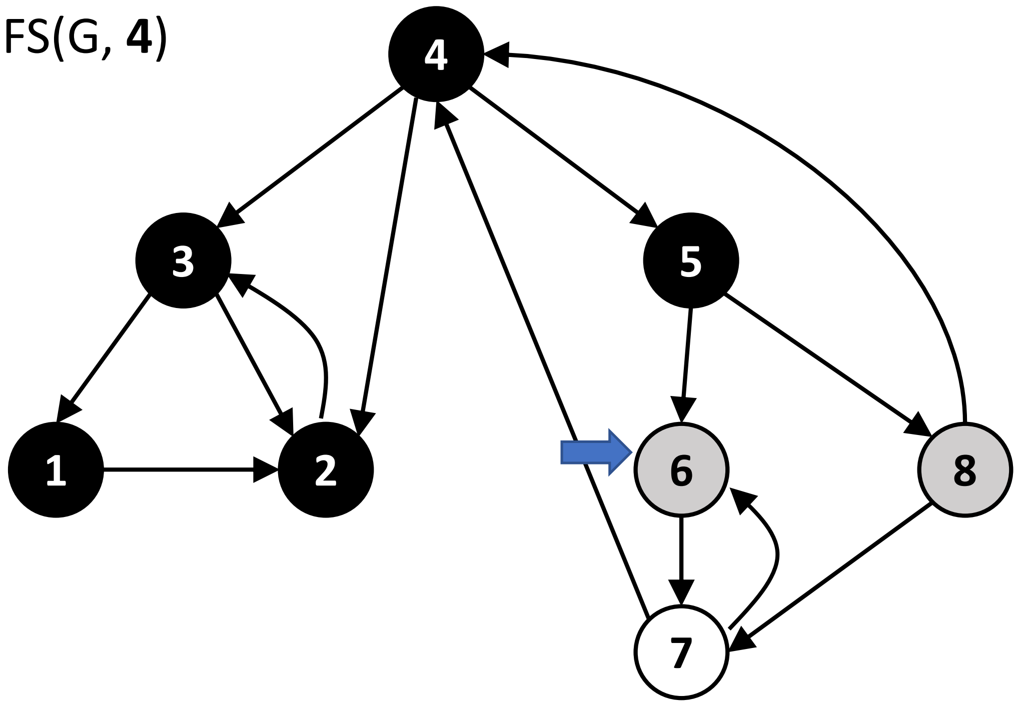
2 2 2

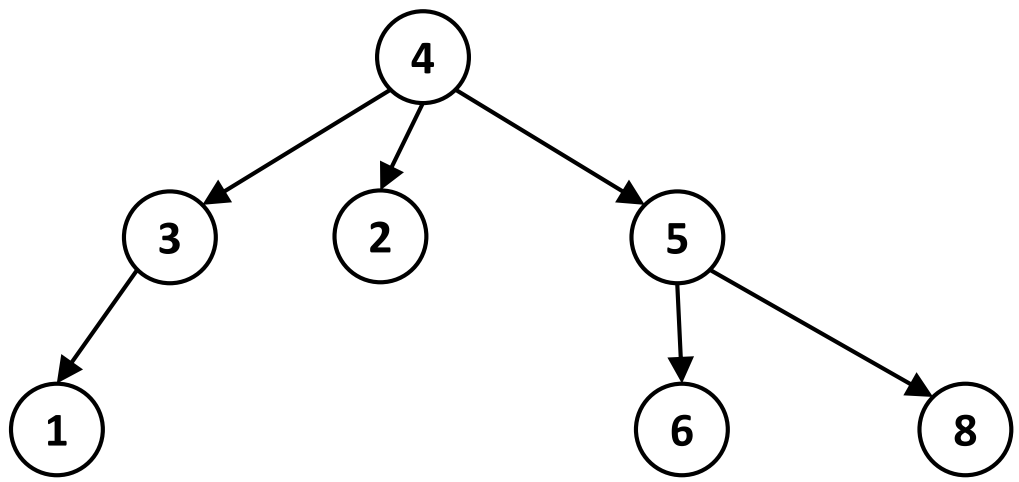
1	6	8

2 2

6	8

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
➡	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0
| 4 |

d = 1 1 1
| 3 | 2 | 5 |

1 1 2
| 2 | 5 | 1 |

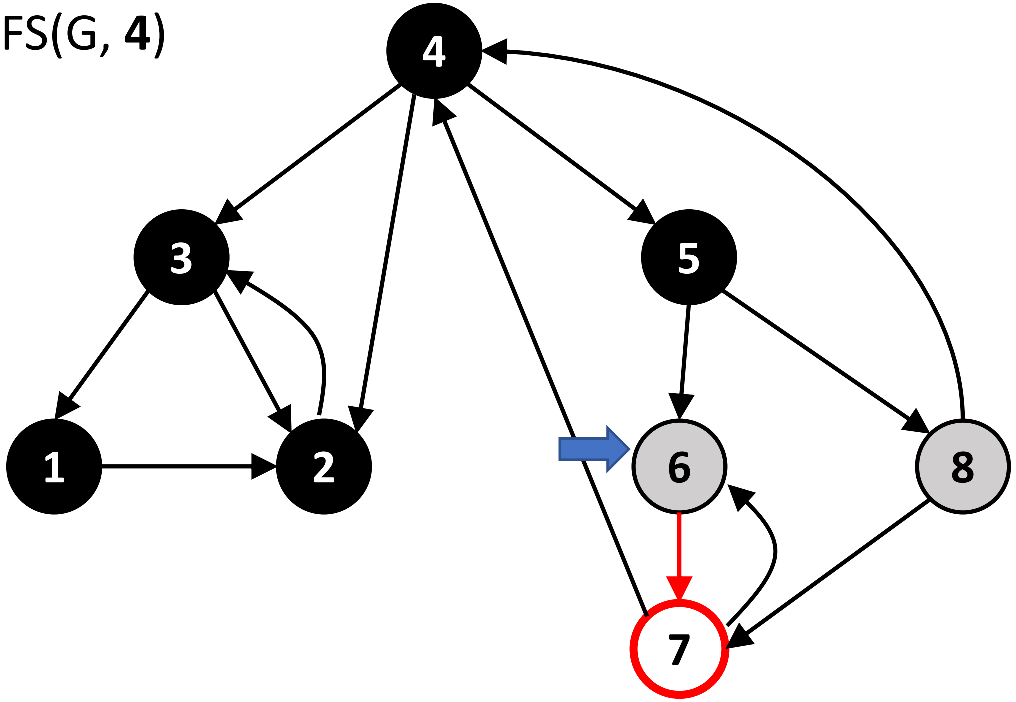
1 2
| 5 | 1 |

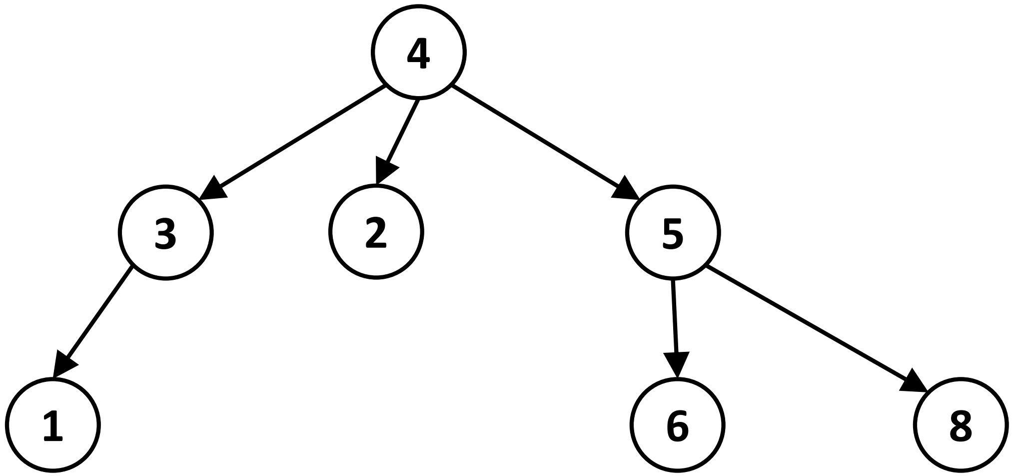
2 2 2
| 1 | 6 | 8 |

2 2
| 6 | 8 |

2
| 8 |

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

2 2 2

1	6	8

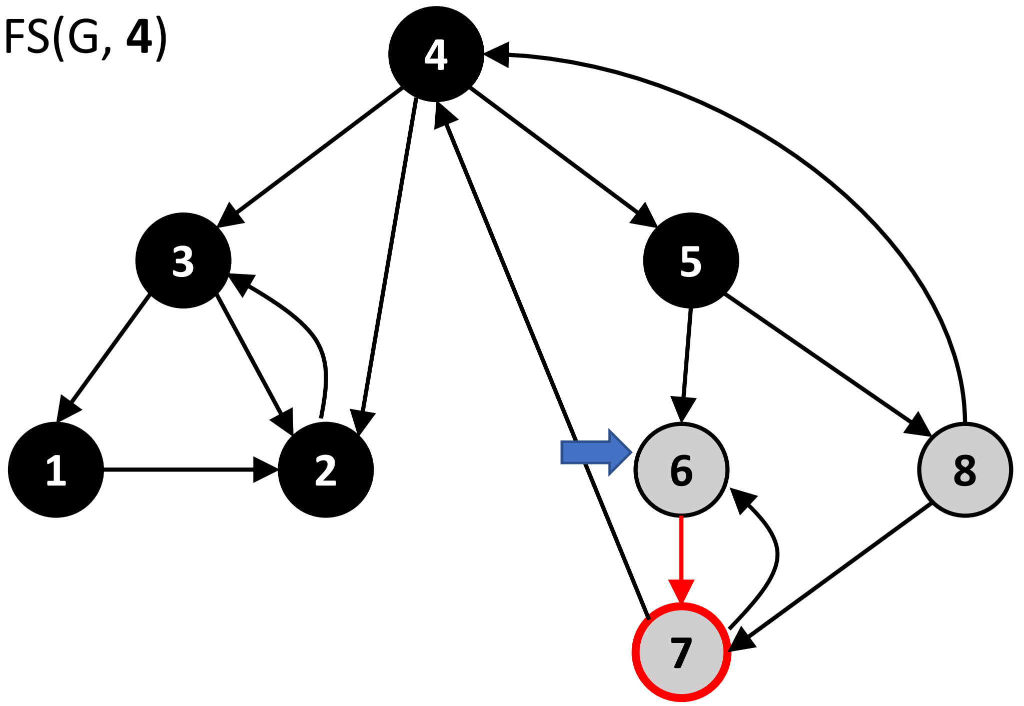
2 2

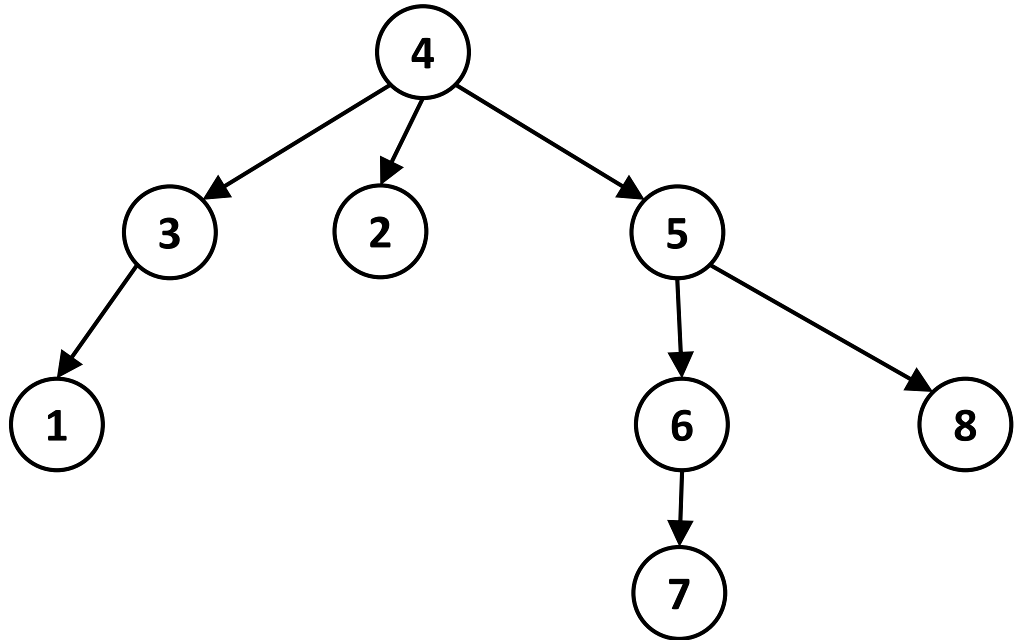
6	8

2

8

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
➡	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d =	0
	4

d =	1	1	1
	3	2	5

d	1	1	2
	2	5	1

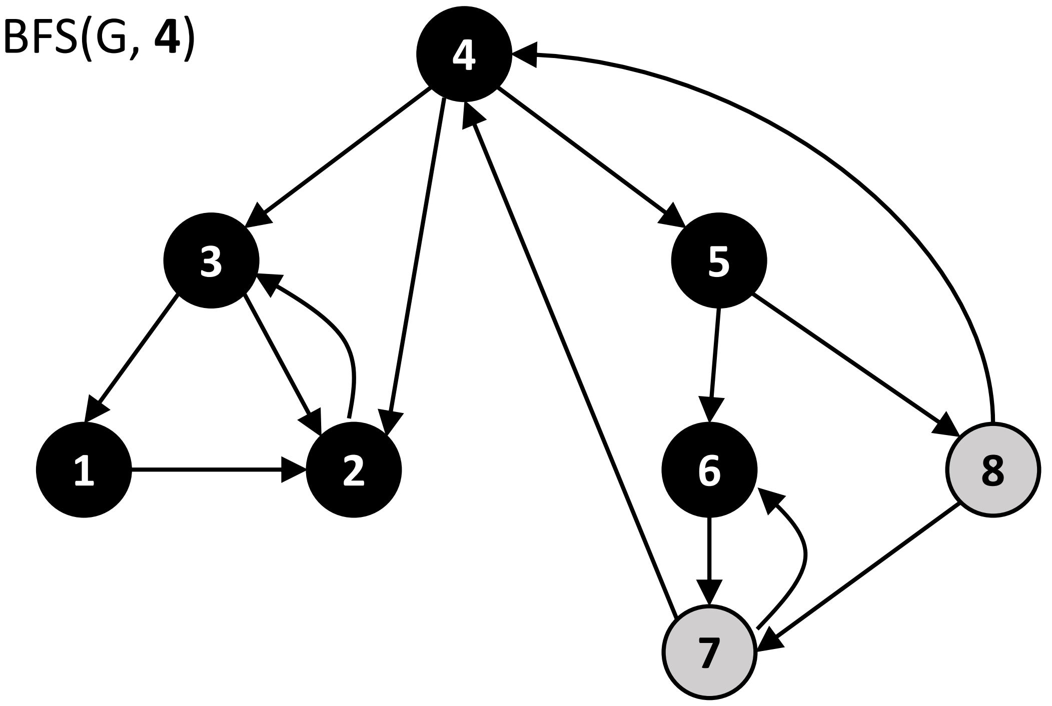
d	1	2
	5	1

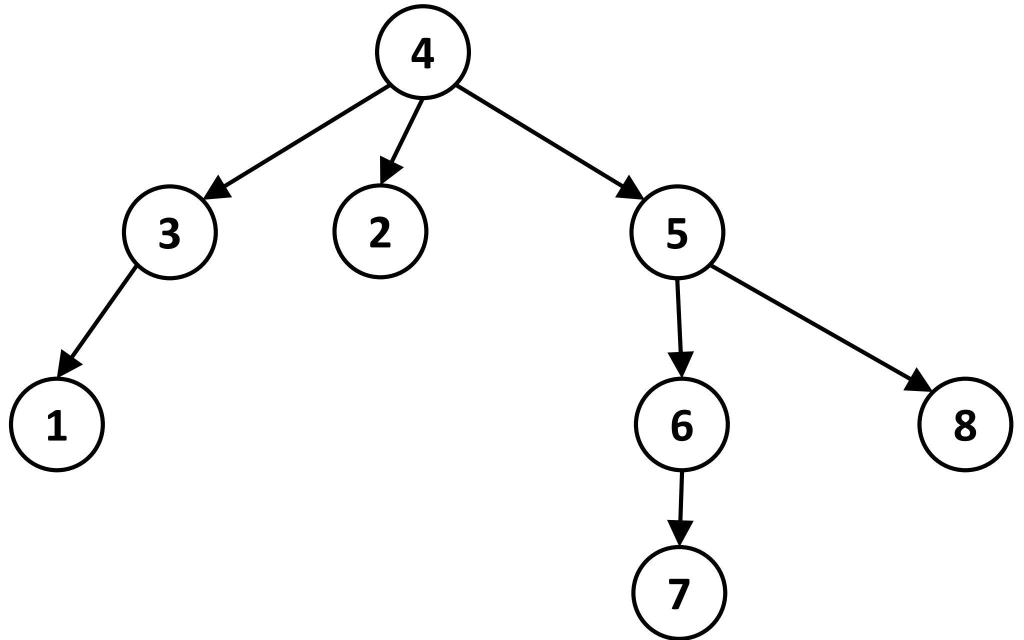
d	2	2	2
	1	6	8

d	2	2
	6	8

d	2	3
	8	7

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0
| 4 |

d = 1 1 1
| 3 | 2 | 5 |

1 1 2
| 2 | 5 | 1 |

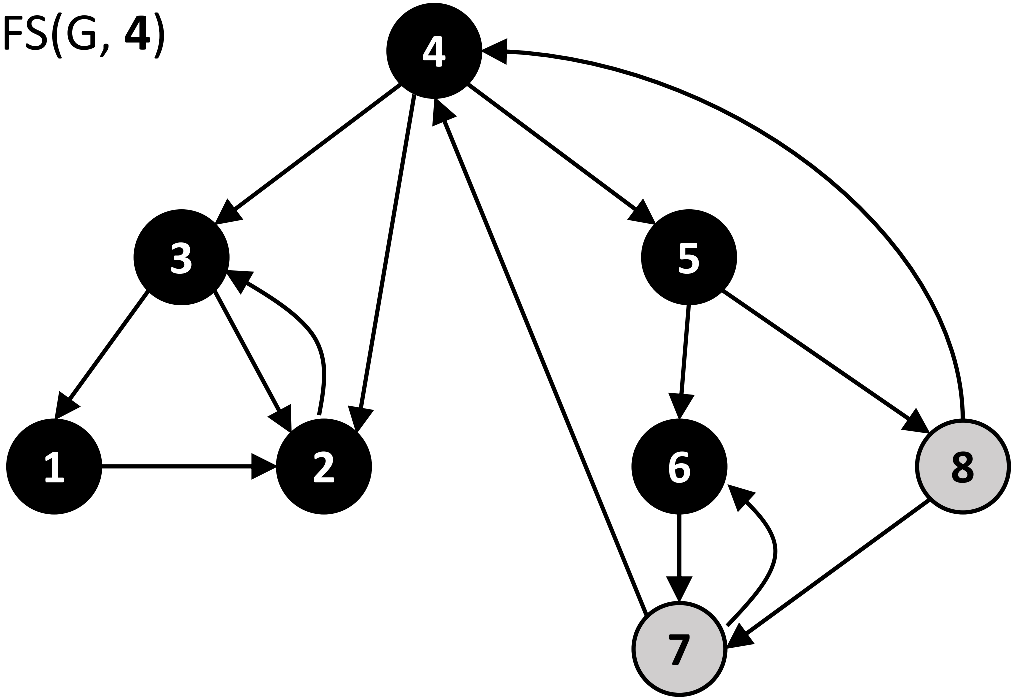
1 2
| 5 | 1 |

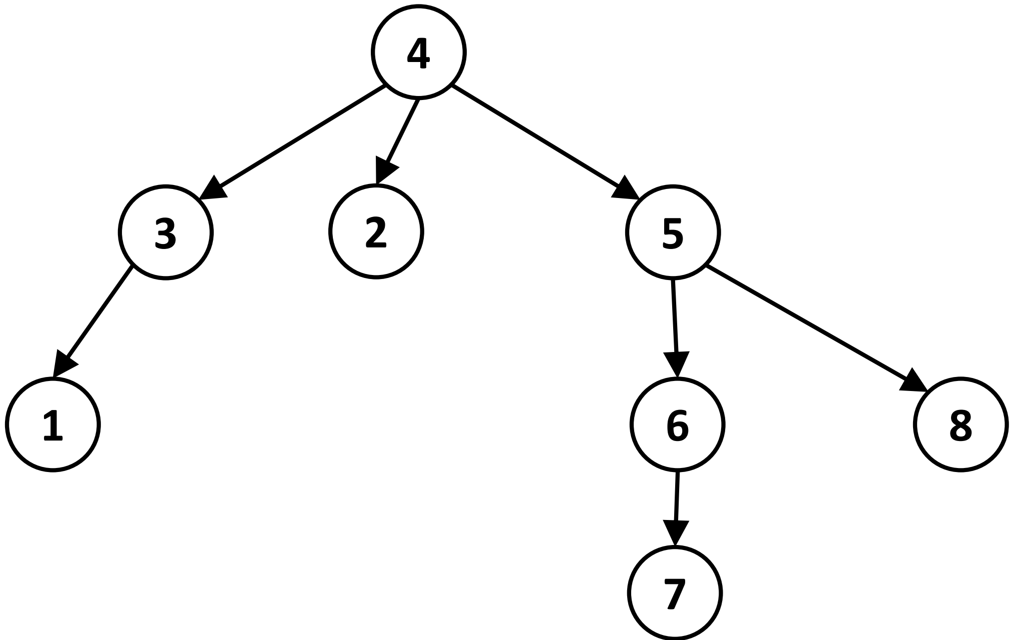
2 2 2
| 1 | 6 | 8 |

2 2
| 6 | 8 |

2 3
| 8 | 7 |

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d =			
	0		
	4		

d =			
	1	1	1
	3	2	5

1	1	2
2	5	1

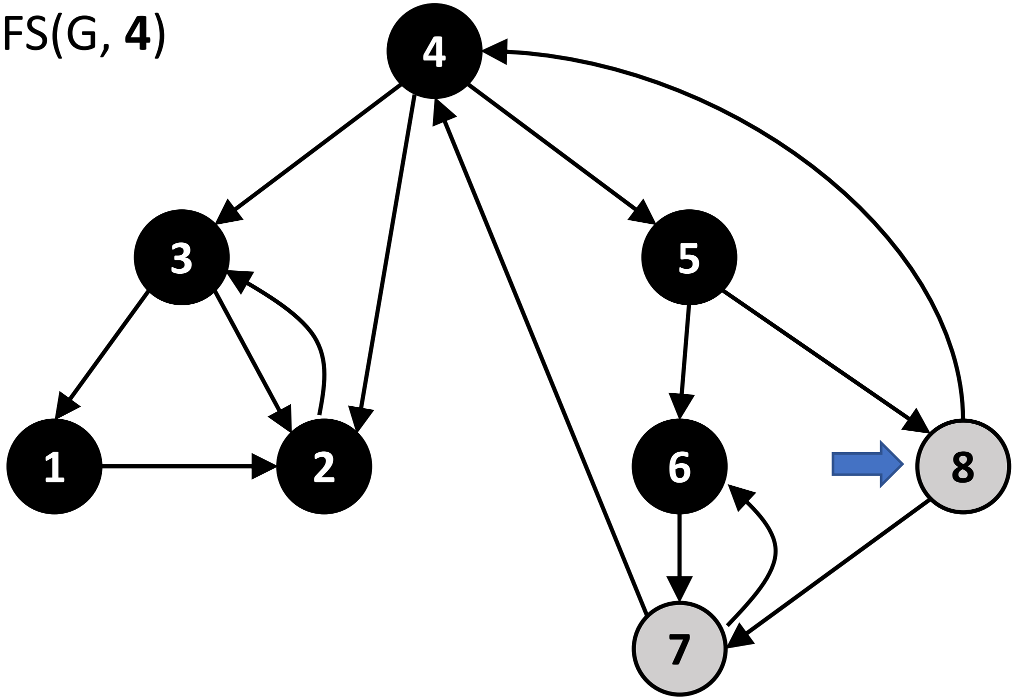
1	2
5	1

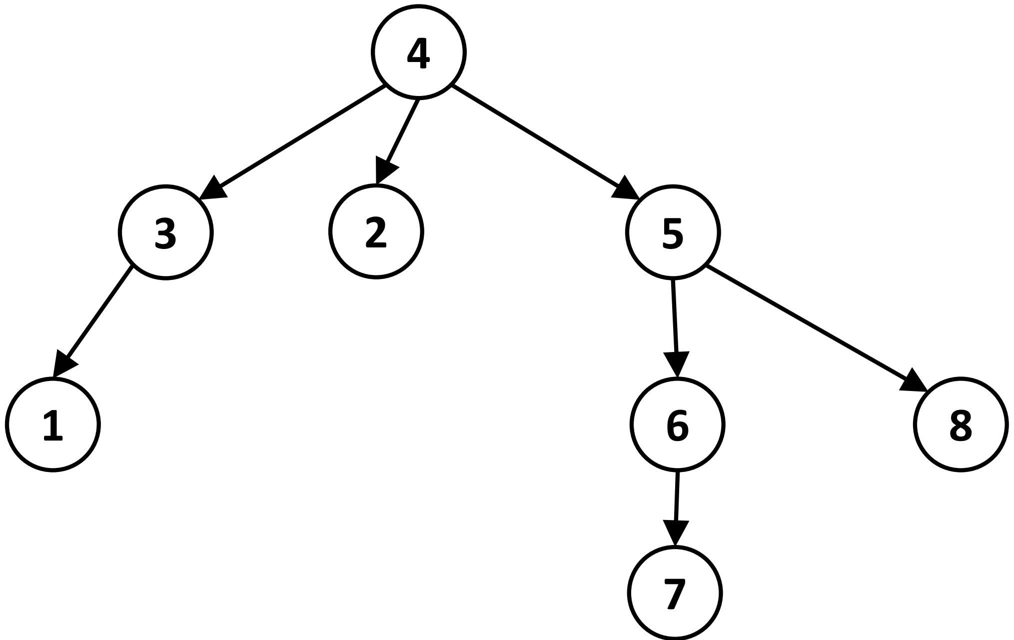
2	2	2
1	6	8

2	2
6	8

2	3
8	7

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
➡	8	→ 7 → 4

Contents of Q :

d =	0
	4

d =	1	1	1
	3	**2**	**5**

1	1	2
2	**5**	**1**

1	2
5	**1**

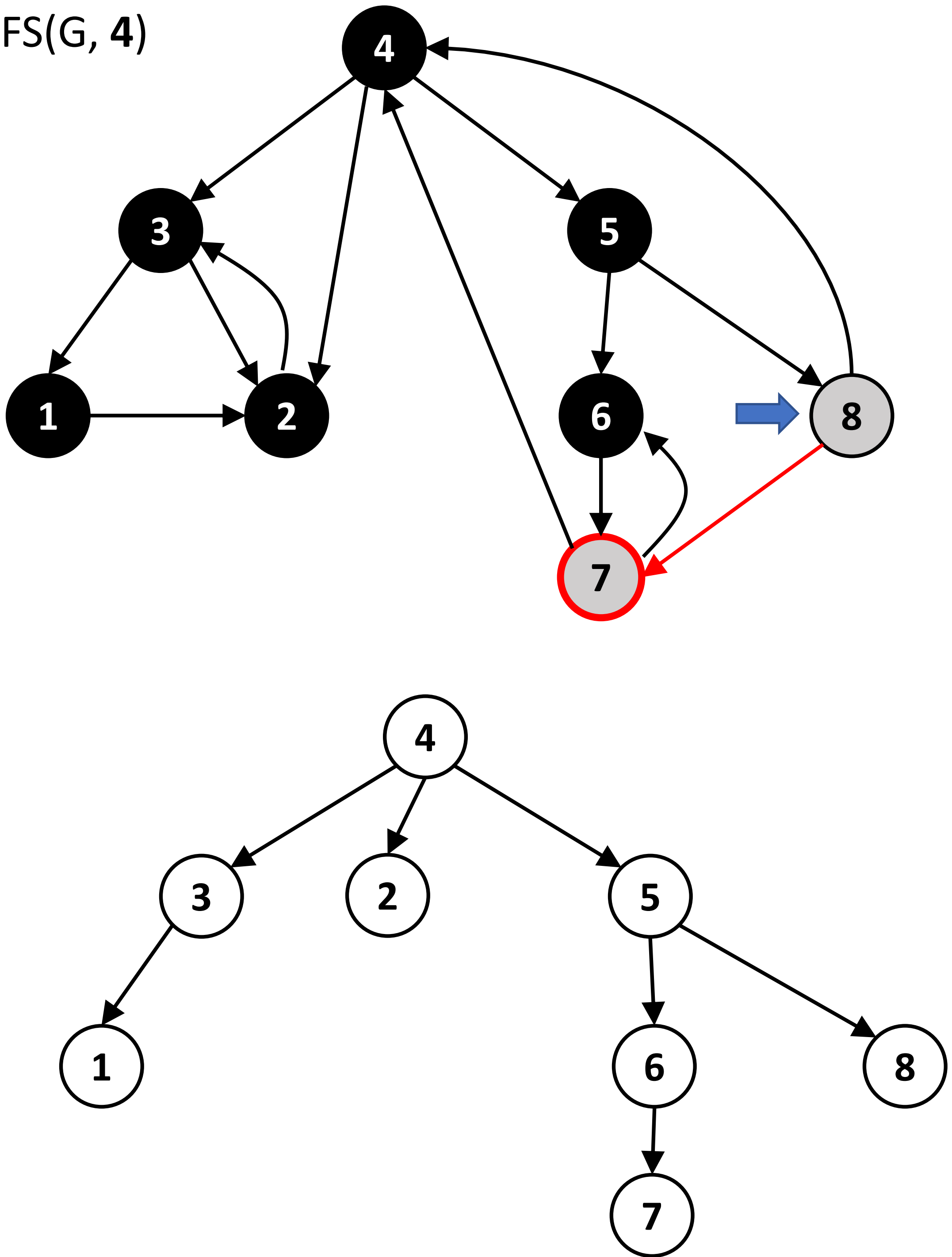
2	2	2
1	**6**	**8**

2	2
6	**8**

2	3
8	**7**

3
7

BFS(G, **4**)



Adj List of G :

✓ 1 → 2

✓ 2 → 3

✓ 3 → 1 → 2

✓ 4 → 3 → 2 → 5

✓ 5 → 6 → 8

✓ 6 → 7

7 → 6 → 4

8 → 7 → 4

Contents of Q :

d = 0
4

d = 1 1 1
3	2	5

1 1 2
2	5	1

1 2
5	1

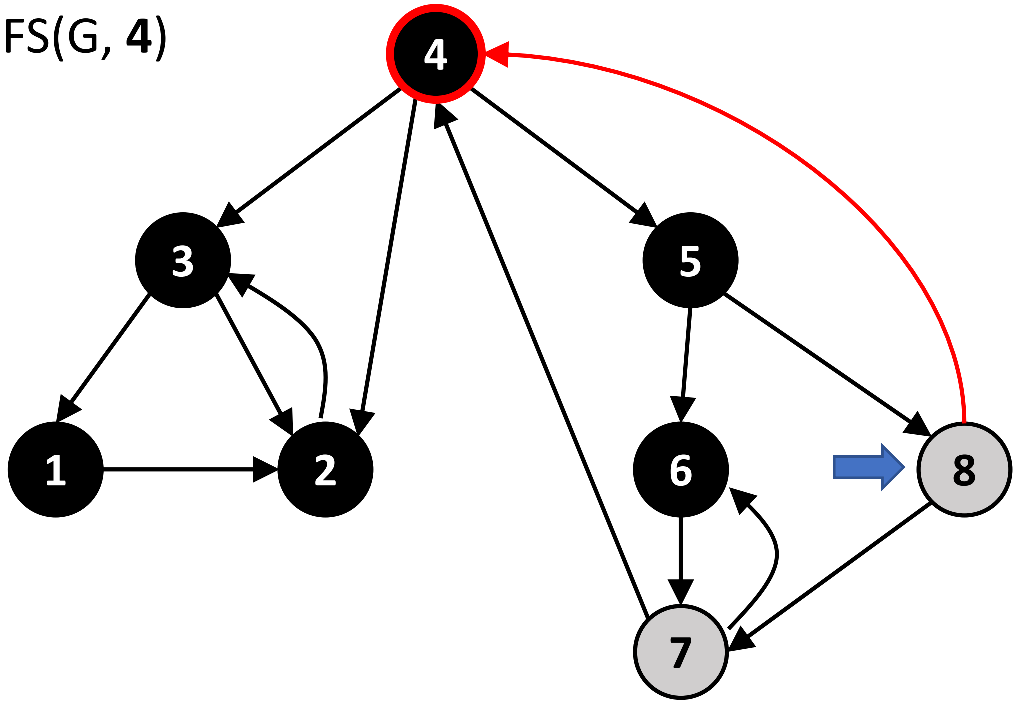
2 2 2
1	6	8

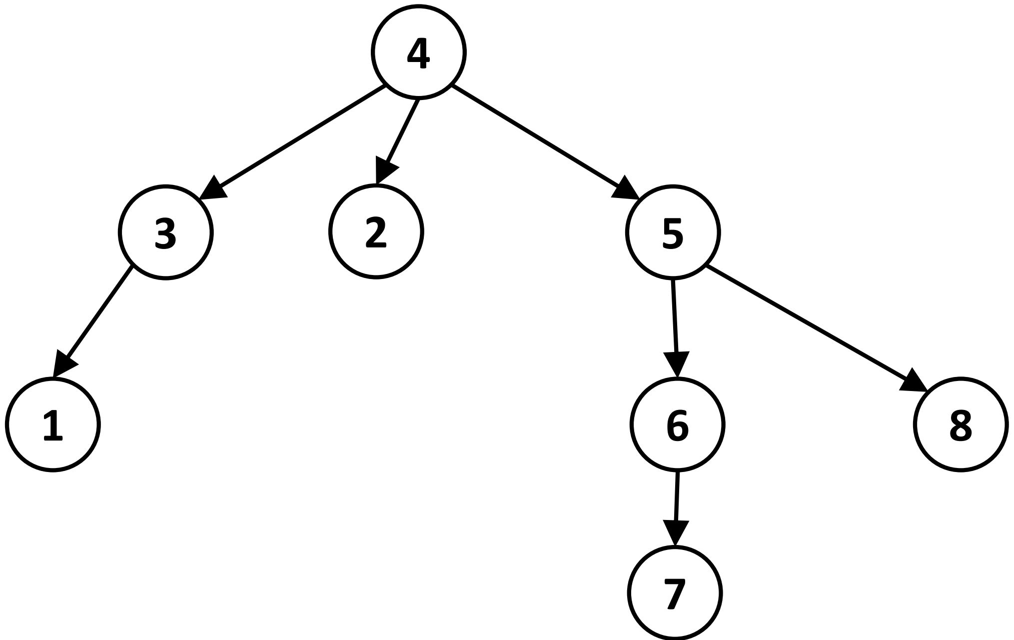
2 2
6	8

2 3
8	7

3
7

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

2 2 2

1	6	8

2 2

6	8

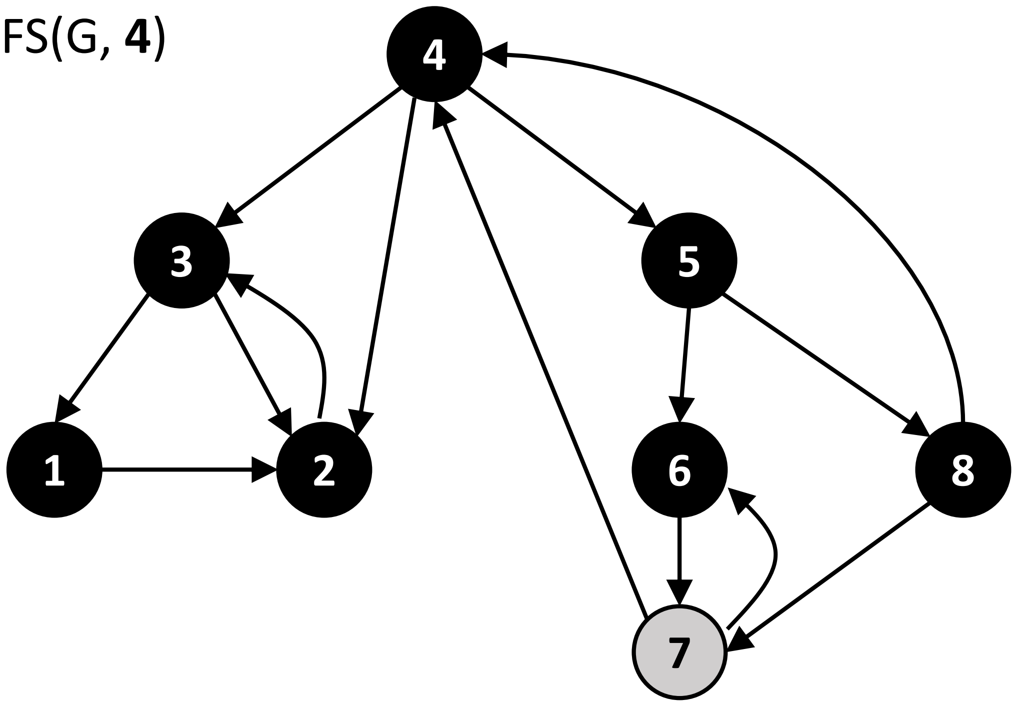
2 3

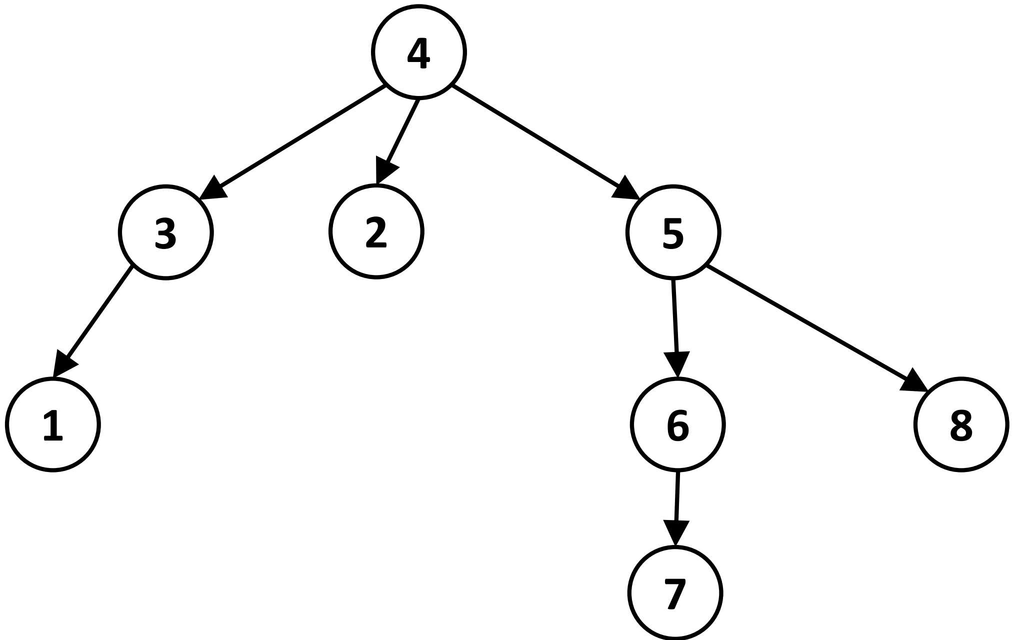
8	7

3

7

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
✓	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

2 2 2

1	6	8

2 2

6	8

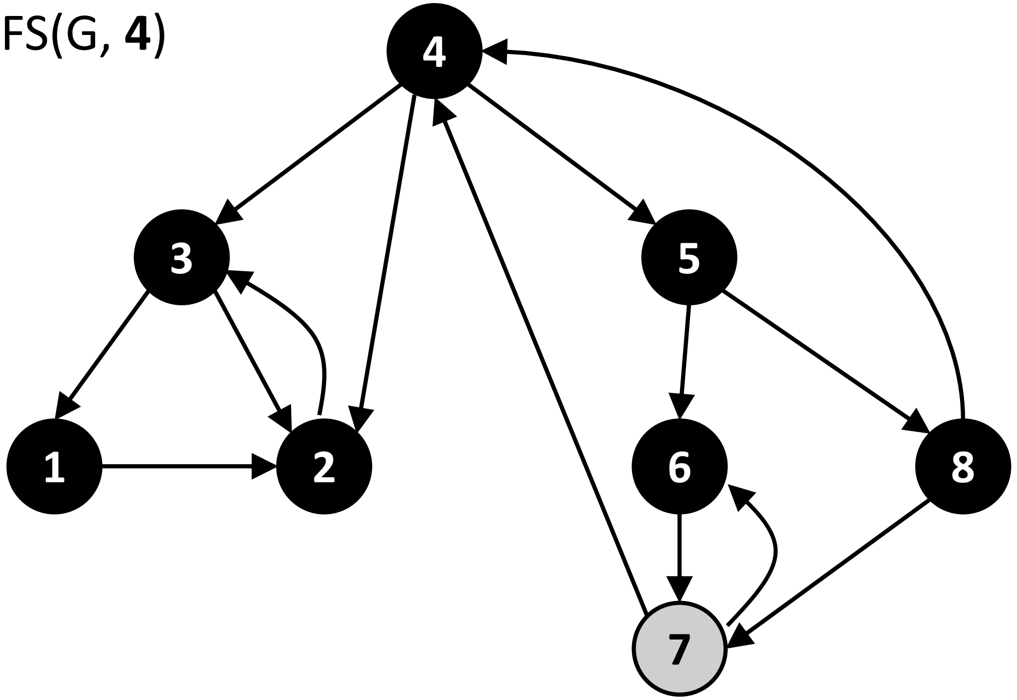
2 3

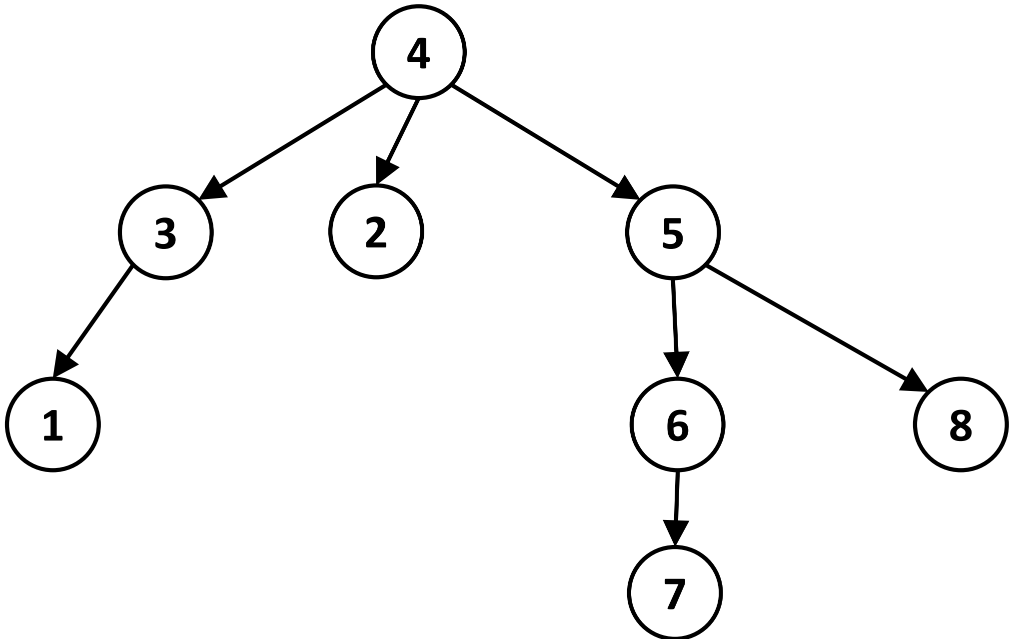
8	7

3

7

BFS(G, **4**)





Adj List of G :

✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
	7	→ 6 → 4
✓	8	→ 7 → 4

Contents of Q :

d = 0

4

d = 1 1 1

3	2	5

1 1 2

2	5	1

1 2

5	1

2 2 2

1	6	8

2 2

6	8

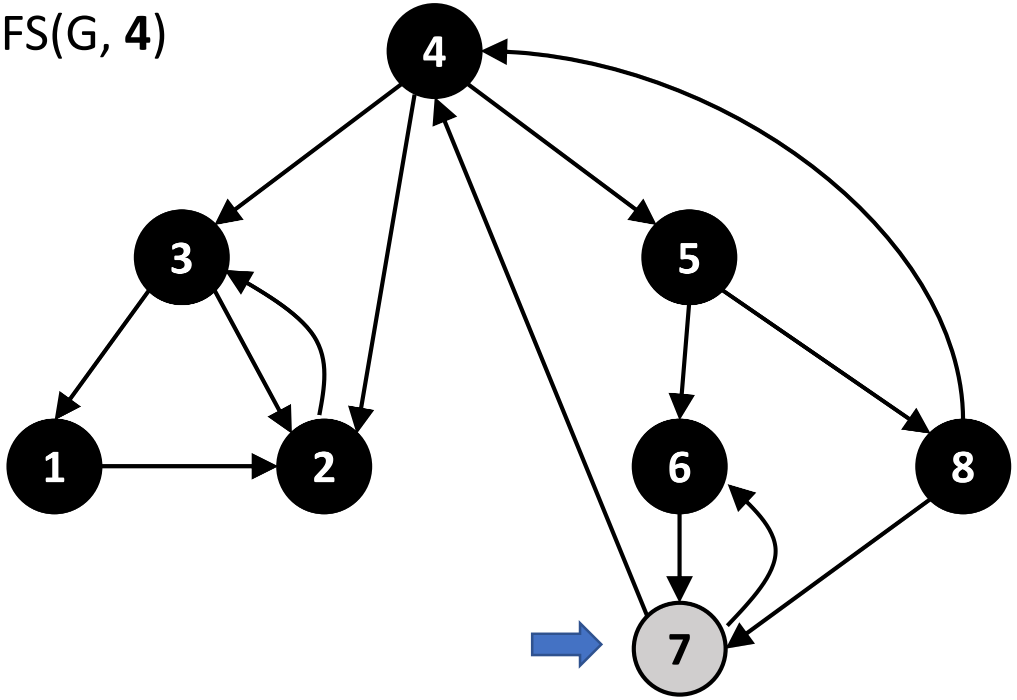
2 3

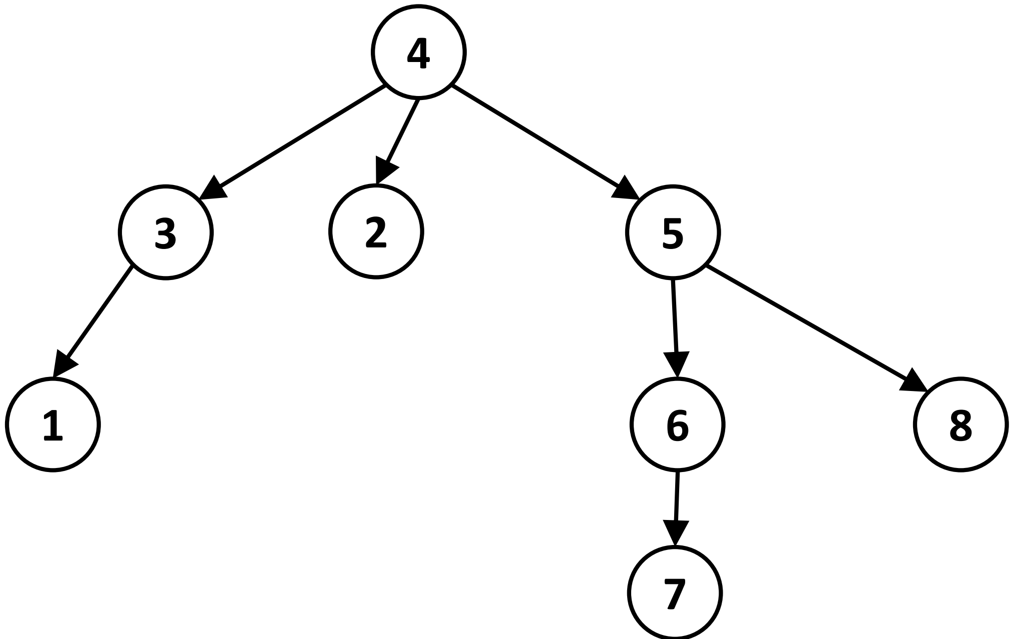
8	7

3

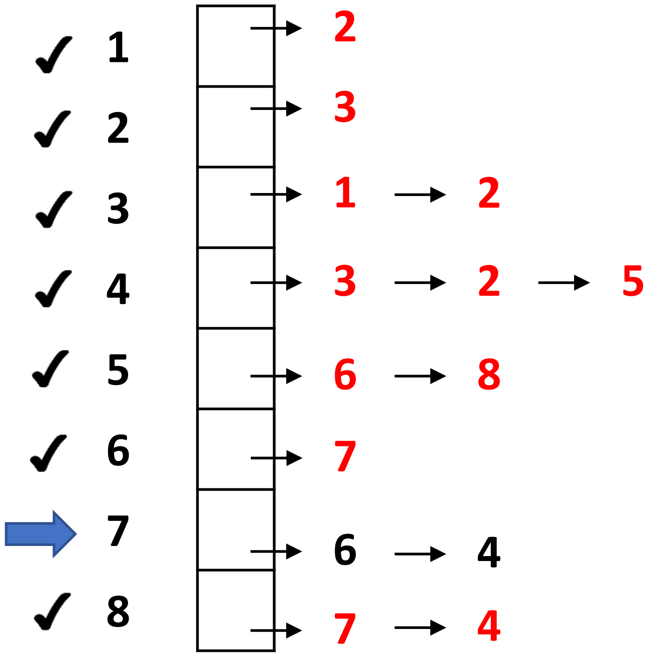
7

BFS(G, **4**)

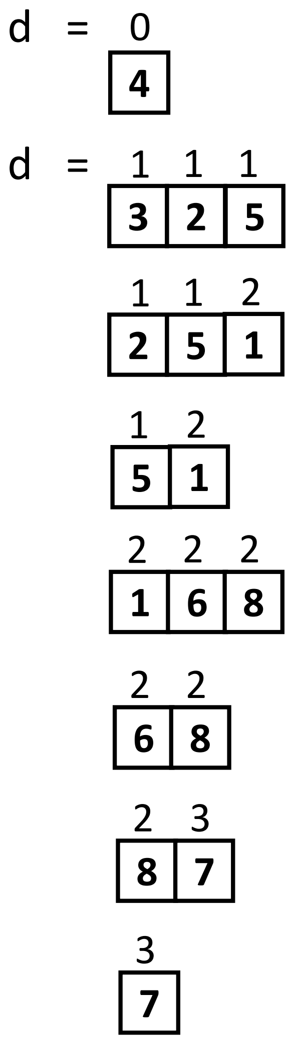




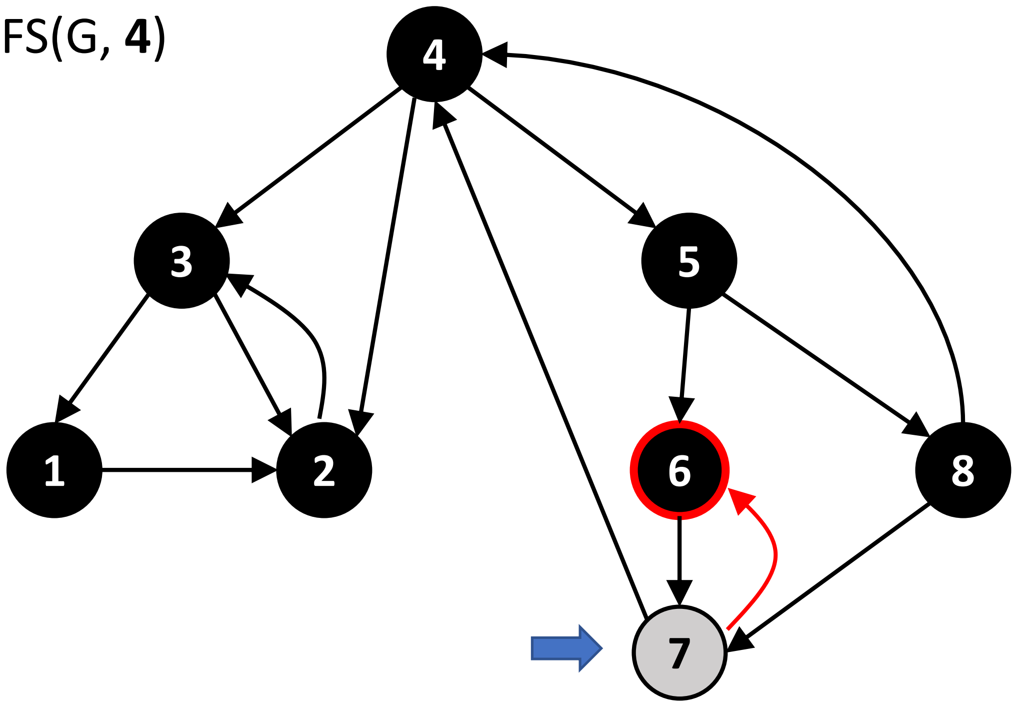
Adj List of G :

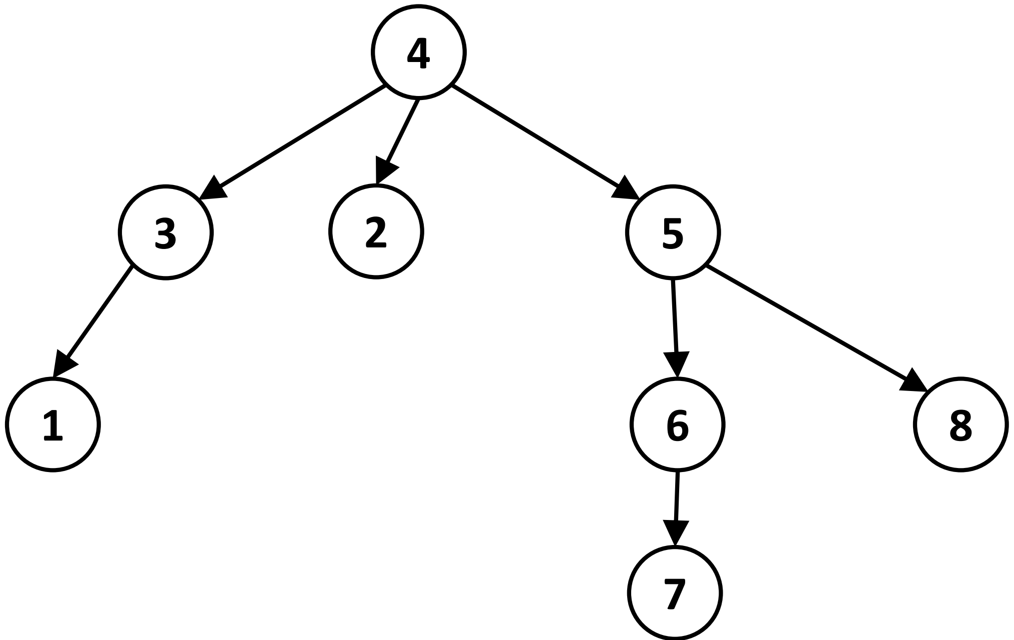


Contents of Q :



BFS(G, **4**)





Adj List of G :

✓ 1 → **2**
✓ 2 → **3**
✓ 3 → **1** → **2**
✓ 4 → **3** → **2** → **5**
✓ 5 → **6** → **8**
✓ 6 → **7**
➡ 7 → **6** → **4**
✓ 8 → **7** → **4**

Contents of Q :

d = 0

4

d = 1 1 1

3	**2**	**5**

1 1 2

2	**5**	**1**

1 2

5	**1**

2 2 2

1	**6**	**8**

2 2

6	**8**

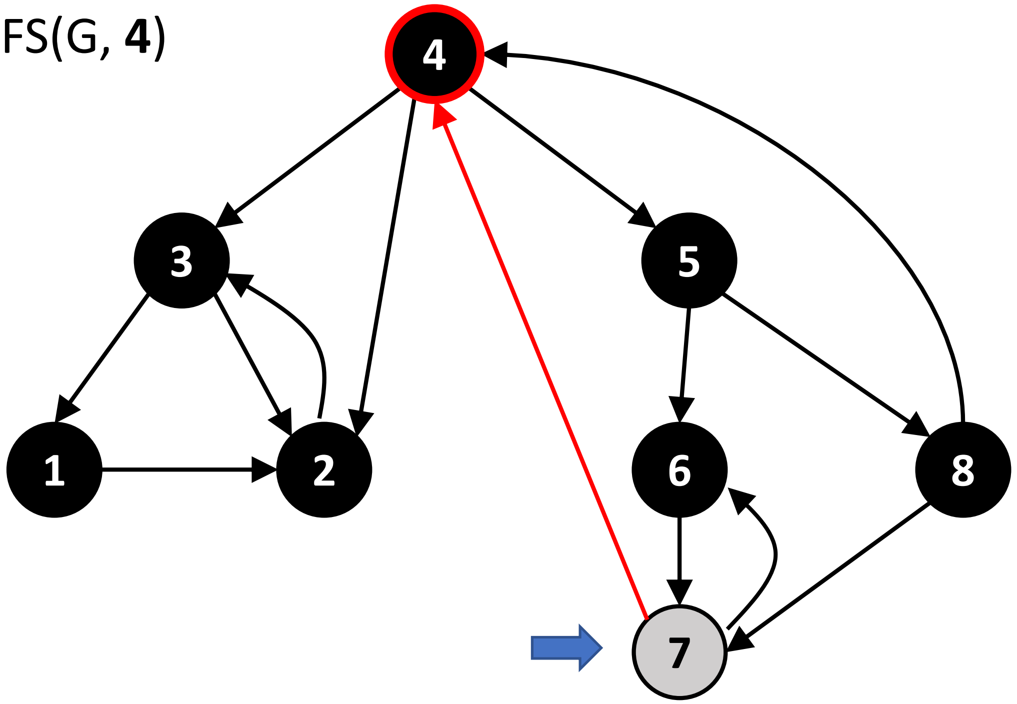
2 3

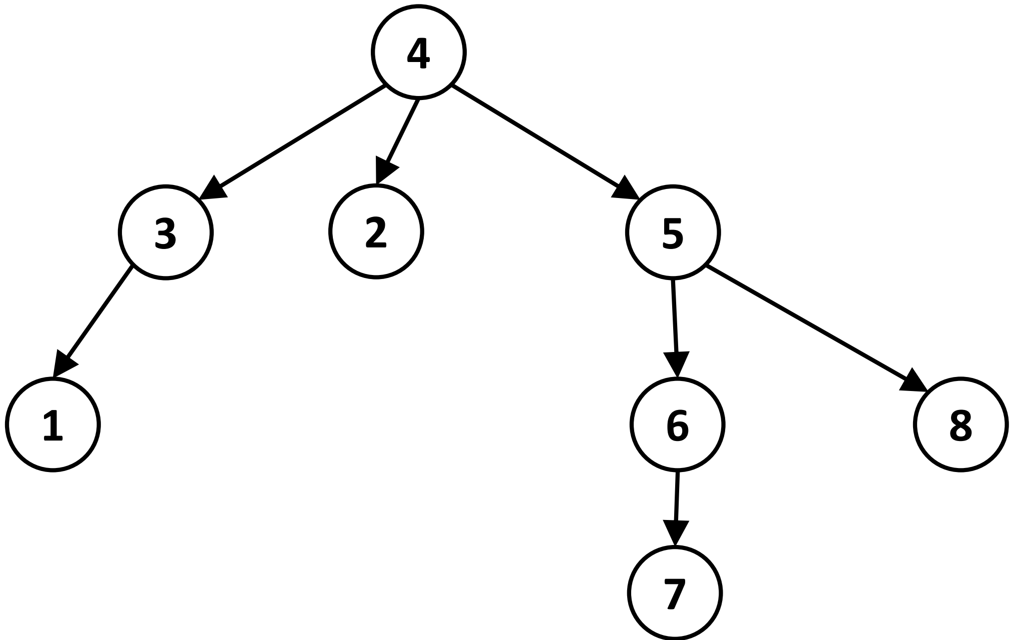
8	**7**

3

7

BFS(G, **4**)





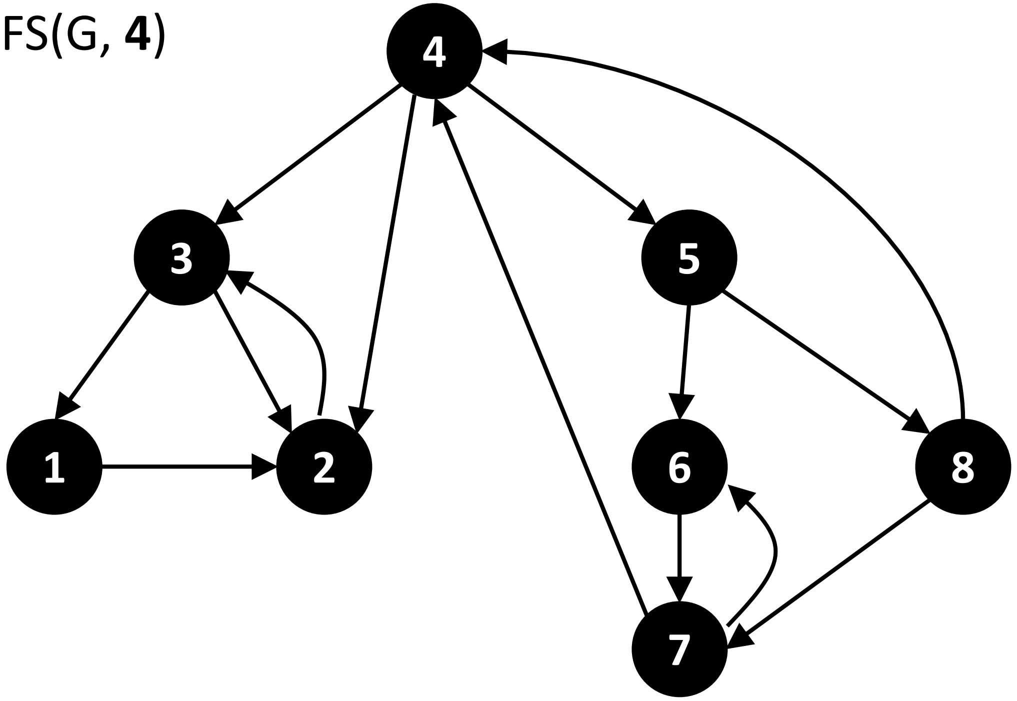
Adj List of G :

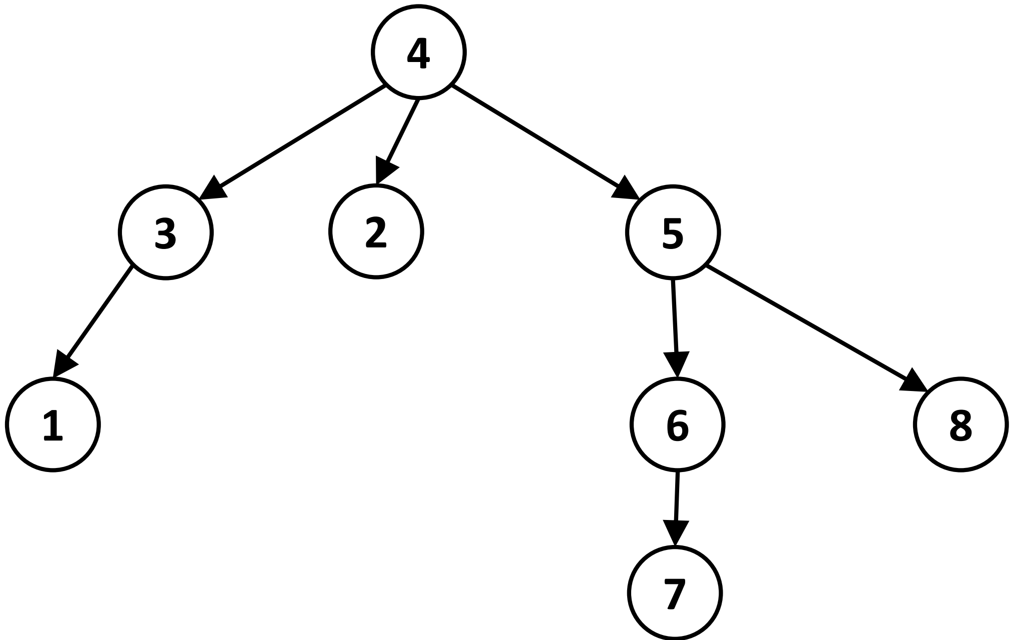
✓	1	→ 2
✓	2	→ 3
✓	3	→ 1 → 2
✓	4	→ 3 → 2 → 5
✓	5	→ 6 → 8
✓	6	→ 7
➡	7	→ 6 → 4
✓	8	→ 7 → 4

Contents of Q :

d	Q
0	4
1, 1, 1	3, 2, 5
1, 1, 2	2, 5, 1
1, 2	5, 1
2, 2, 2	1, 6, 8
2, 2	6, 8
2, 3	8, 7
3	7

BFS(G, **4**)





Adj List of G :

- ✓ 1 → **2**
- ✓ 2 → **3**
- ✓ 3 → **1** → **2**
- ✓ 4 → **3** → **2** → **5**
- ✓ 5 → **6** → **8**
- ✓ 6 → **7**
- ✓ 7 → **6** → **4**
- ✓ 8 → **7** → **4**

Contents of Q :

d =	0
	4

d =	1	1	1
	3	**2**	**5**

1	1	2
2	**5**	**1**

1	2
5	**1**

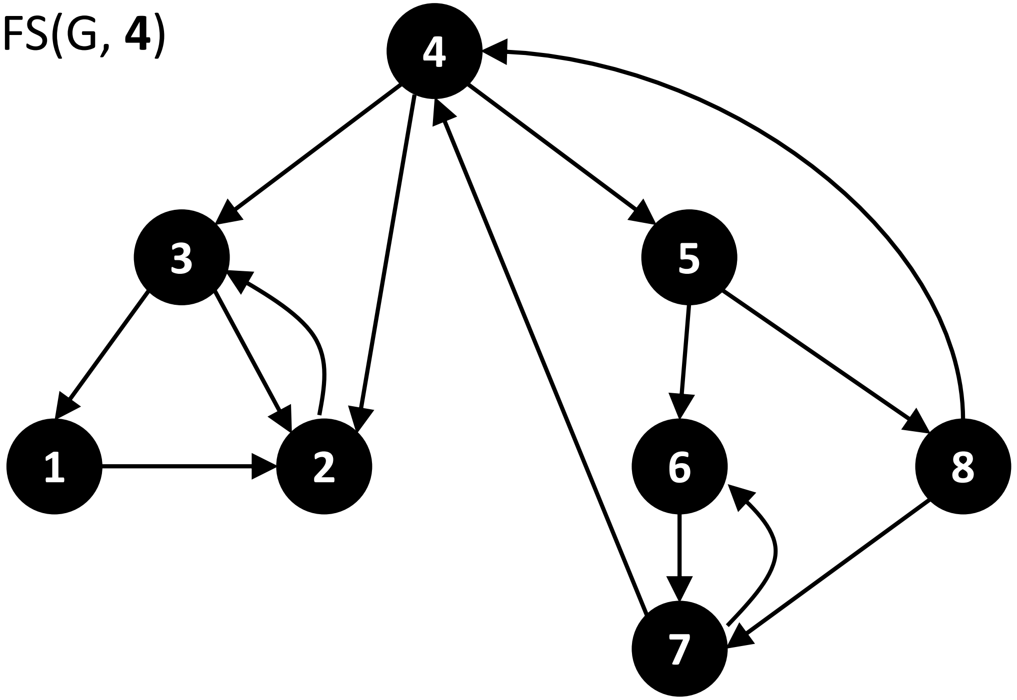
2	2	2
1	**6**	**8**

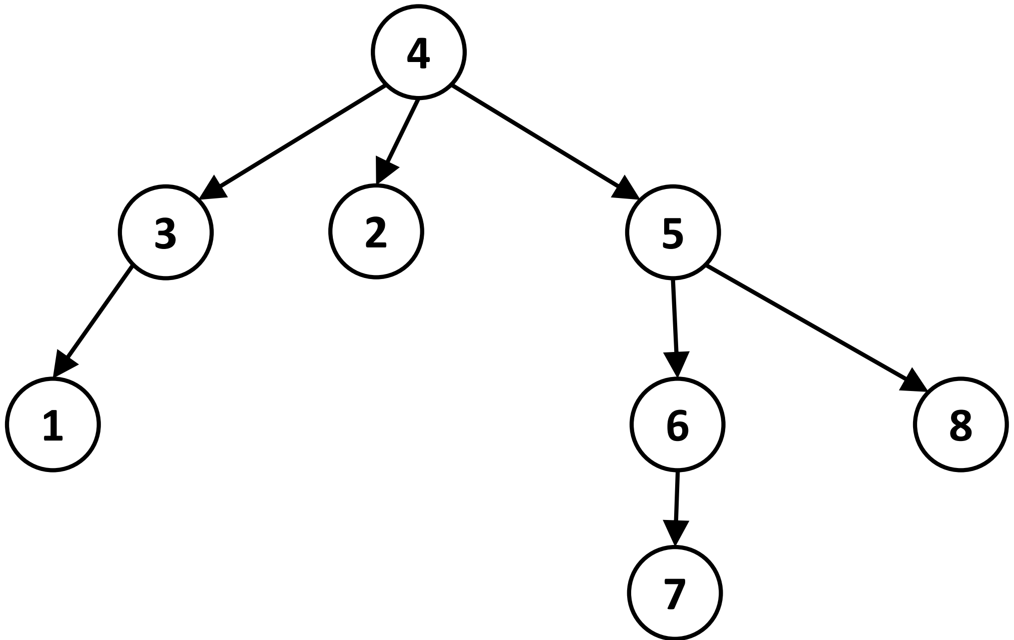
2	2
6	**8**

2	3
8	**7**

3
7

BFS(G, **4**)





Adj List of G :

✓ 1 → 2
✓ 2 → 3
✓ 3 → 1 → 2
✓ 4 → 3 → 2 → 5
✓ 5 → 6 → 8
✓ 6 → 7
✓ 7 → 6 → 4
✓ 8 → 7 → 4

Worst-Case Time Complexity of BFS:

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

Contents of Q :

d = 0
[4]

d = 1 1 1
[3 | 2 | 5]

1 1 2
[2 | 5 | 1]

1 2
[5 | 1]

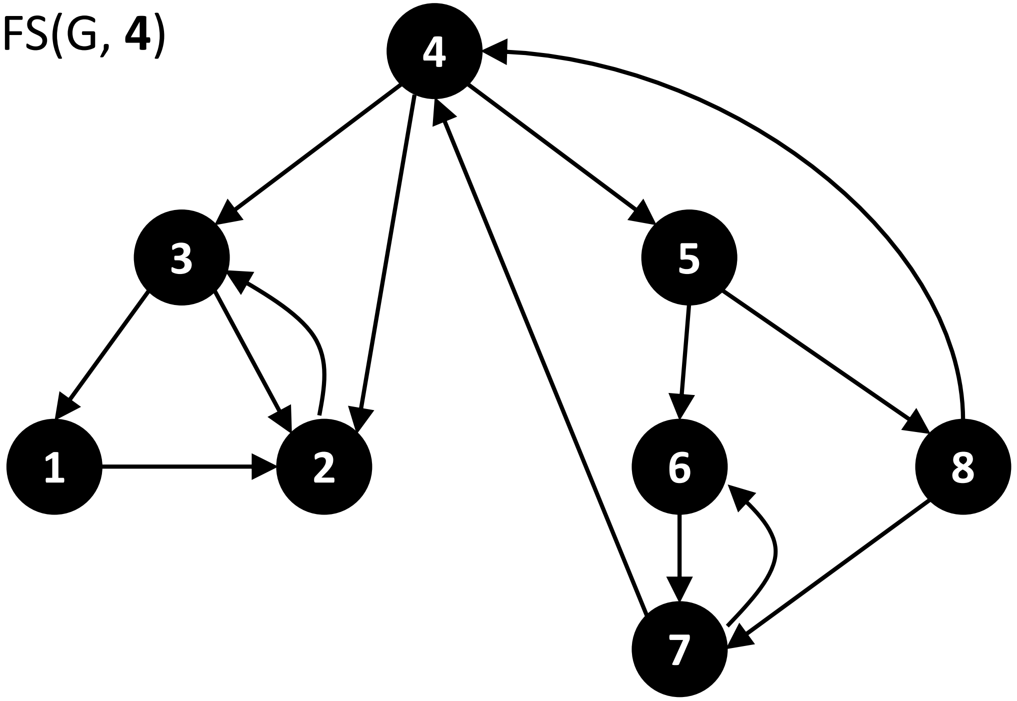
2 2 2
[1 | 6 | 8]

2 2
[6 | 8]

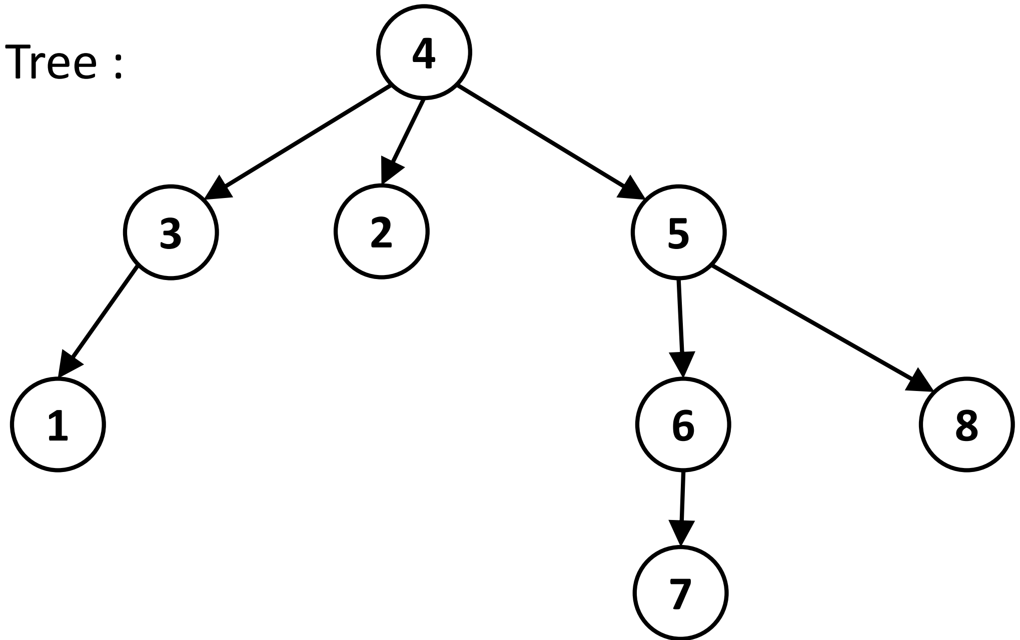
2 3
[8 | 7]

3
[7]

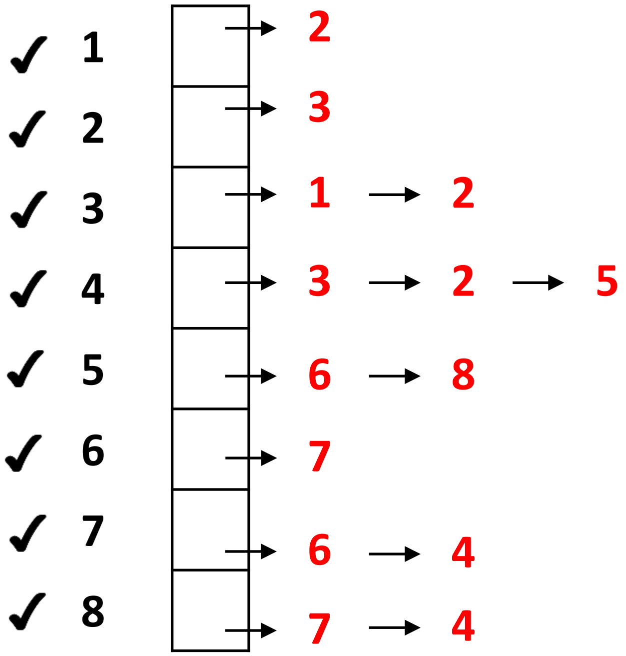
BFS(G, **4**)



BFS Tree :



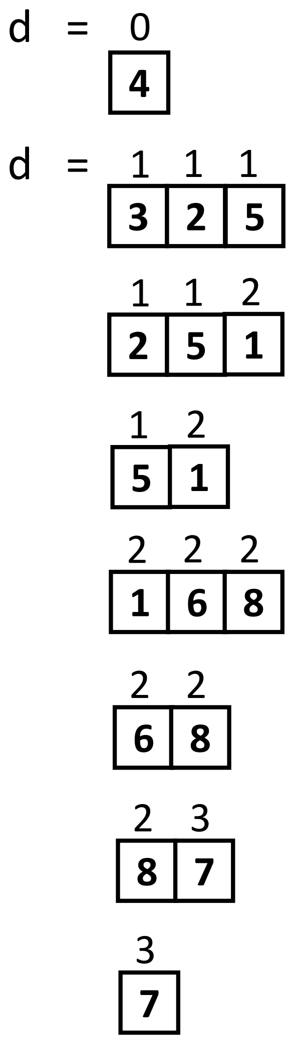
Adj List of G :



Worst-Case Time Complexity of BFS:

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

Contents of Q :



Breadth First Search

Proof of Correctness

v's discovery path from s : $s \rightarrow u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow u \rightarrow v$

Length of this path : d[v]

v's discovery path from s : $s \rightarrow u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow u \rightarrow v$

Length of this path : d[v]

v's shortest path from s : $s \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k \rightarrow v$

Length of shortest path : δ(s,v)
(Distance)

v's discovery path from s : $s \to u_1 \to u_2 \to \ldots \to u \to v$

Length of this path : d[v]

v's shortest path from s : $s \to v_1 \to v_2 \to \ldots \to v_k \to v$

Length of shortest path (Distance) : δ(s,v)

Lemma 0

After BFS(s), for every v ∈ V,

$$d[v] \geq \delta(s,v)$$

We would like to prove the following:

Main Theorem:

After BFS(s), for every v ∈ V,

$$d[v] = \delta(s,v)$$

We would like to prove the following:

Main Theorem:

After BFS(s), for every $v \in V$,

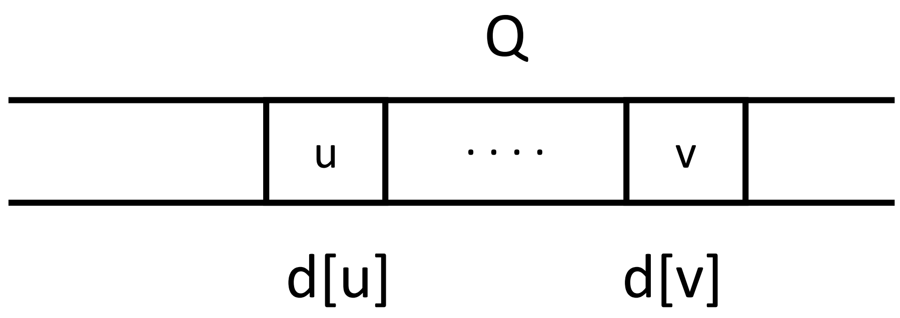
$$d[v] = \delta(s,v)$$

In other words, we would like to show that the discovery path is a shortest path to v

We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then:

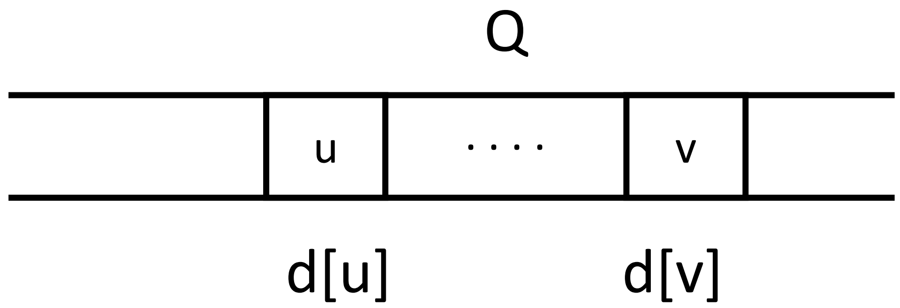


We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then

$$d[u] \leq d[v]$$



We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then

$$d[u] \leq d[v]$$

Proof of Lemma 1:

We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then

$$d[u] \leq d[v]$$

Proof of Lemma 1:

Suppose, for contradiction, that Lemma 1 is false.

We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then

$$d[u] \leq d[v]$$

Proof of Lemma 1:

Suppose, for contradiction, that Lemma 1 is false.

- Let v be the **first** node that enter Q such that d[u] > d[v] for some node u that entered Q before v.

We first prove the following:

Lemma 1:

If u enters Q before v enters Q during the the execution of BFS(s), then

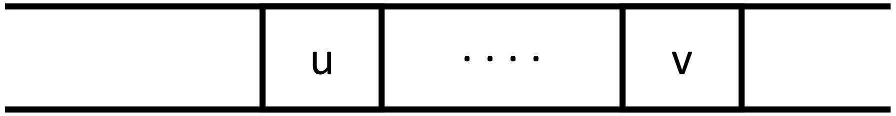
$$d[u] \leq d[v]$$

Proof of Lemma 1:

Suppose, for contradiction, that Lemma 1 is false.

- Let v be the **first** node that enter Q such that d[u] > d[v] for some node u that entered Q before v.

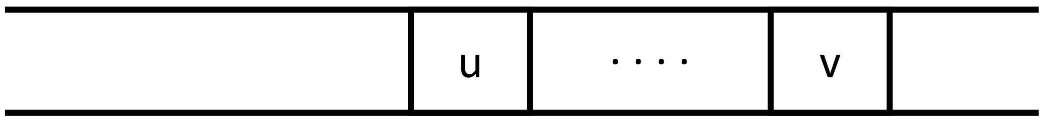
Q



d[u] > d[v]

Proof of Lemma 1 contd:

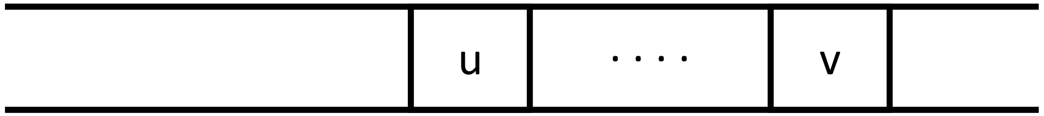
Q



d[u] > d[v]

Proof of Lemma 1 contd:

Q

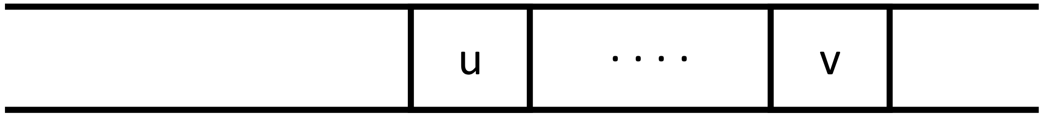


d[u] > d[v]

- v $\neq$ s because no vertex u enters Q before s

Proof of Lemma 1 contd:

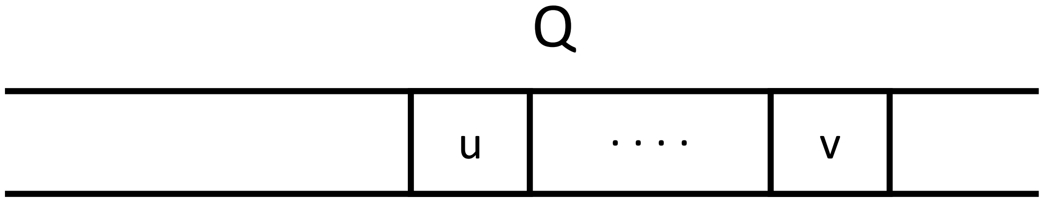
Q



d[u] > d[v]

- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

Proof of Lemma 1 contd:

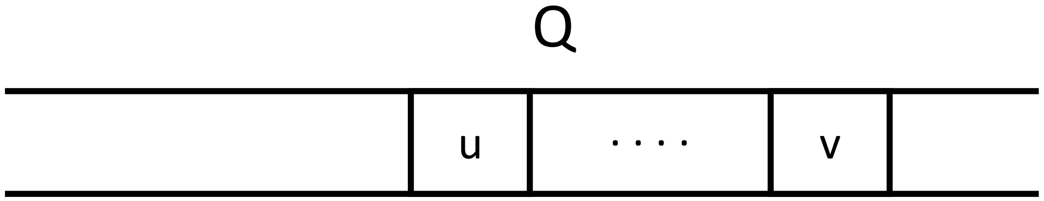


$d[u] > d[v]$

- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

$\Rightarrow$ u and v entered Q during the exploration of some nodes, say u' and v' respectively

Proof of Lemma 1 contd:



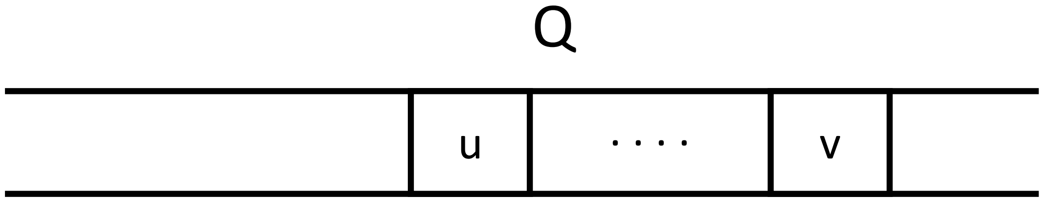
d[u] > d[v]

- v ≠ s because no vertex u enters Q before s
- u ≠ s because d[s] = 0 and d[v] ≥ 0

⇒ u and v entered Q during the exploration of some nodes, say u' and v' respectively

⇒ d[u] = d[u'] + 1 and d[v] = d[v'] + 1

Proof of Lemma 1 contd:



$d[u] > d[v]$

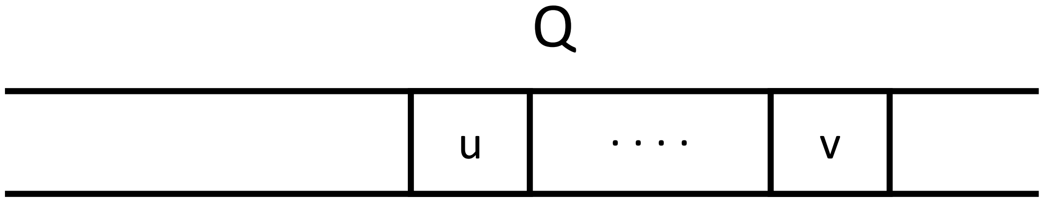
- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u' and v' respectively

  $\Rightarrow d[u] = d[u'] + 1$ and $d[v] = d[v'] + 1$

- Since $d[u] \neq d[v]$, $d[u'] \neq d[v']$

Proof of Lemma 1 contd:



$d[u] > d[v]$

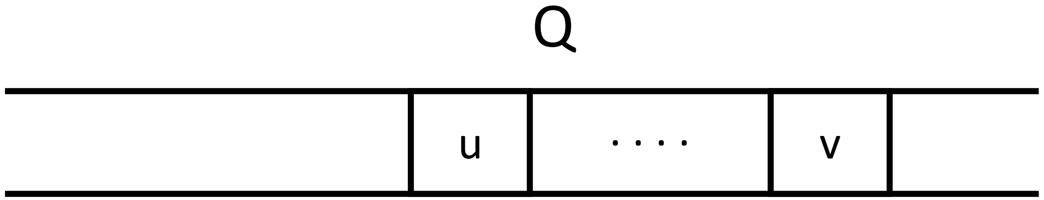
- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u' and v' respectively

  $\Rightarrow d[u] = d[u'] + 1$ and $d[v] = d[v'] + 1$

- Since $d[u] \neq d[v]$, $d[u'] \neq d[v']$ $\Rightarrow$ $u' \neq v'$ i.e. u' and v' are distinct nodes

Proof of Lemma 1 contd:



$d[u] > d[v]$

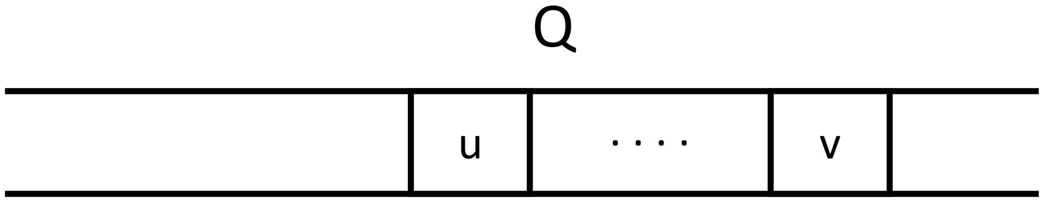
- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u' and v' respectively

  $\Rightarrow$ $d[u] = d[u'] + 1$ and $d[v] = d[v'] + 1$

- Since $d[u] \neq d[v]$, $d[u'] \neq d[v']$ $\Rightarrow$ $u' \neq v'$ i.e. u' and v' are distinct nodes
- u was discovered before v $\Rightarrow$ u' was discovered before v'

Proof of Lemma 1 contd:



$d[u] > d[v]$

- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

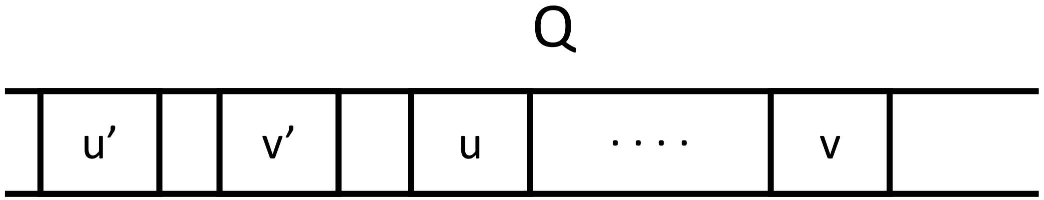
  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u’ and v’ respectively

  $\Rightarrow d[u] = d[u'] + 1$ and $d[v] = d[v'] + 1$

- Since $d[u] \neq d[v]$, $d[u'] \neq d[v']$ $\Rightarrow$ $u' \neq v'$ i.e. u’ and v’ are distinct nodes
- u was discovered before v $\Rightarrow$ u’ was discovered before v’

  $\Rightarrow$ u’ entered Q before v’

Proof of Lemma 1 contd:



d[u] > d[v]

- v ≠ s because no vertex u enters Q before s
- u ≠ s because d[s] = 0 and d[v] ≥ 0

⇒ u and v entered Q during the exploration of some nodes, say u’ and v’ respectively

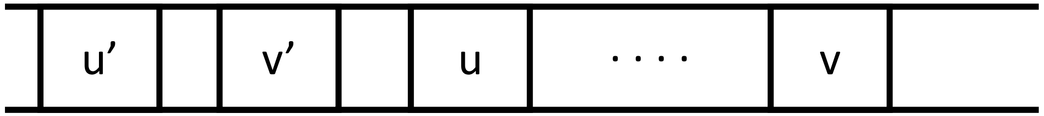
⇒ d[u] = d[u’] + 1 and d[v] = d[v’] + 1

- Since d[u] ≠ d[v], d[u’] ≠ d[v’] ⇒ u’ ≠ v’ i.e. u’ and v’ are distinct nodes
- u was discovered before v ⇒ u’ was discovered before v’

⇒ u’ entered Q before v’

Proof of Lemma 1 contd:

Q



d[u] > d[v]

- v ≠ s because no vertex u enters Q before s
- u ≠ s because d[s] = 0 and d[v] ≥ 0

  ⇒ u and v entered Q during the exploration of some nodes, say u’ and v’ respectively

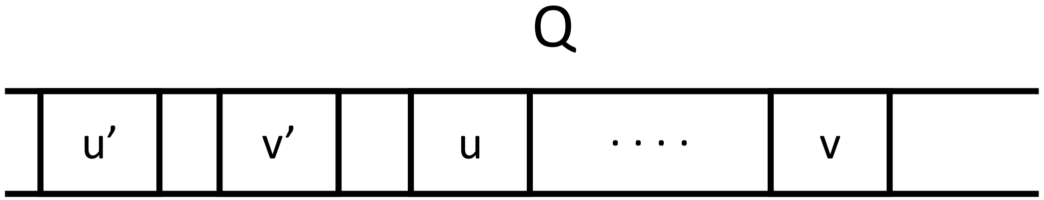
  ⇒ d[u] = d[u’] + 1 and d[v] = d[v’] + 1

- Since d[u] ≠ d[v], d[u’] ≠ d[v’] ⇒ u’ ≠ v’ i.e. u’ and v’ are distinct nodes
- u was discovered before v ⇒ u’ was discovered before v’

  ⇒ u’ entered Q before v’

  ⇒ d[u’] ≤ d[v’] By definition of v

Proof of Lemma 1 contd:



$d[u] > d[v]$

- $v \neq s$ because no vertex u enters Q before s
- $u \neq s$ because $d[s] = 0$ and $d[v] \geq 0$

  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u’ and v’ respectively

  $\Rightarrow$ $d[u] = d[u'] + 1$ and $d[v] = d[v'] + 1$

- Since $d[u] \neq d[v]$, $d[u'] \neq d[v']$ $\Rightarrow$ $u' \neq v'$ i.e. u’ and v’ are distinct nodes
- u was discovered before v $\Rightarrow$ u’ was discovered before v’

  $\Rightarrow$ u’ entered Q before v’

  $\Rightarrow$ $d[u'] \leq d[v']$ By definition of v

  $\Rightarrow$ $d[u] \leq d[v]$

Proof of Lemma 1 contd:

Q

u’		v’		u	. . . .	v	

d[u] > d[v]

- v $\neq$ s because no vertex u enters Q before s
- u $\neq$ s because d[s] = 0 and d[v] $\geq$ 0

  $\Rightarrow$ u and v entered Q during the exploration of some nodes, say u’ and v’ respectively

  $\Rightarrow$ d[u] = d[u’] + 1 and d[v] = d[v’] + 1

- Since d[u] $\neq$ d[v], d[u’] $\neq$ d[v’] $\Rightarrow$ u’ $\neq$ v’ i.e. u’ and v’ are distinct nodes
- u was discovered before v $\Rightarrow$ u’ was discovered before v’

  $\Rightarrow$ u’ entered Q before v’

  $\Rightarrow$ d[u’] $\leq$ d[v’] By definition of v

  $\Rightarrow$ d[u] $\leq$ d[v] Contradiction !

Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Main Theorem:

After BFS(s), for every v ∈ V, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Main Theorem:

After BFS(s), for every v ∈ V, d[v] = δ(s,v)

Proof of Main Theorem:

Suppose, for contradiction, that there exists x ∈ V such that d[x] $\neq$ δ(s,x) .

Main Theorem:

After BFS(s), for every v ∈ V, d[v] = δ(s,v)

Proof of Main Theorem:

Suppose, for contradiction, that there exists x ∈ V such that d[x] $\neq$ δ(s,x) . Clearly x $\neq$ s.

Main Theorem:

After BFS(s), for every v ∈ V, d[v] = δ(s,v)

Proof of Main Theorem:

Suppose, for contradiction, that there exists x ∈ V such that d[x] $\neq$ δ(s,x) . Clearly x $\neq$ s.

- Let v be the **closest** node from s such that d[v] $\neq$ δ(s,v)

Main Theorem:

After BFS(s), for every v ∈ V, d[v] = δ(s,v)

Proof of Main Theorem:

Suppose, for contradiction, that there exists x ∈ V such that d[x] $\neq$ δ(s,x) . Clearly x $\neq$ s.

- Let v be the **closest** node from s such that d[v] $\neq$ δ(s,v)
- By Lemma 0, d[v] > δ(s,v)

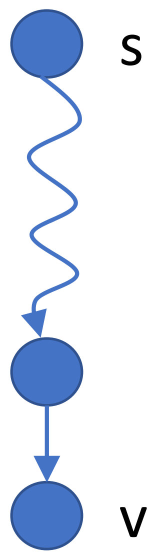
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G



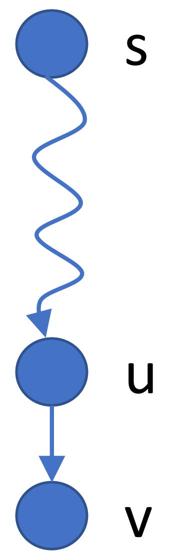
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
    - Let (u,v) be the last edge on that path



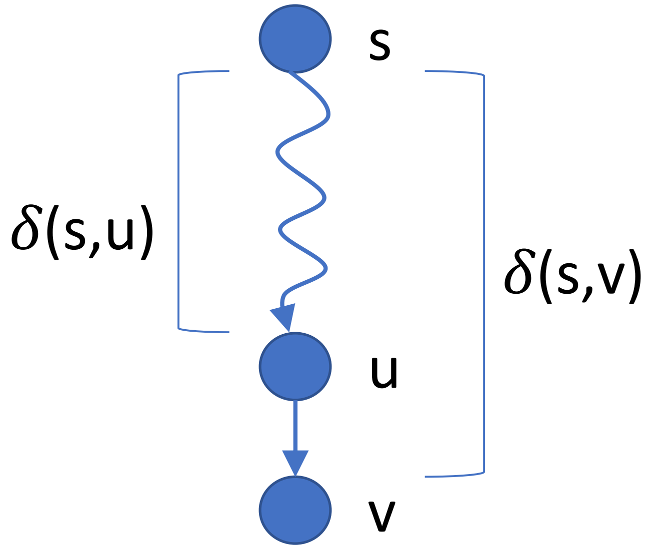
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path



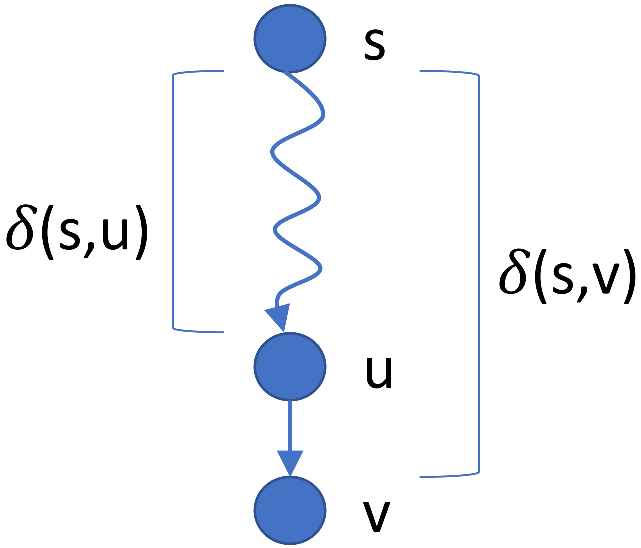
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$



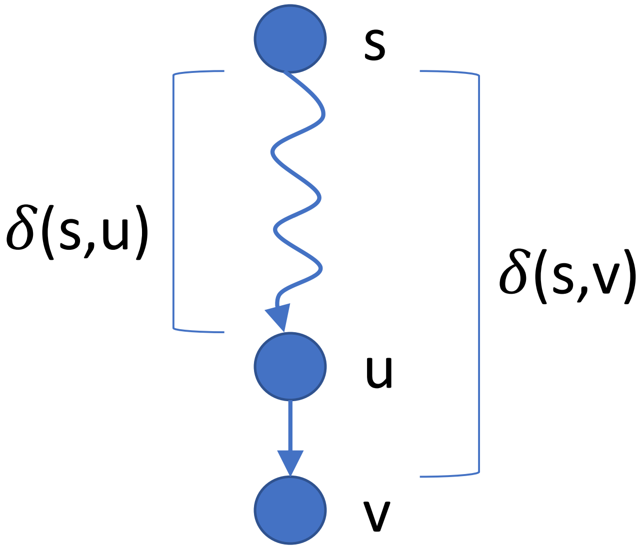
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
    - Let (u,v) be the last edge on that path
    - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)



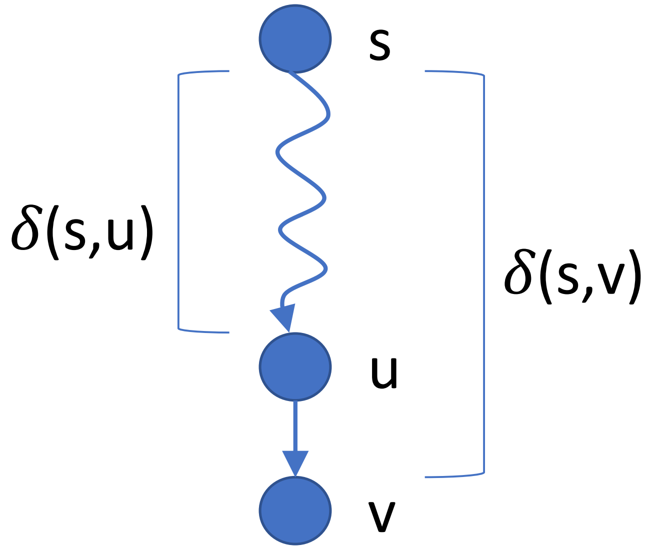
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$



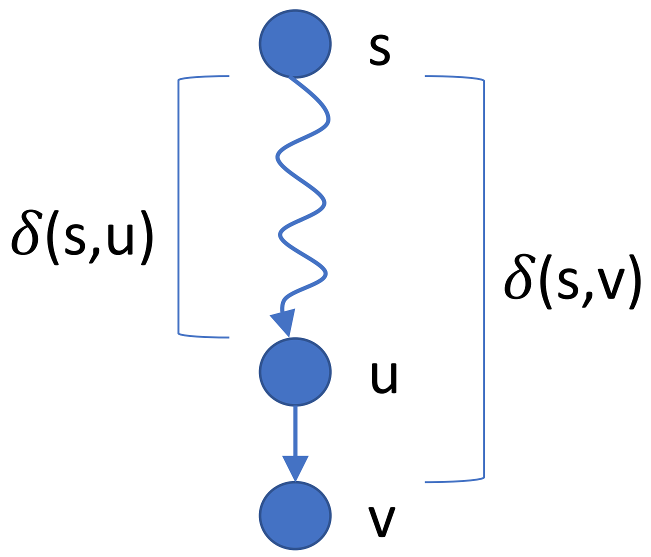
Main Theorem:

After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
    - Let (u,v) be the last edge on that path
    - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$



Main Theorem:

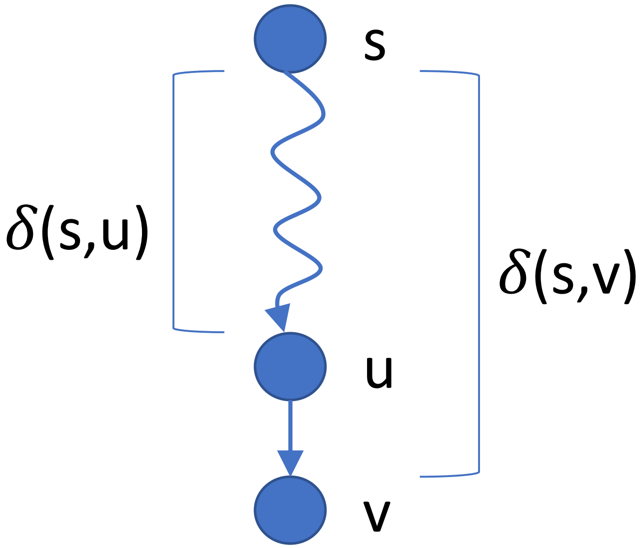
After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$

$$= \delta(s,u) + 1$$



Main Theorem:

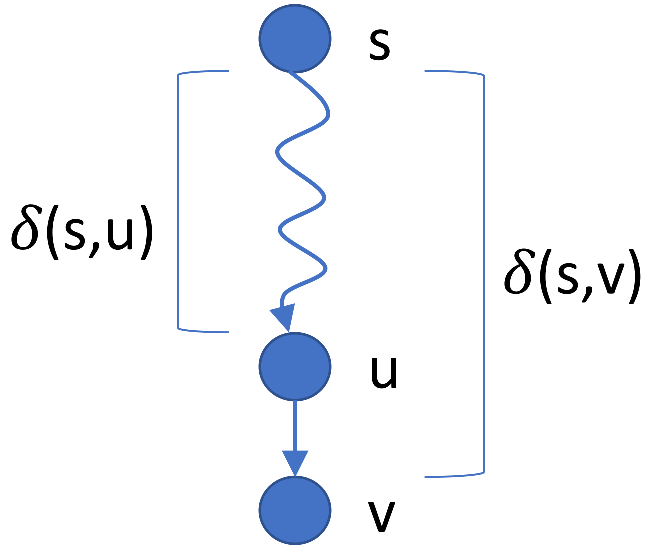
After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$

  $= \delta(s,u) + 1$



Main Theorem:

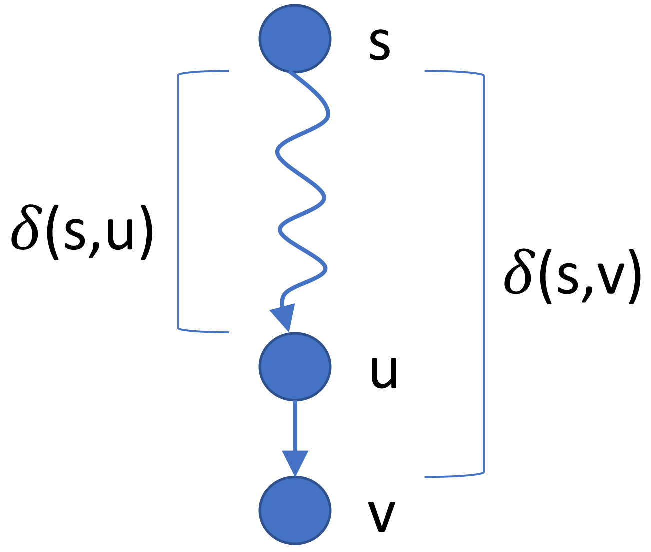
After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$

$= d[u] + 1$



Main Theorem:

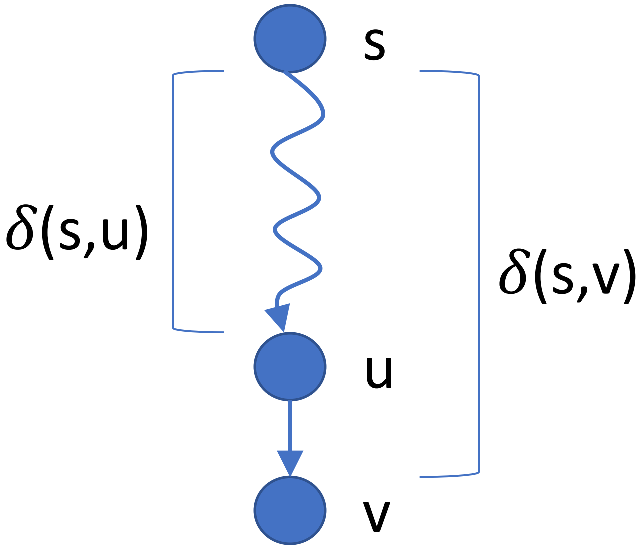
After BFS(s), for every $v \in V$, $d[v] = \delta(s,v)$

Proof of Main Theorem:

Suppose, for contradiction, that there exists $x \in V$ such that $d[x] \neq \delta(s,x)$. Clearly $x \neq s$.

- Let v be the **closest** node from s such that $d[v] \neq \delta(s,v)$
- By Lemma 0, $d[v] > \delta(s,v)$
- Consider some shortest path in s to v in G
  - Let (u,v) be the last edge on that path
  - Clearly $\delta(s,v) = \delta(s,u) + 1$
- Since u is closer to s than v, $d[u] = \delta(s,u)$ (By definition of v)
- So: $d[v] > \delta(s,v)$

$= d[u] + 1$ $\Rightarrow$ $\boxed{d[v] > d[u] + 1}$ (*)



Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow d[v] = d[u] + 1$

Contradicting (*) !

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$

Contradicting (*) !

Case 2. v is black

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$

Contradicting (*) !

Case 2. v is black

$\Rightarrow$ v was explored before u is explored

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$

Contradicting (*) !

Case 2. v is black

$\Rightarrow$ v was explored before u is explored

$\Rightarrow$ v entered Q before u enters Q

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$

Contradicting (*) !

Case 2. v is black

$\Rightarrow$ v was explored before u is explored

$\Rightarrow$ v entered Q before u enters Q

$\Rightarrow$ By Lemma 1, $d[v] \leq d[u]$

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Now consider the color of v **just before** u is explored. 3 possible cases

Case 1. v is white

$\Rightarrow$ When u is explored, u discovers v

$\Rightarrow$ $d[v] = d[u] + 1$ Contradicting (*) !

Case 2. v is black

$\Rightarrow$ v was explored before u is explored

$\Rightarrow$ v entered Q before u enters Q

$\Rightarrow$ By Lemma 1, $d[v] \leq d[u]$ Contradicting (*) !

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

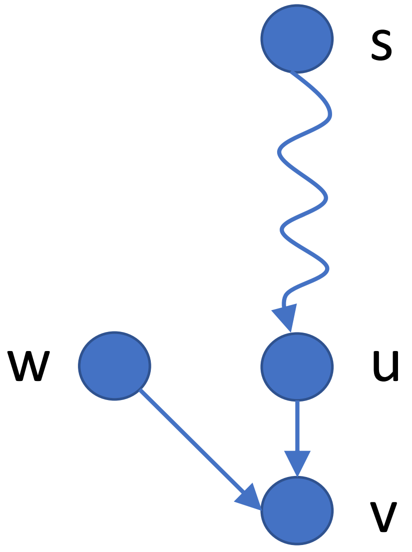
Case 3. v is grey (discovered but not explored)

Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored



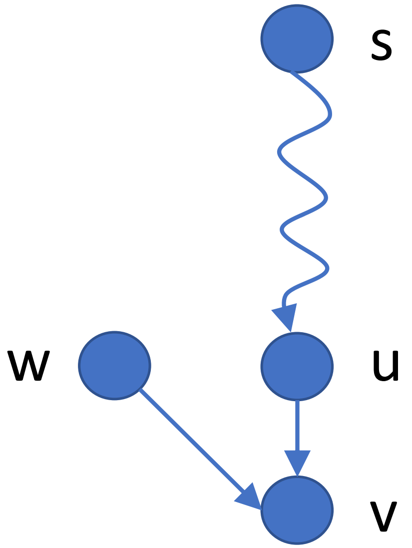
Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored

$\Rightarrow$ w is explored before u and $d[v] = d[w] + 1$



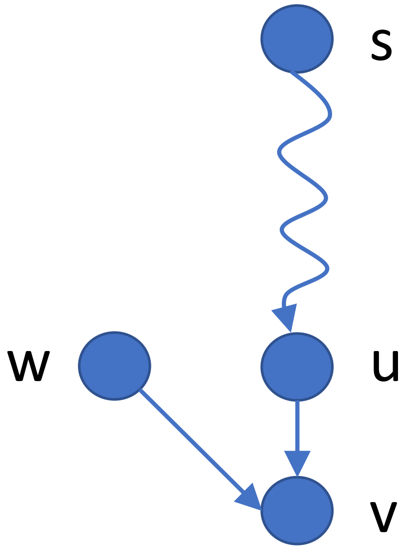
Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Case 3. v is grey (discovered but not explored)

⇒ Some node w discovered v before u is explored

⇒ $\underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$



Proof of Main Theorem contd:

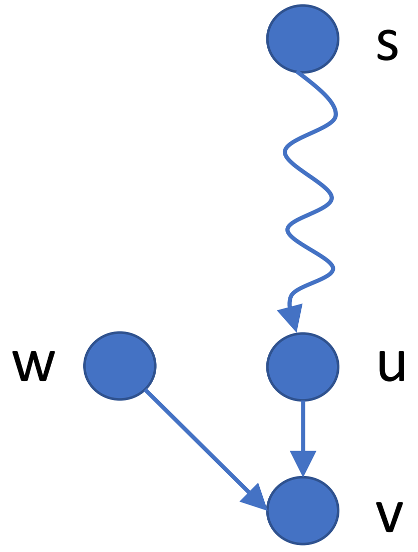
$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored

$\Rightarrow$ $\underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$

(a) $\Rightarrow$ w enters Q before u



Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

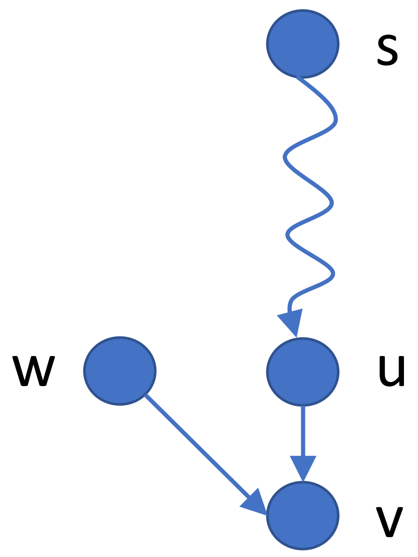
Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored

$\Rightarrow$ $\underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$

(a) $\Rightarrow$ w enters Q before u

$\Rightarrow d[w] \leq d[u]$ By Lemma 1



Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Case 3. v is grey (discovered but not explored)

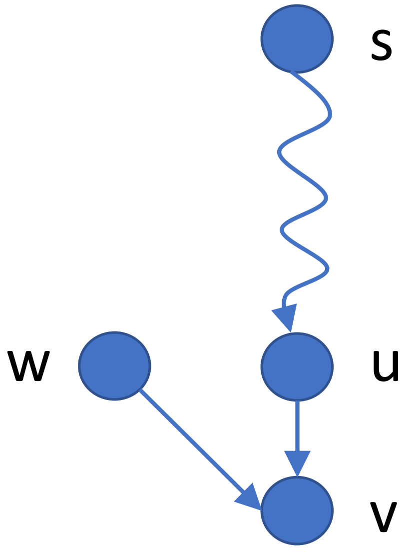
$\Rightarrow$ Some node w discovered v before u is explored

$\Rightarrow$ $\underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$

(a) $\Rightarrow$ w enters Q before u

$\Rightarrow d[w] \leq d[u]$ By Lemma 1

$\Rightarrow d[w] + 1 \leq d[u] + 1$



Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \qquad (*)$$

Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored

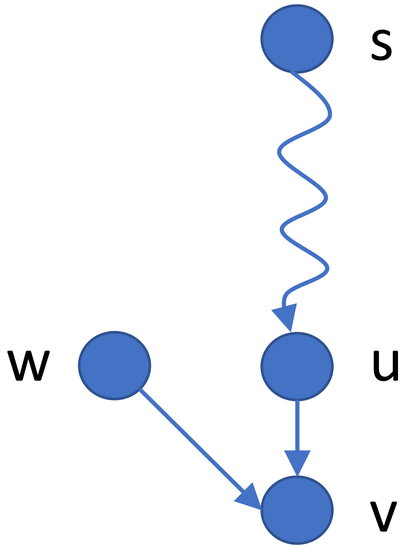
$\Rightarrow$ $\underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$

(a) $\Rightarrow$ w enters Q before u

$\Rightarrow d[w] \leq d[u]$ By Lemma 1

$\Rightarrow d[w] + 1 \leq d[u] + 1$

(b) $\Rightarrow d[v] \leq d[u] + 1$



Proof of Main Theorem contd:

$$\boxed{d[v] > d[u] + 1} \quad (*)$$

Case 3. v is grey (discovered but not explored)

$\Rightarrow$ Some node w discovered v before u is explored

$\Rightarrow \underbrace{\text{w is explored before u}}_{(a)}$ and $\underbrace{d[v] = d[w] + 1}_{(b)}$

(a) $\Rightarrow$ w enters Q before u

$\Rightarrow d[w] \leq d[u]$ By Lemma 1

$\Rightarrow d[w] + 1 \leq d[u] + 1$

(b) $\Rightarrow d[v] \leq d[u] + 1$ Contradicting (*) !