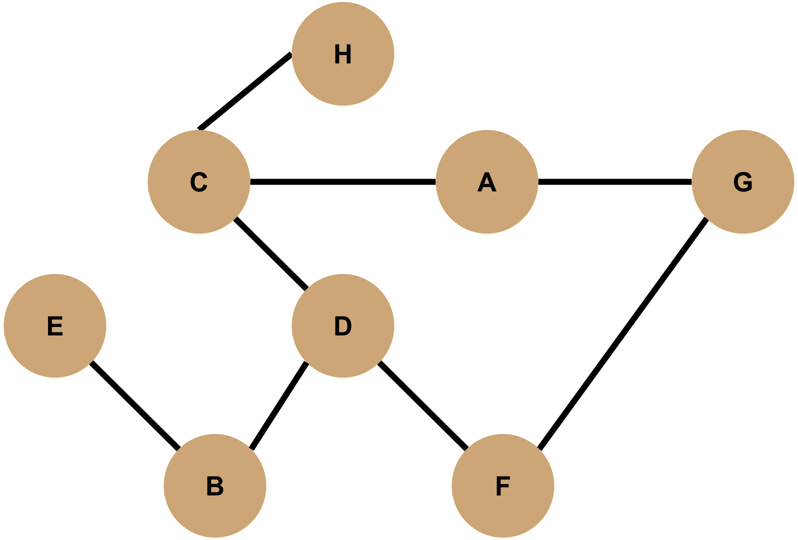
BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	F
B	F
C	F
D	F
E	F
F	F
G	F
H	F

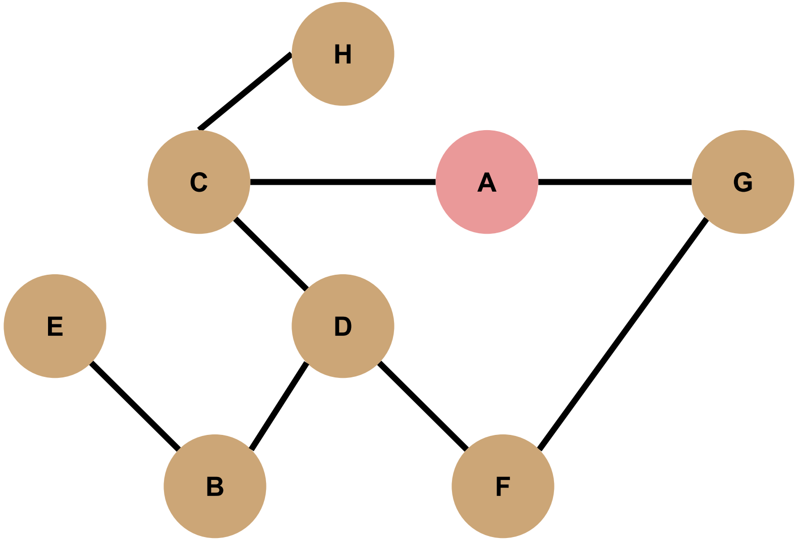
Order of BFS:

Queue: []

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	F
D	F
E	F
F	F
G	F
H	F

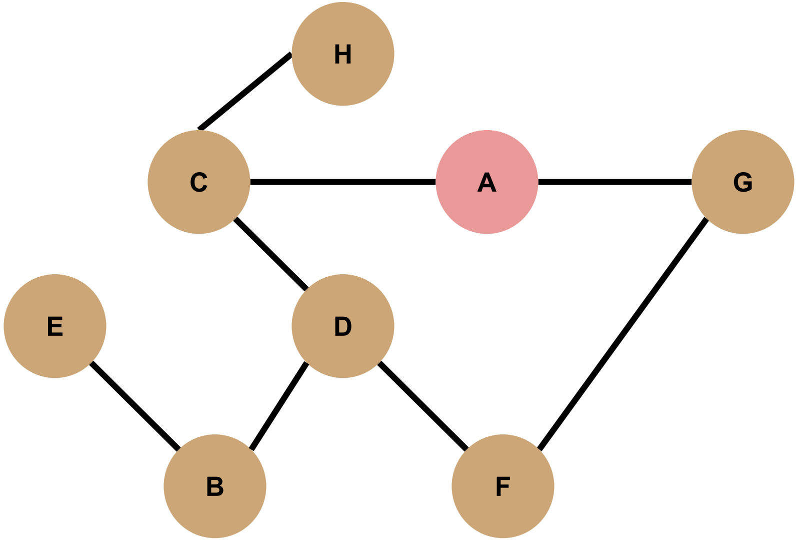
Order of BFS:

Queue: [A]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	F
D	F
E	F
F	F
G	F
H	F

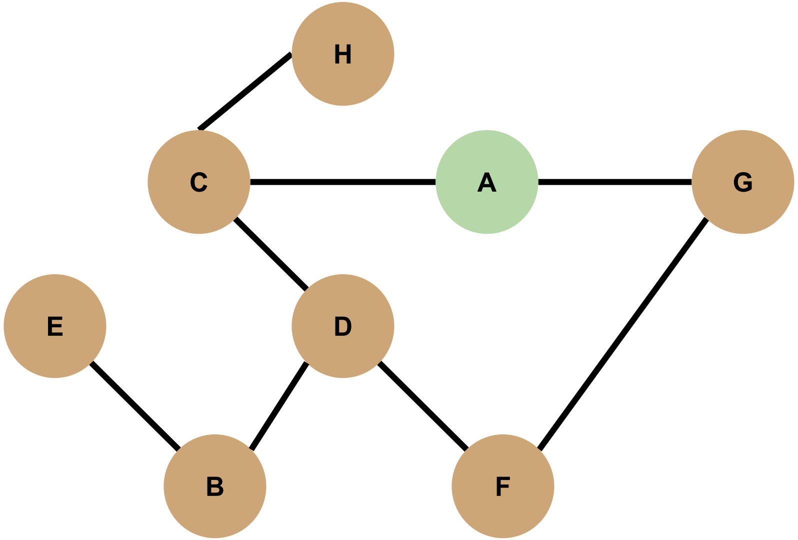
Order of BFS:

Queue: [A]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ● Dequeue vertex v
2. For every unmarked neighbor n:
    ● Mark neighbor
    ● Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	F
D	F
E	F
F	F
G	F
H	F

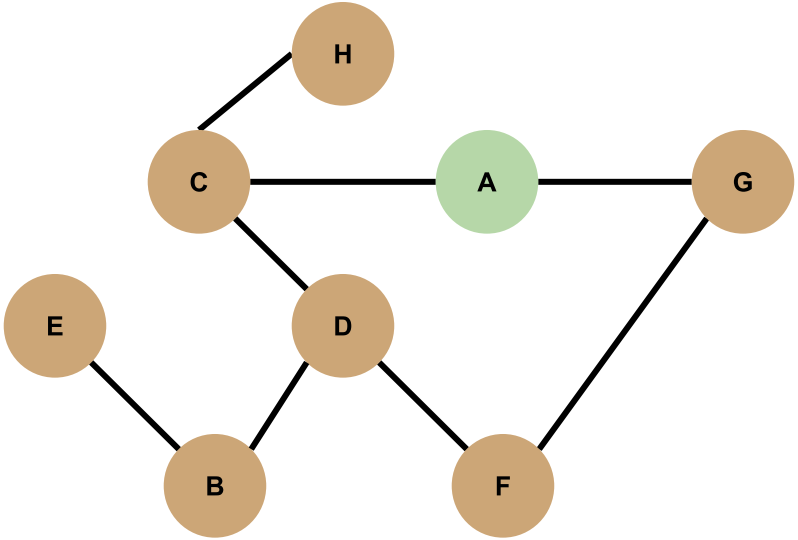
Order of BFS: A

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	F
D	F
E	F
F	F
G	F
H	F

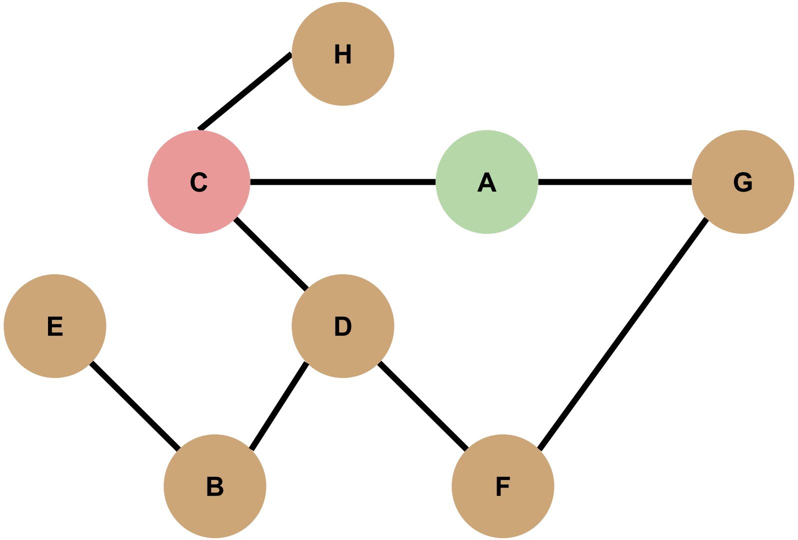
Order of BFS: A

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •    Dequeue vertex v
2. For every unmarked neighbor n:
    •    Mark neighbor
    •    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	F
H	F

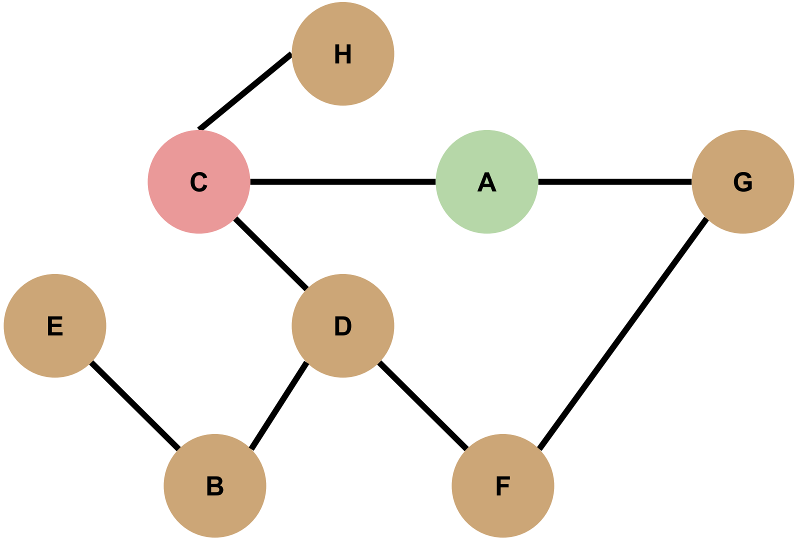
Order of BFS: A

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	F
H	F

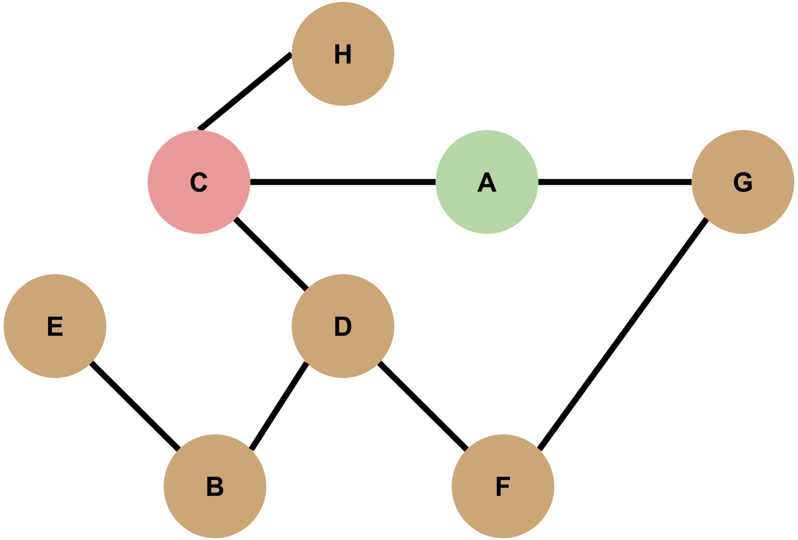
Order of BFS: A

Queue: [C]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •    Dequeue vertex v
2. For every unmarked neighbor n:
    •    Mark neighbor
    •    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	F
H	F

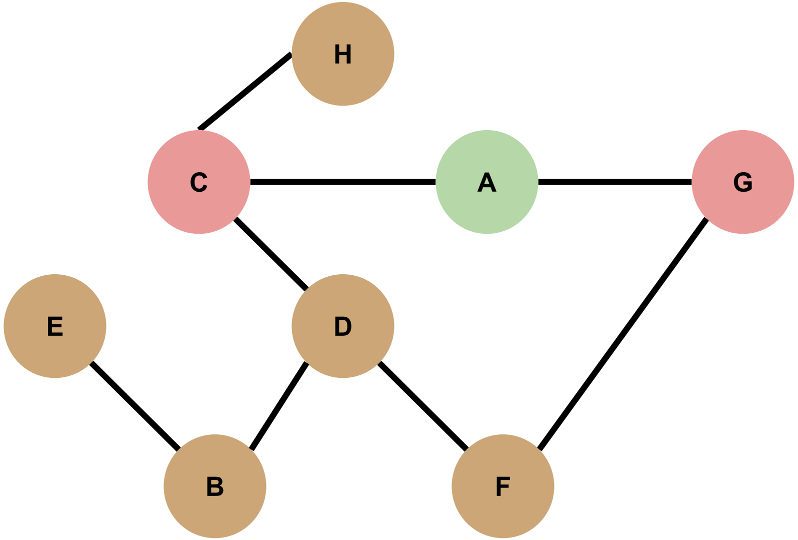
Order of BFS: A

Queue: [C]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

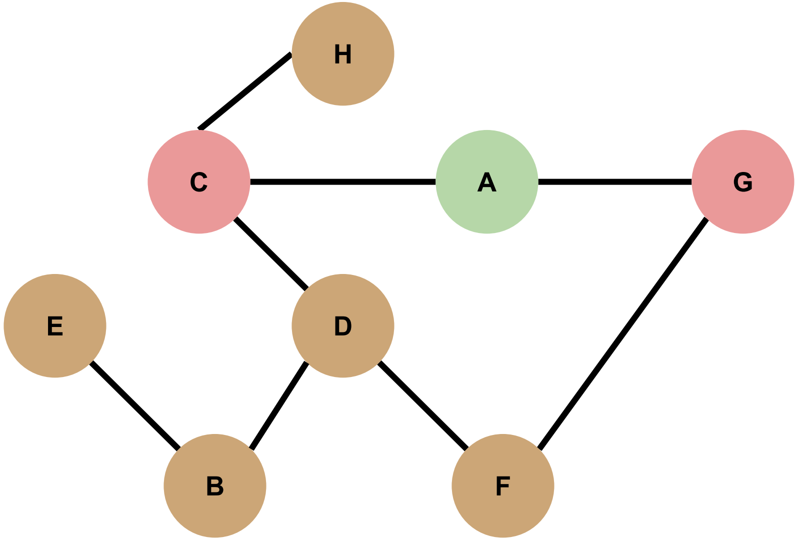
Order of BFS: A

Queue: [C]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

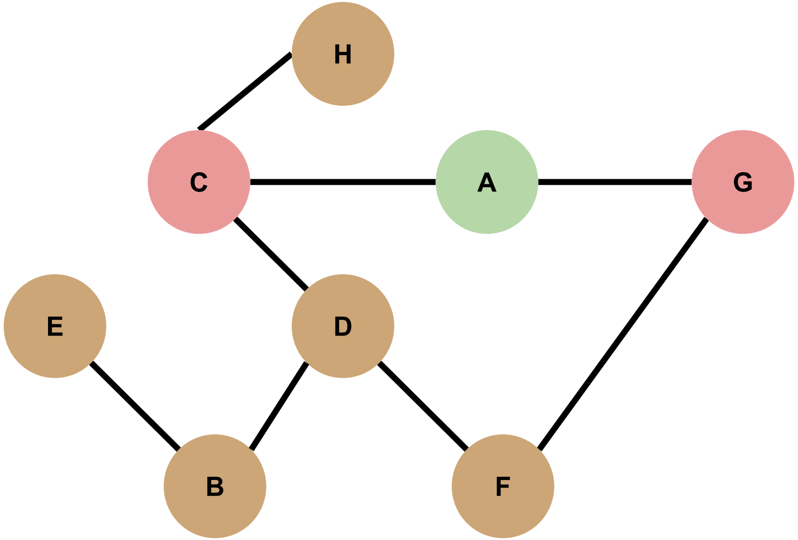
Order of BFS: A

Queue: [C, G]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●    Dequeue vertex v
2. For every unmarked neighbor n:
    ●    Mark neighbor
    ●    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

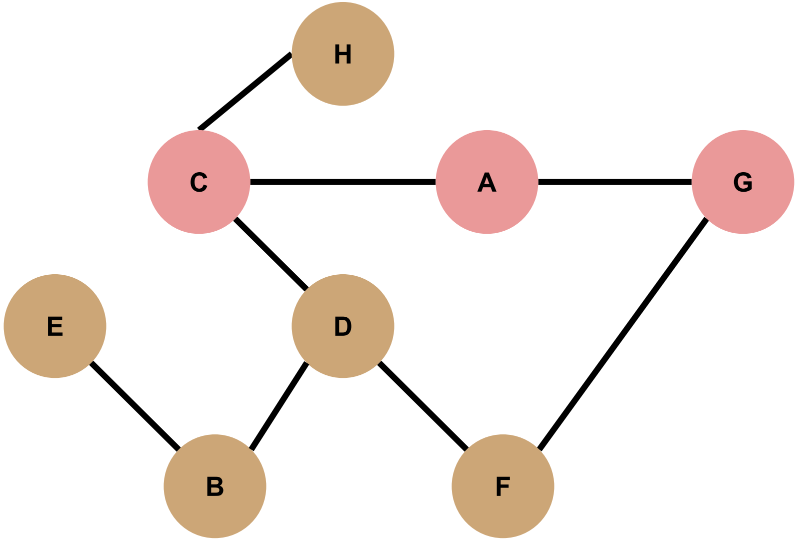
Order of BFS: A

Queue: [C, G]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Order of BFS: A

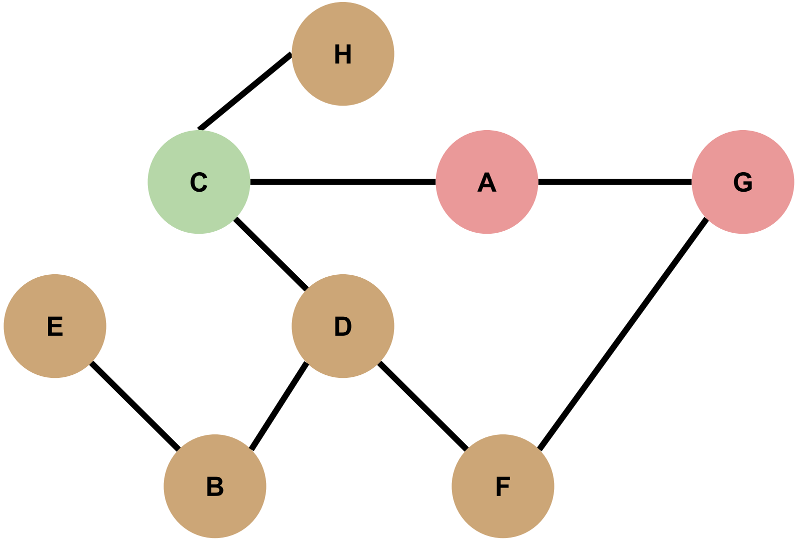
Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

Queue: [C, G]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ● Dequeue vertex v
2. For every unmarked neighbor n:
    ● Mark neighbor
    ● Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

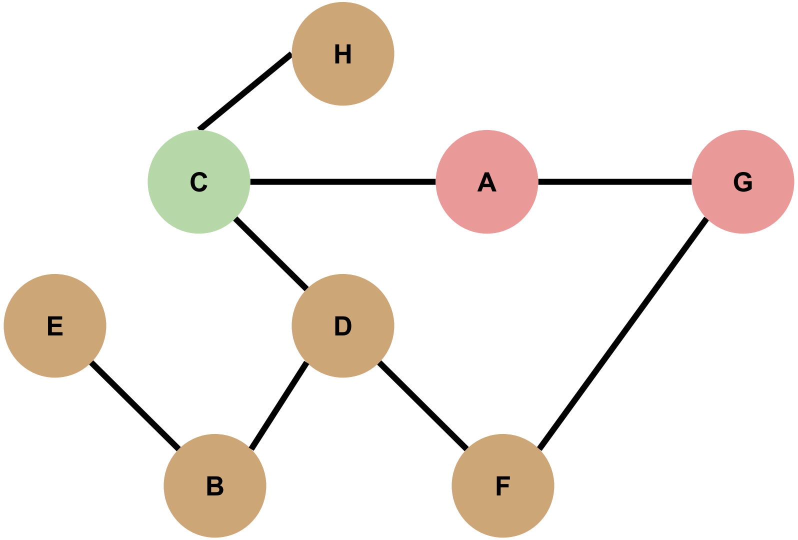
Order of BFS: A, C

Queue: [G]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●    Dequeue vertex v
2. For every unmarked neighbor n:
    ●    Mark neighbor
    ●    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	F
E	F
F	F
G	T
H	F

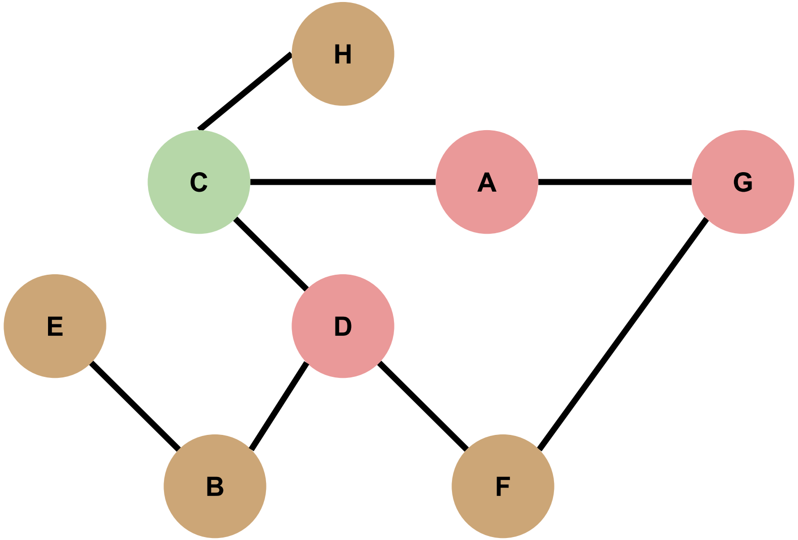
Order of BFS: A, C

Queue: [G]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	F

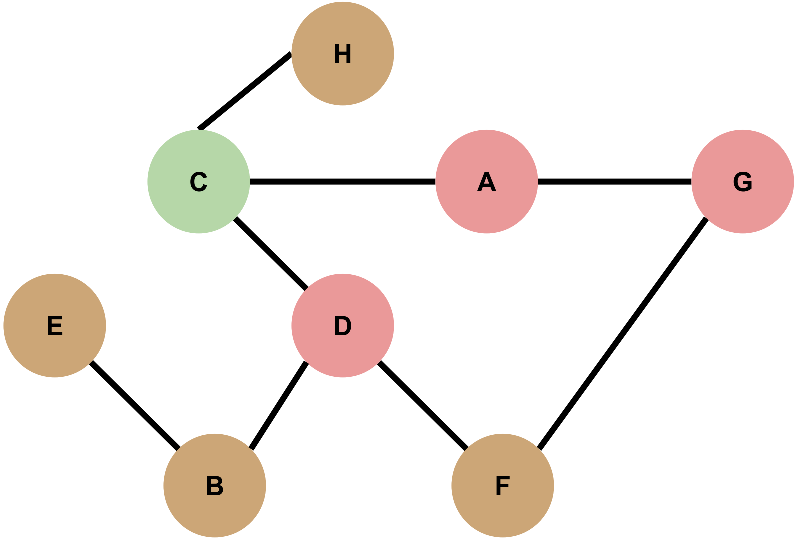
Order of BFS: A, C

Queue: [G]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	F

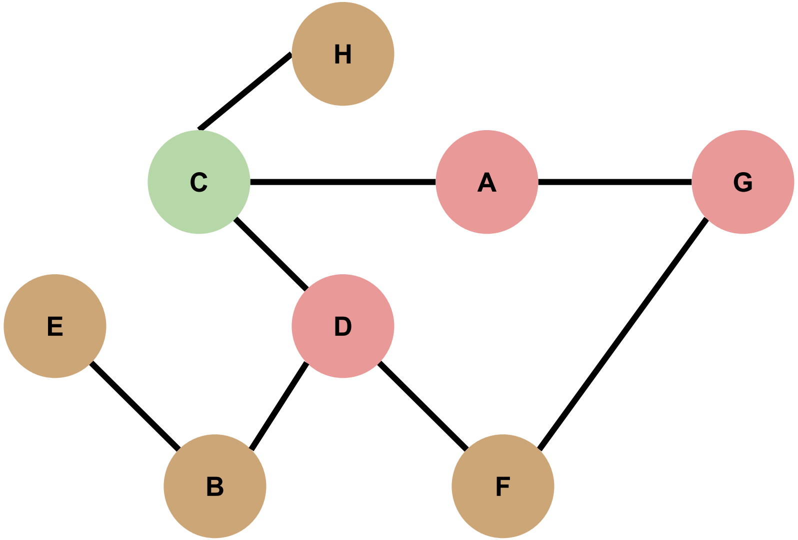
Order of BFS: A, C

Queue: [G, D]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •  Dequeue vertex v
2. For every unmarked neighbor n:
    •  Mark neighbor
    •  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	F

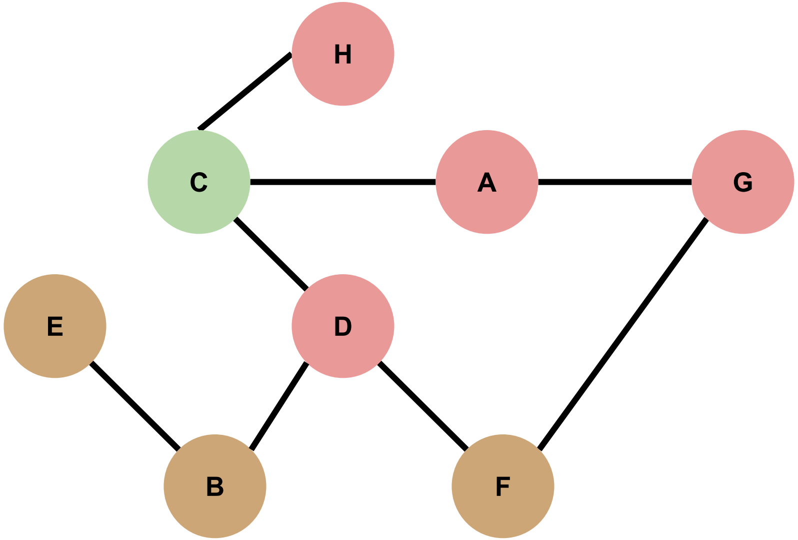
Order of BFS: A, C

Queue: [G, D]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

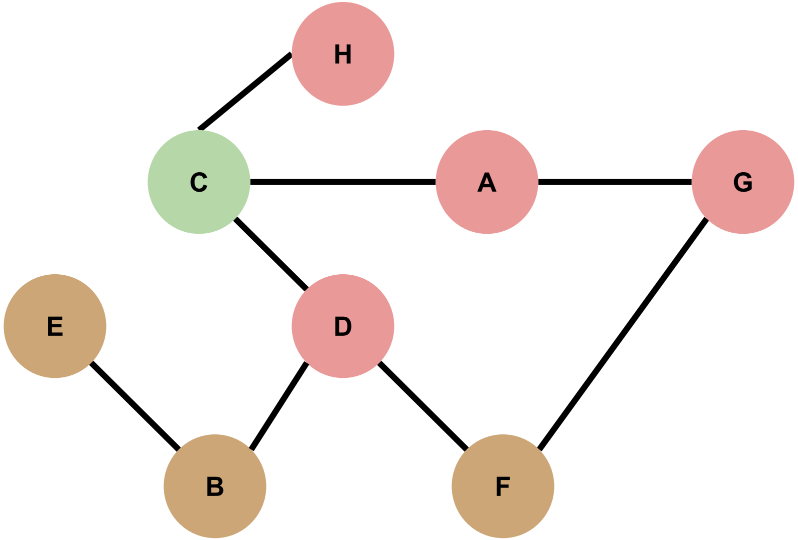
Order of BFS: A, C

Queue: [G, D]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •  Dequeue vertex v
2. For every unmarked neighbor n:
    •  Mark neighbor
    •  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

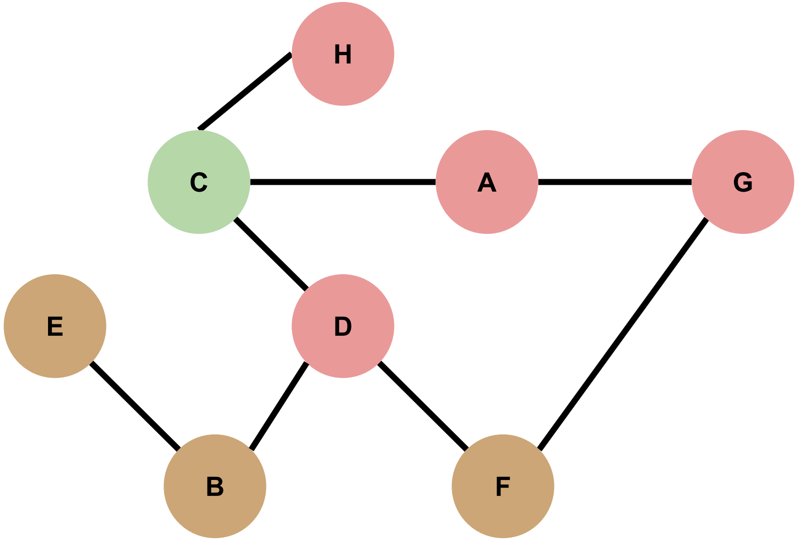
Order of BFS: A, C

Queue: [G, D, H]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

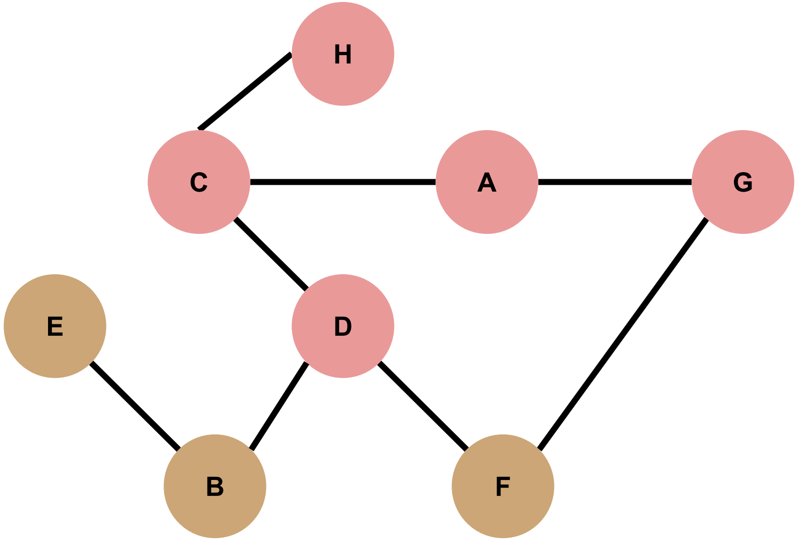
Order of BFS: A, C

Queue: [G, D, H]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

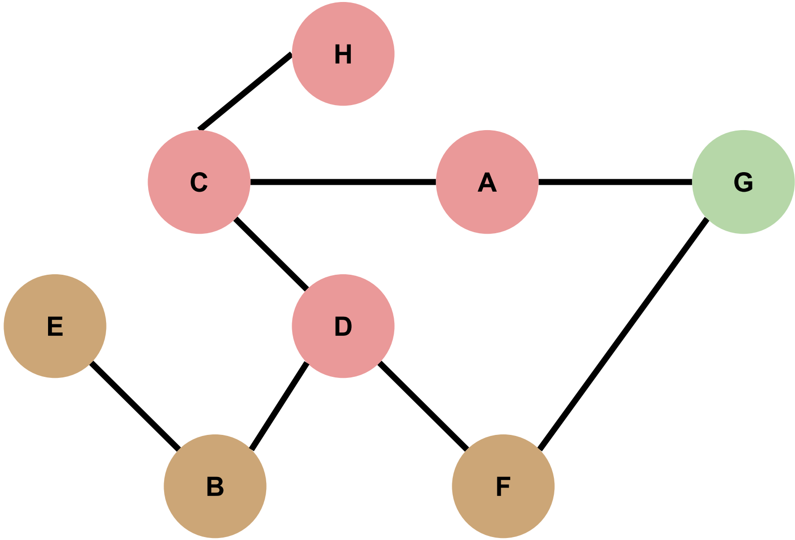
Order of BFS: A, C

Queue: [G, D, H]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

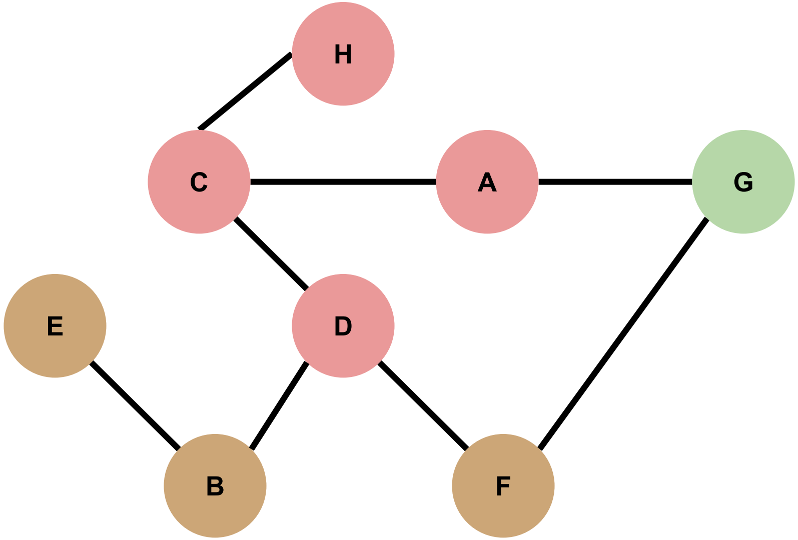
Order of BFS: A, C, G

Queue: [D, H]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	F
G	T
H	T

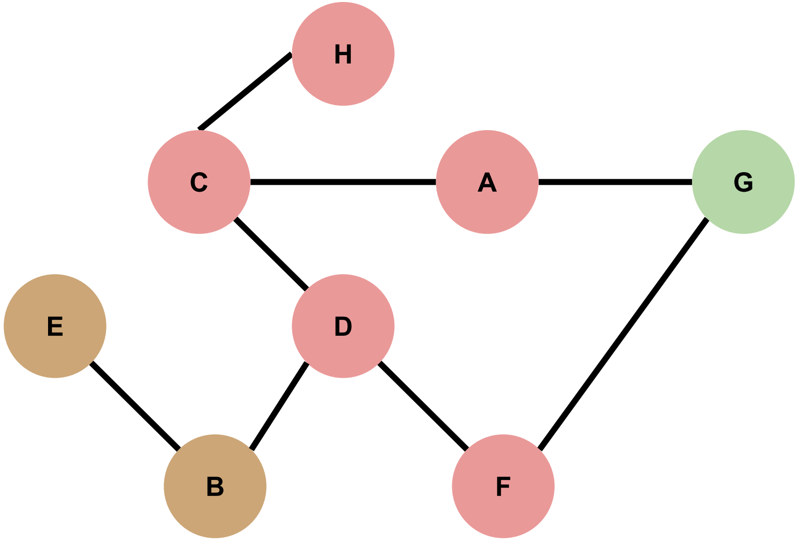
Order of BFS: A, C, G

Queue: [D, H]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●    Dequeue vertex v
2. For every unmarked neighbor n:
    ●    Mark neighbor
    ●    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

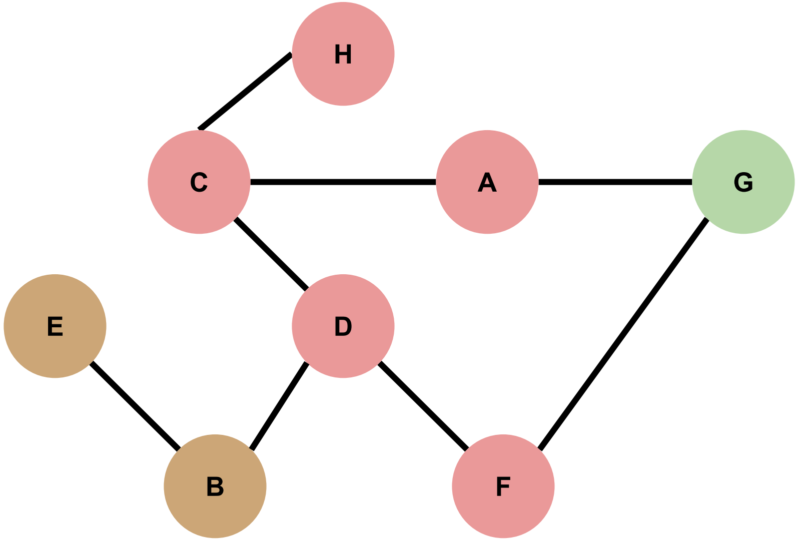
Order of BFS: A, C, G

Queue: [D, H]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

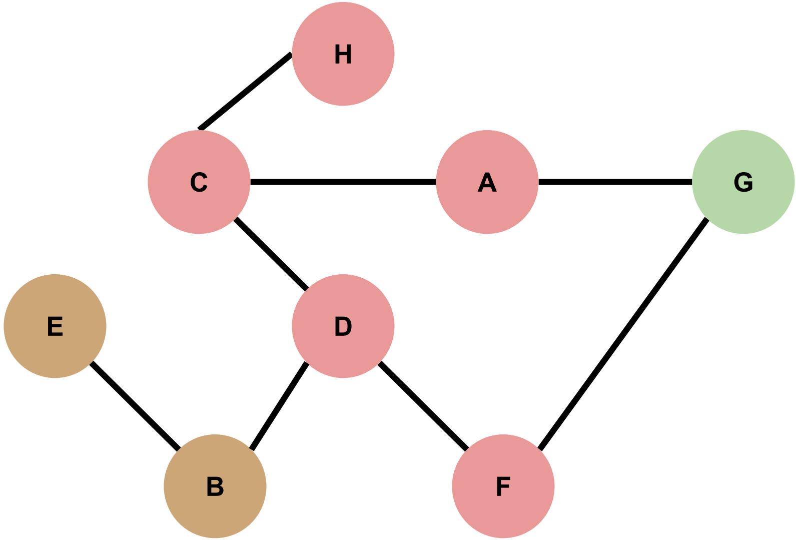
Order of BFS: A, C, G

Queue: [D, H, F]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

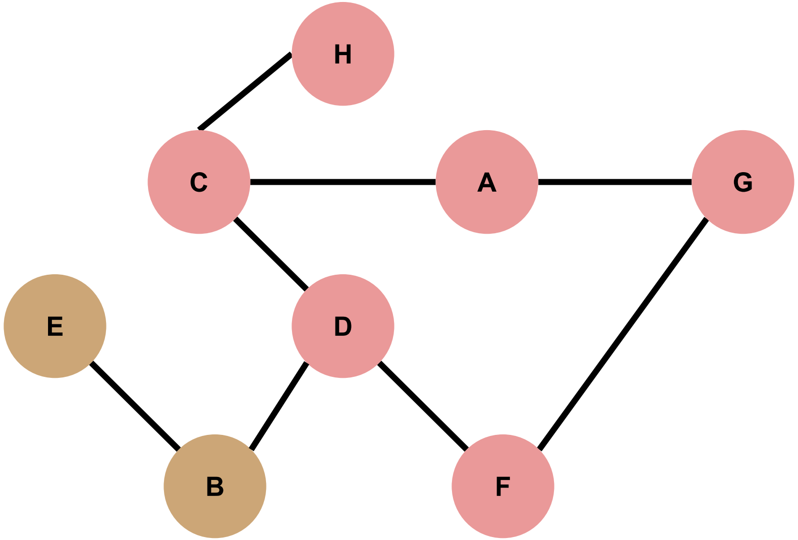
Order of BFS: A, C, G

Queue: [D, H, F]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

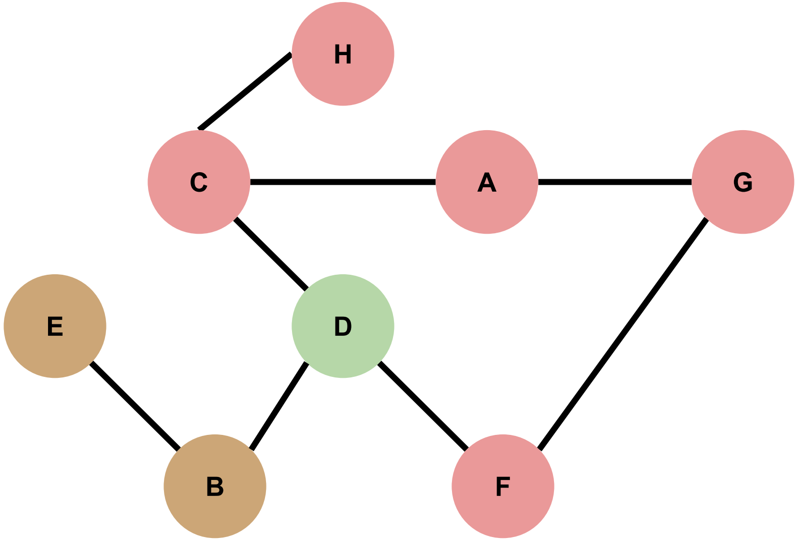
Order of BFS: A, C, G

Queue: [D, H, F]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●    Dequeue vertex v
2. For every unmarked neighbor n:
    ●    Mark neighbor
    ●    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

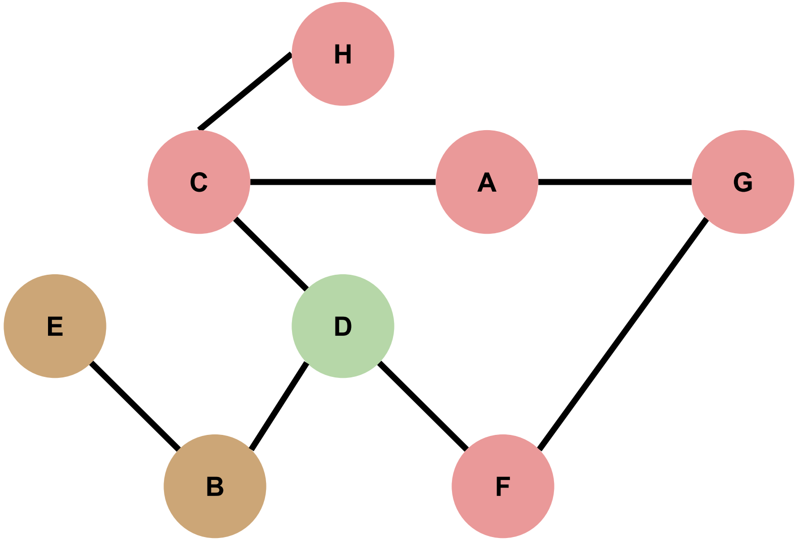
Order of BFS: A, C, G, D

Queue: [H, F]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •  Dequeue vertex v
2. For every unmarked neighbor n:
    •  Mark neighbor
    •  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	F
C	T
D	T
E	F
F	T
G	T
H	T

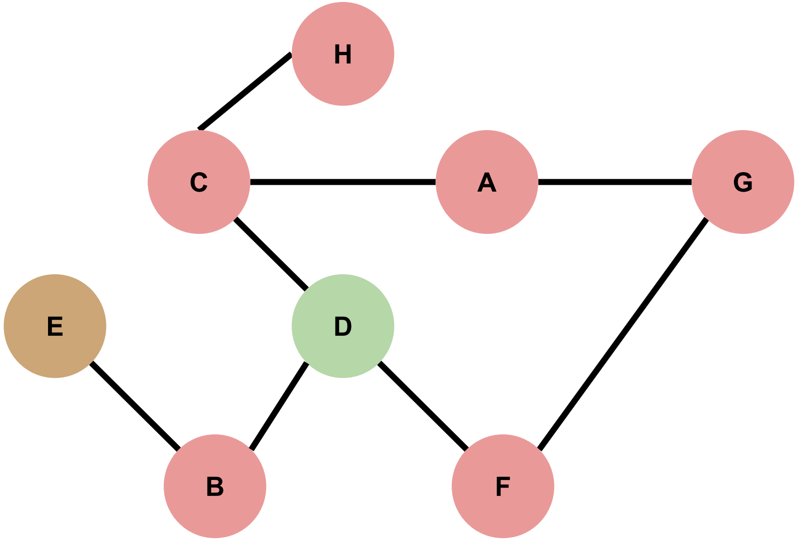
Order of BFS: A, C, G, D

Queue: [H, F]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●    Dequeue vertex v
2. For every unmarked neighbor n:
    ●    Mark neighbor
    ●    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

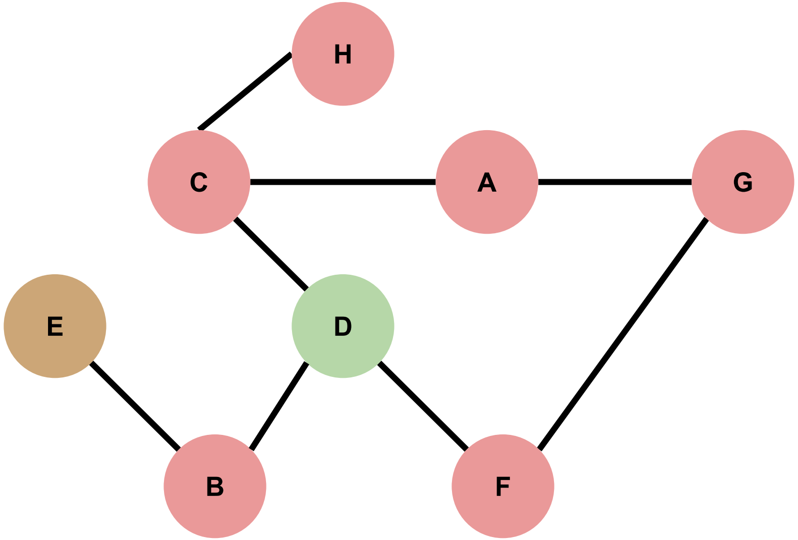
Order of BFS: A, C, G, D

Queue: [H, F]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •   Dequeue vertex v
2. For every unmarked neighbor n:
    •   Mark neighbor
    •   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

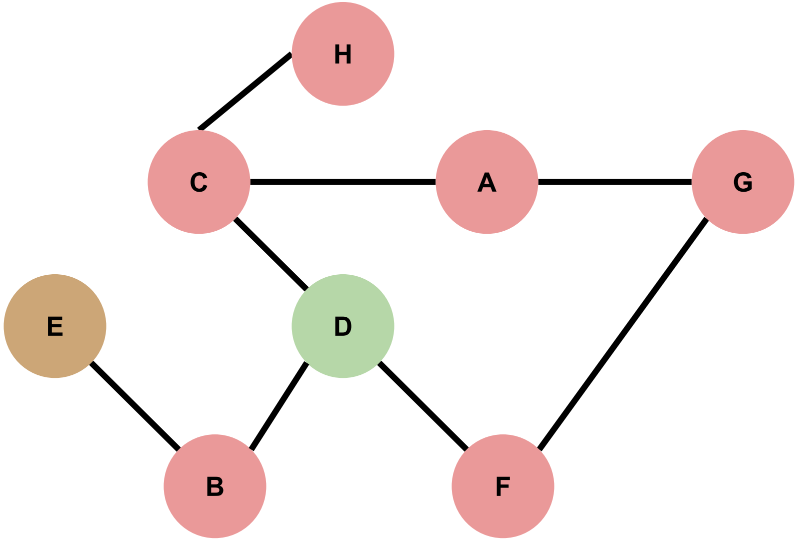
Order of BFS: A, C, G, D

Queue: [H, F, B]

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

Order of BFS: A, C, G, D

Queue: [H, F, B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

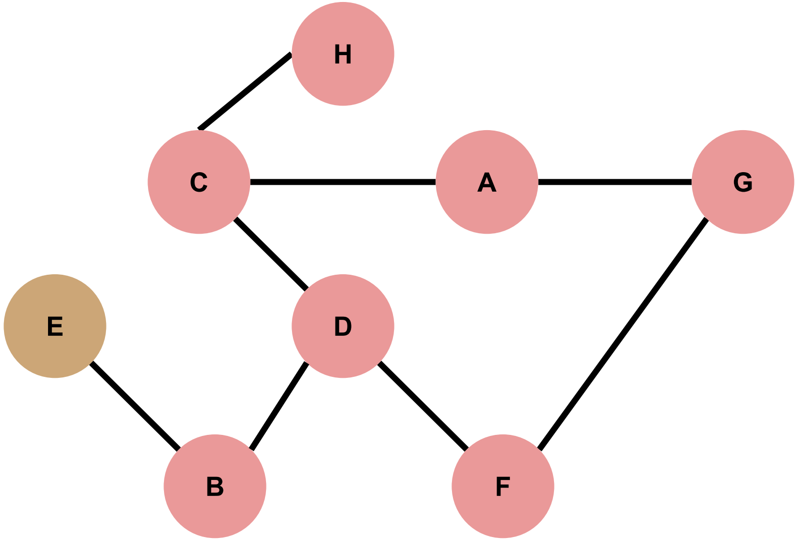
Initialize Queue with Starting Vertex & Mark it

1. While Queue is not empty:

- *Dequeue vertex v*

2. For every unmarked neighbor n:

- *Mark neighbor*
- *Enqueue n to Queue*



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

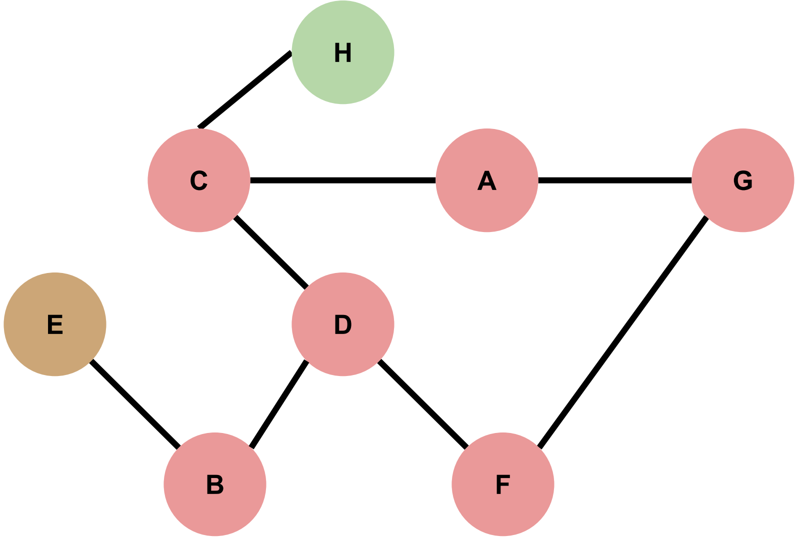
Order of BFS: A, C, G, D

Queue: [H, F, B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •  Dequeue vertex v
2. For every unmarked neighbor n:
    •  Mark neighbor
    •  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

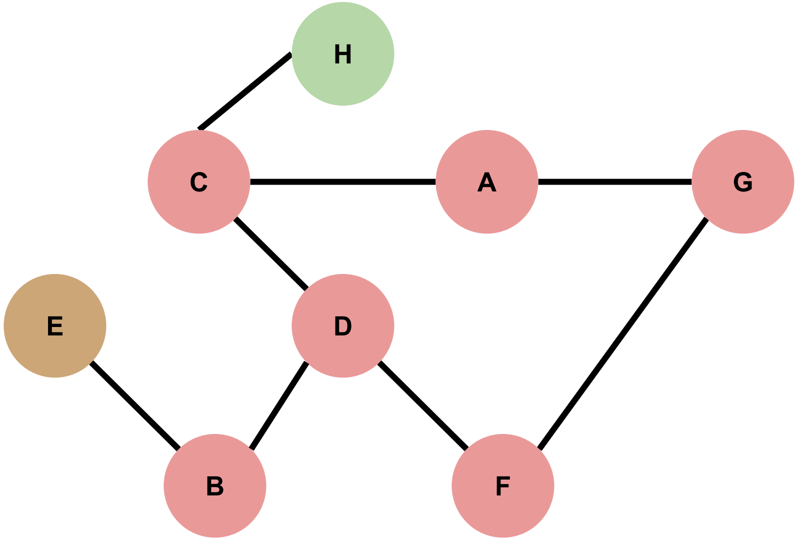
Order of BFS: A, C, G, D, H

Queue: [F, B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

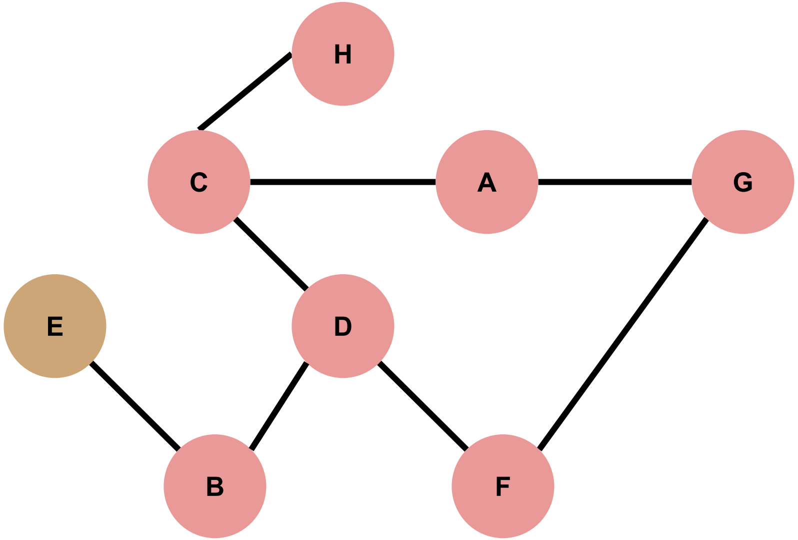
Order of BFS: A, C, G, D, H

Queue: [F, B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

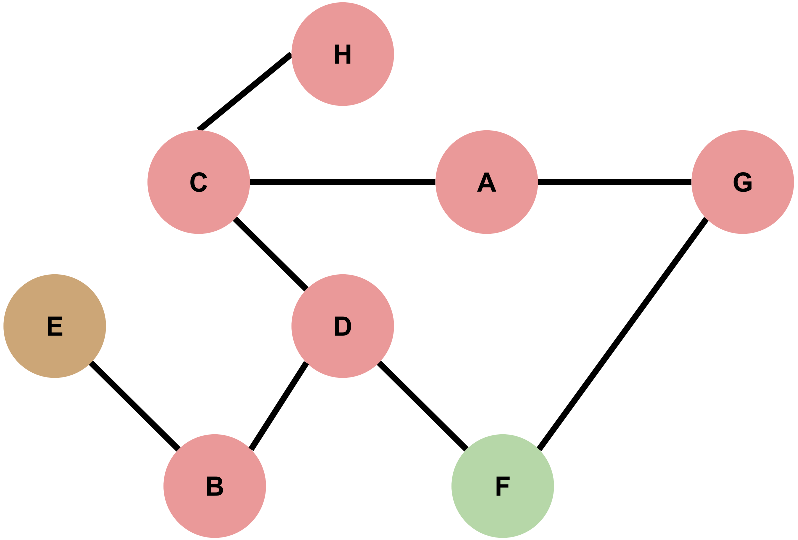
Order of BFS: A, C, G, D, H

Queue: [F, B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •  Dequeue vertex v
2. For every unmarked neighbor n:
    •  Mark neighbor
    •  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

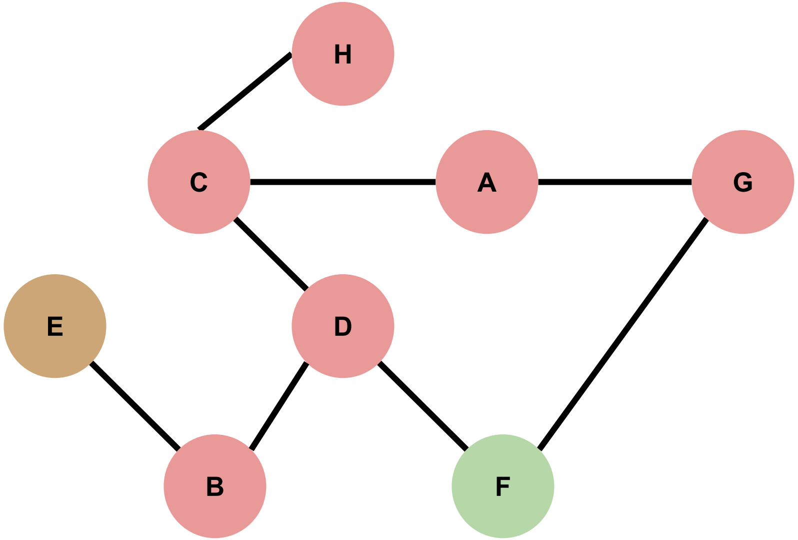
Order of BFS: A, C, G, D, H, F

Queue: [B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

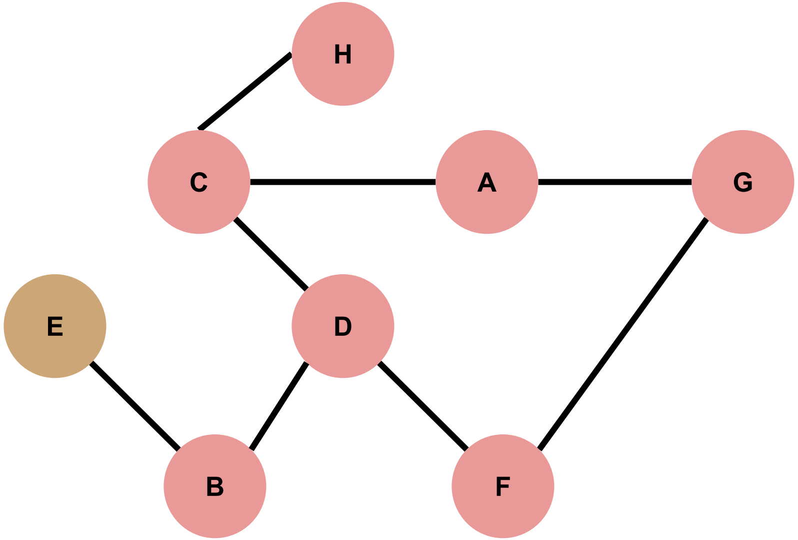
Order of BFS: A, C, G, D, H, F

Queue: [B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

Order of BFS: A, C, G, D, H, F

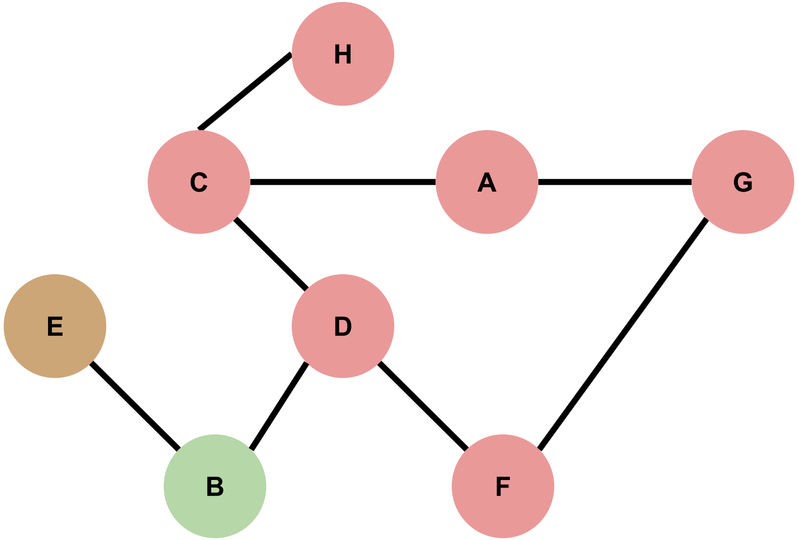
Queue: [B]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
- *Dequeue vertex v*

2. For every unmarked neighbor n:
- *Mark neighbor*
- *Enqueue n to Queue*



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

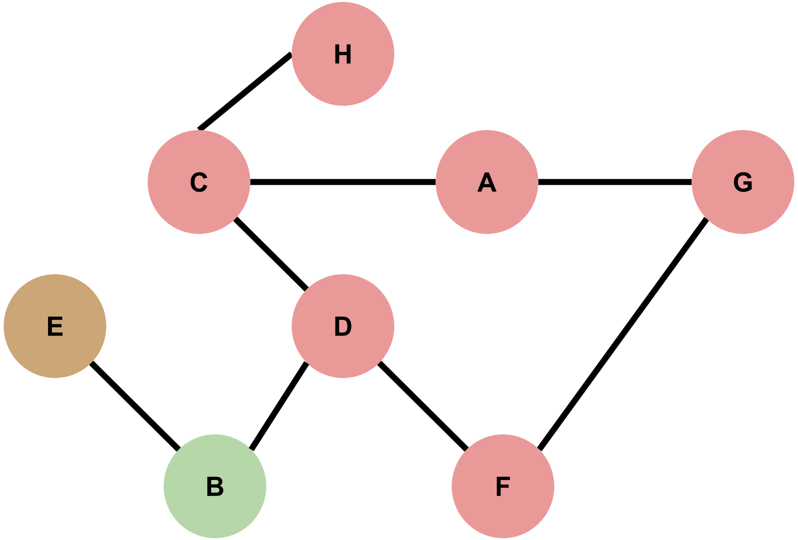
Order of BFS: A, C, G, D, H, F, B

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	F
F	T
G	T
H	T

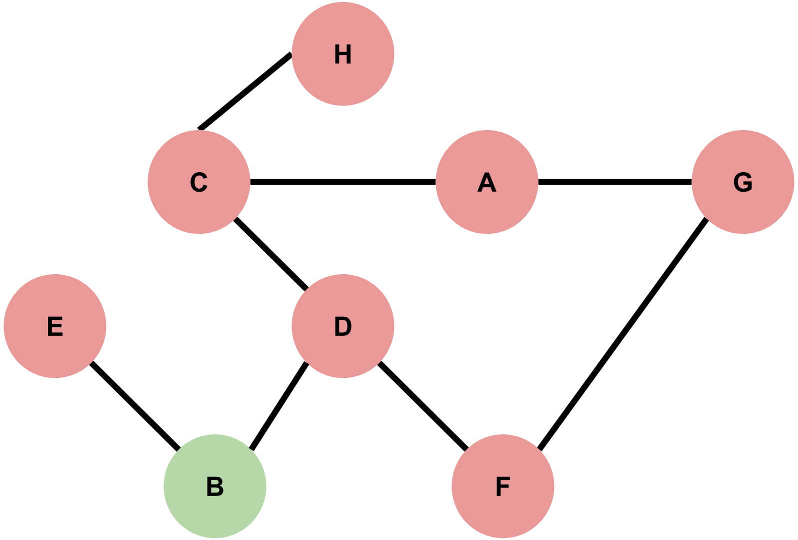
Order of BFS: A, C, G, D, H, F, B

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

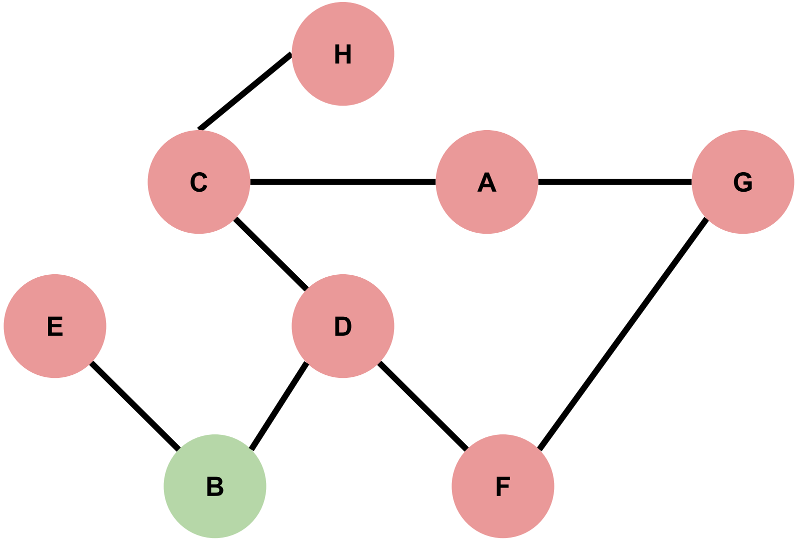
Order of BFS: A, C, G, D, H, F, B

Queue: []

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

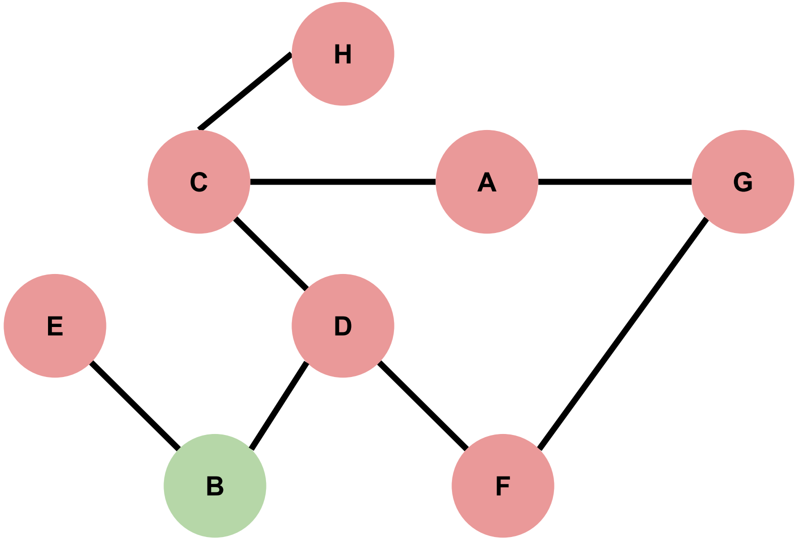
Order of BFS: A, C, G, D, H, F, B

Queue: [E]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    •    Dequeue vertex v
2. For every unmarked neighbor n:
    •    Mark neighbor
    •    Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

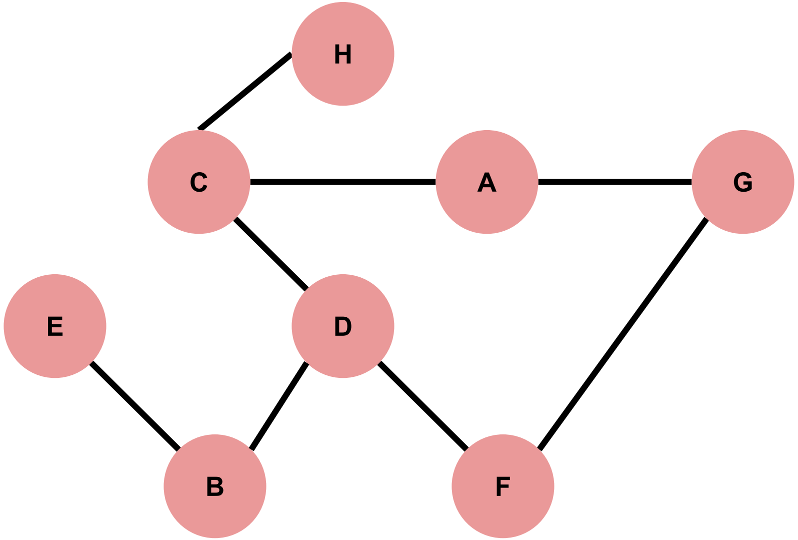
Order of BFS: A, C, G, D, H, F, B

Queue: [E]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

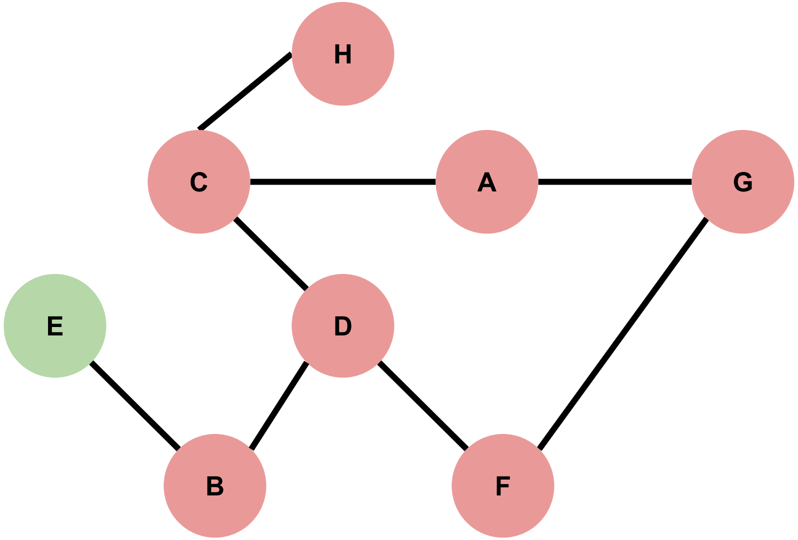
Order of BFS: A, C, G, D, H, F, B

Queue: [E]

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    • Dequeue vertex v
2. For every unmarked neighbor n:
    • Mark neighbor
    • Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

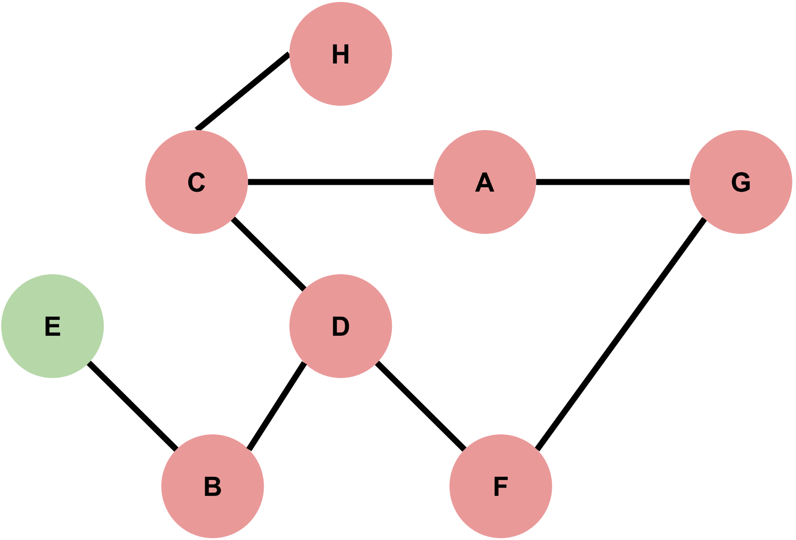
Order of BFS: A, C, G, D, H, F, B, E

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

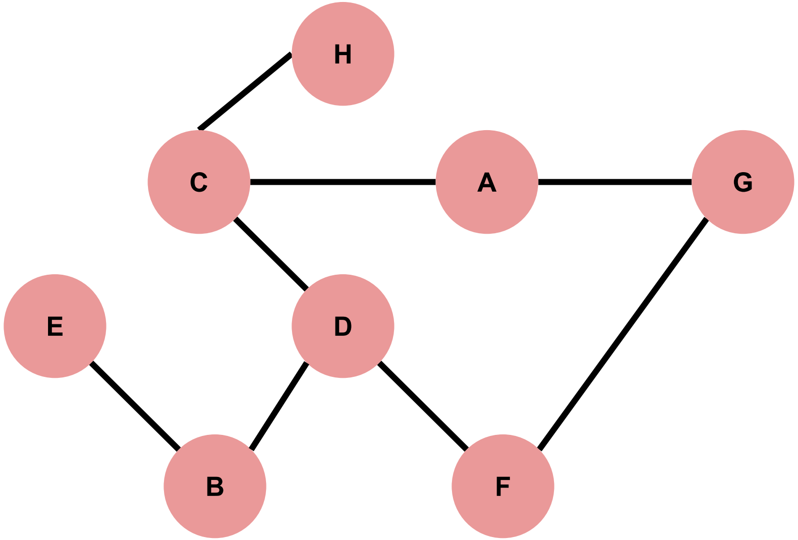
Order of BFS: A, C, G, D, H, F, B, E

Queue: []

BFS

Starting from A, write the order in which vertices are visited using ***BFS****.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●  Dequeue vertex v
2. For every unmarked neighbor n:
    ●  Mark neighbor
    ●  Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

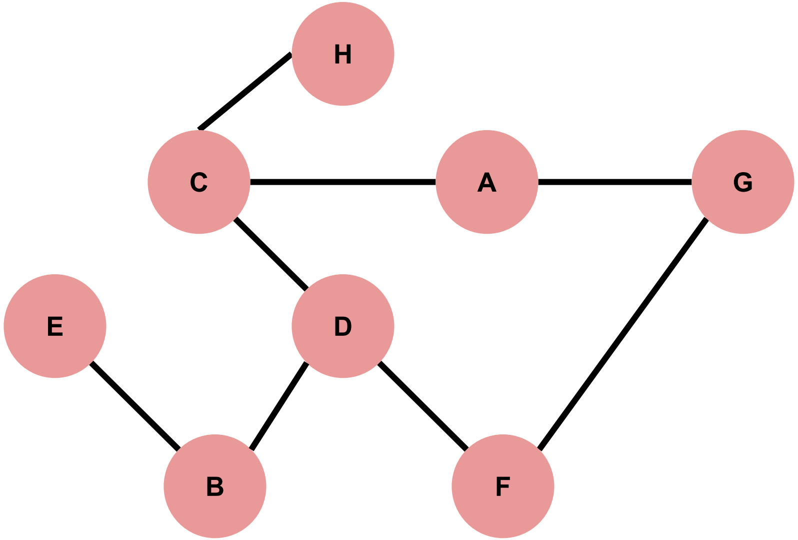
Order of BFS: A, C, G, D, H, F, B, E

Queue: []

BFS

*Starting from A, write the order in which vertices are visited using **BFS**.*

```
Initialize Queue with Starting Vertex & Mark it
1. While Queue is not empty:
    ●   Dequeue vertex v
2. For every unmarked neighbor n:
    ●   Mark neighbor
    ●   Enqueue n to Queue
```



Vertex	marked[]
A	T
B	T
C	T
D	T
E	T
F	T
G	T
H	T

Order of BFS: A, C, G, D, H, F, B, E

Queue: []