Complete the following method **areNeighbors** using Princeton's Graph API. * 8 points
The method should return true if u & v are neighbors.

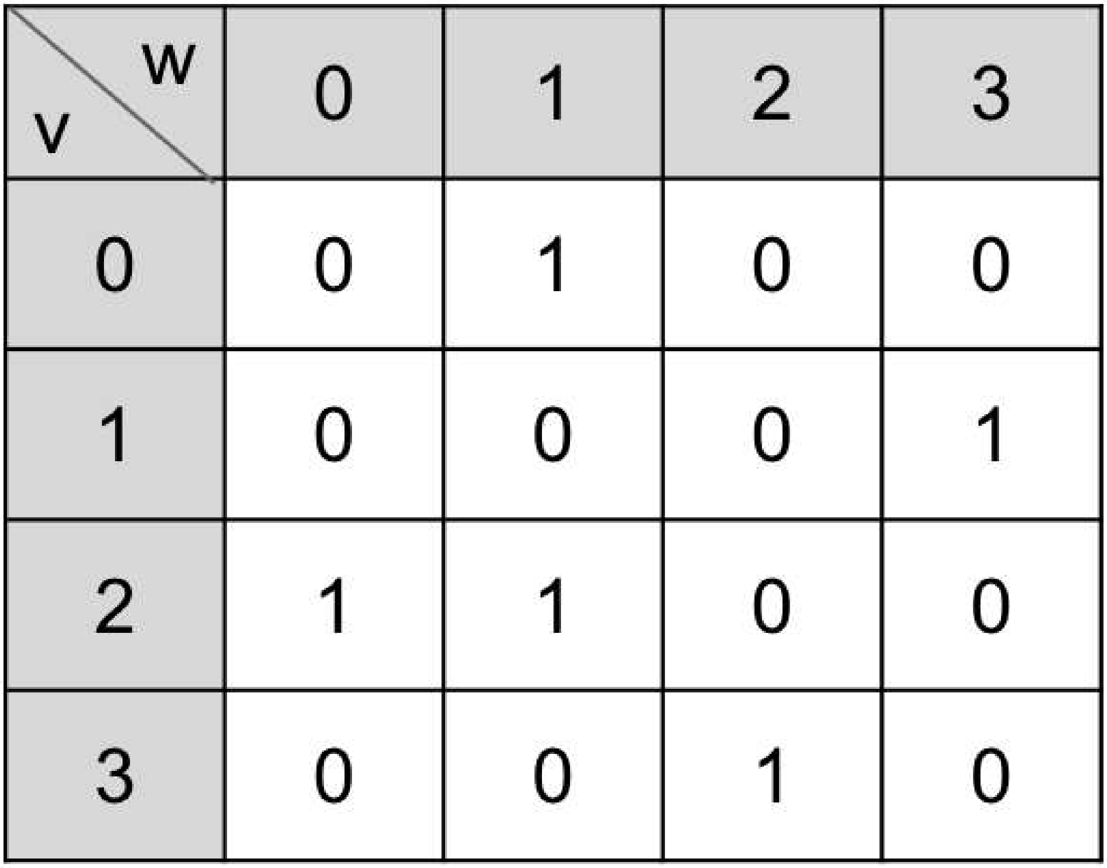
```
/** Returns true if u & v are neighbors */
public static boolean areNeighbors(Graph g, int u, int v) {
  for (int w : ___1___) {
    if (___2___ == u) {
      return ___3___;
    }
  }

  return ___4___;
}
```

	g.V()	g.adj(v)	v	w	true	false
Blank 1	○	○	○	○	○	○
Blank 2	○	○	○	○	○	○
Blank 3	○	○	○	○	○	○
Blank 4	○	○	○	○	○	○

Select all of the following that are true about a graph represented by the following Adjacency Matrix

2 points



v \ w	0	1	2	3
0	0	1	0	0
1	0	0	0	1
2	1	1	0	0
3	0	0	1	0

□ There is an edge from 2 to 3

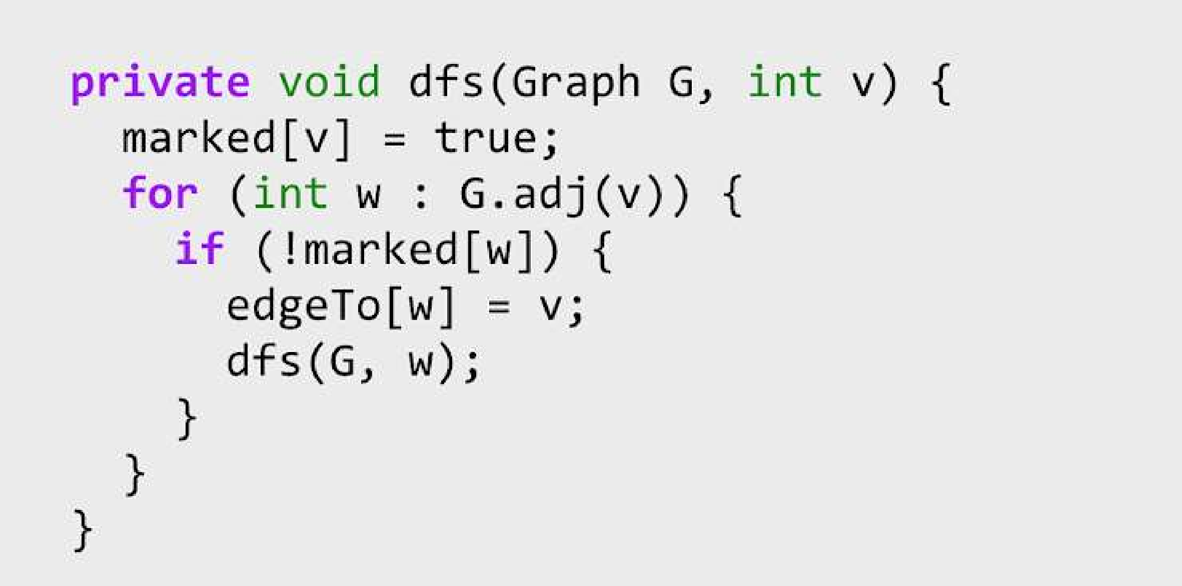
□ There is an edge from 3 to 2

□ There is an edge from 0 to itself

□ This is a directed graph

When performing DFS, why do we set edgeTo[w] = v?

2 points



```
private void dfs(Graph G, int v) {
  marked[v] = true;
  for (int w : G.adj(v)) {
    if (!marked[w]) {
      edgeTo[w] = v;
      dfs(G, w);
    }
  }
}
```

○ To indicate that v is our source vertex for the overall DFS traversal

○ To indicate that v is the next vertex that should be explored in our DFS traversal

○ To indicate that the path from s to w includes edge v→w

○ To indicate that the path from s to v includes edge w→v

Select all that are true about the DepthFirstPaths and BreadthFirstPaths algorithms introduced in lecture.

2 points

□ Depth First Paths will return the shortest path from the source to any vertex

□ Breadth First Paths will run asymptotically faster than Depth First Paths

□ When visiting a vertex in Breadth First Paths, we add its unmarked neighbors to a First-In-First-Out queue

□ Breadth First Paths & Depth First Paths are guaranteed to reach the same set of vertices

A copy of your responses will be emailed to yiyunchen@berkeley.edu.

Submit

Clear form