Graph representations

- Adjacency list
- Adjacency matrix
Adjacency list

$V$

- $u$ connects to $e$, $g$
- $v$ connects to $e$, $f$
- $w$ connects to $f$, $g$, $h$
- $z$ connects to $h$
Adjacency matrix

\begin{align*}
\begin{array}{ccc}
0 & 1 & 2 & 3 \\
\hline
u & e & g \\
v & e & f \\
w & g & f & h \\
z & h \\
\end{array}
\end{align*}
Graph traversals

- Depth first search (DFS)
- Breadth first search (BFS)
DFS
DFS
DFS
DFS
### DepthFirstSearch($G$)

1. Mark each vertex in $V$ with 0 as a mark of being unvisited
2. $\text{count} \leftarrow 0$
3. for each vertex $v$ in $V$ do
   4. if $v$ is marked with 0 then
   5. DFS($v$)

### DFS($v$)

1. $\text{count} \leftarrow \text{count} + 1$
2. Mark $v$ with $\text{count}$
3. for each vertex $w$ in $V$ adjacent to $v$ do
   4. if $w$ is marked with 0 then
   5. DFS($w$)
BFS
BFS

0

1

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P
BFS
BFS
BreadthFirstSearch($G$)

1. Mark each vertex in $V$ with 0 as a mark of being unvisited
2. $count \leftarrow 0$
3. for each vertex $v$ in $V$ do
4. if $v$ is marked with 0 then
5. BFS($v$)

BFS($v$)

1. $count \leftarrow count + 1$
2. Mark $v$ with $count$
3. Initialize a queue with $v$
4. while queue is not empty do
5. for each vertex $w$ in $V$ adjacent to the front vertex do
6. if $w$ is marked with 0 then
7. $count \leftarrow count + 1$
8. Mark $w$ with $count$
9. Add $w$ to the queue
10. Remove the front vertex from the queue