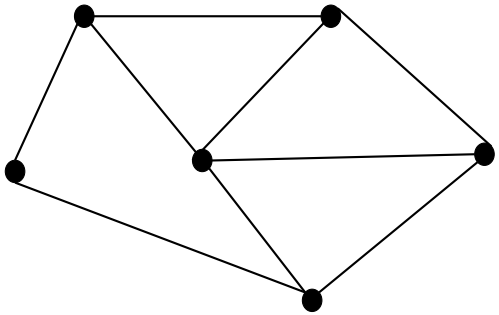
Graphs

10/9/24

Graphs

- (V, E) = set of vertices V and set of edges E
  - Edges can be directed or undirected
  - Vertices and edges may have extra data attached to them



Applications

- Geographic information
- Computer networks
- Web graph, citation graph
- Facebook graph
- Precedence relationships
- State transitions

Depth-First Search (DFS)

```
mark origin as visited;
vertexStack.push(origin);
while(!vertexStack.isEmpty()) {
        topVertex = vertexStack.peek();
        if(topVertex has an unvisited neighbor) {
                nextNeighbor = next unvisited neighbor of topVertex;
                mark nextNeighbor as visited;
                vertexStack.push(nextNeighbor);
        } else
                vertexStack.pop();
}
```

Recursive DFS

```
DFS(Vertex v) {
        while(v has another neighbor) {
                u = next neighbor of v;
                if(u is unmarked) {
                        mark u;
                        DFS(u);
                }
        }
}
```

Breadth-First Search (BFS)

```
mark origin as visited;
vertexQueue.enqueue(origin);
while(!vertexQueue.isEmpty()) {
        frontVertex = vertexQueue.dequeue();
        while(frontVertex has a neighbor) {
                nextNeighbor = next neighbor of frontVertex;
                if(nextNeighbor is not visited) {
                        mark nextNeighbor as visited;
                        vertexQueue.enqueue(nextNeighbor);
                }
        }
}
```

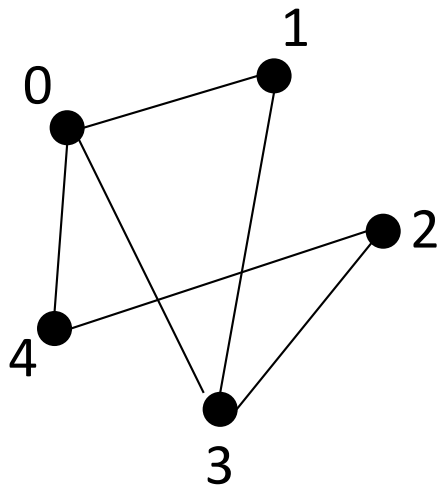
Breadth-First Search (BFS)

```
mark origin as visited;
vertexQueue.enqueue(origin);
while(!vertexQueue.isEmpty()) {
        frontVertex = vertexQueue.dequeue();
        while(frontVertex has a neighbor) {
                nextNeighbor = next neighbor of frontVertex;
                if(nextNeighbor is not visited) {
                        mark nextNeighbor as visited;
                        vertexQueue.enqueue(nextNeighbor);
                }
        }
}
```

How can this be used to find a path from origin to a specific other vertex t?

Graph representation: Adjacency list

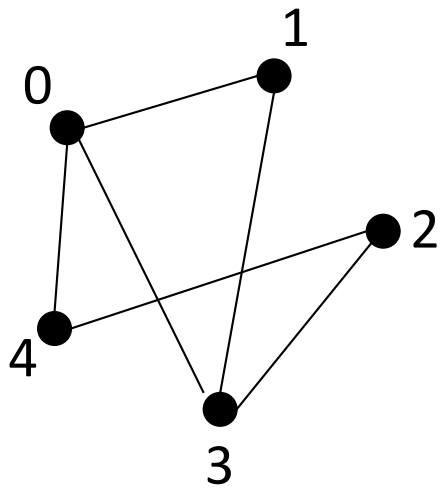
List of vertices, each with a list of incident edges (or neighboring vertices)



Graph representation: Adjacency matrix

- Cell i,j is 1 if vertices i and j are adjacent

		from 0	1	2	3	4
	0	0	1	0	1	1
	1	1	0	0	1	0
to	2	0	0	0	1	1
	3	1	1	1	0	0
	4	1	0	1	0	0

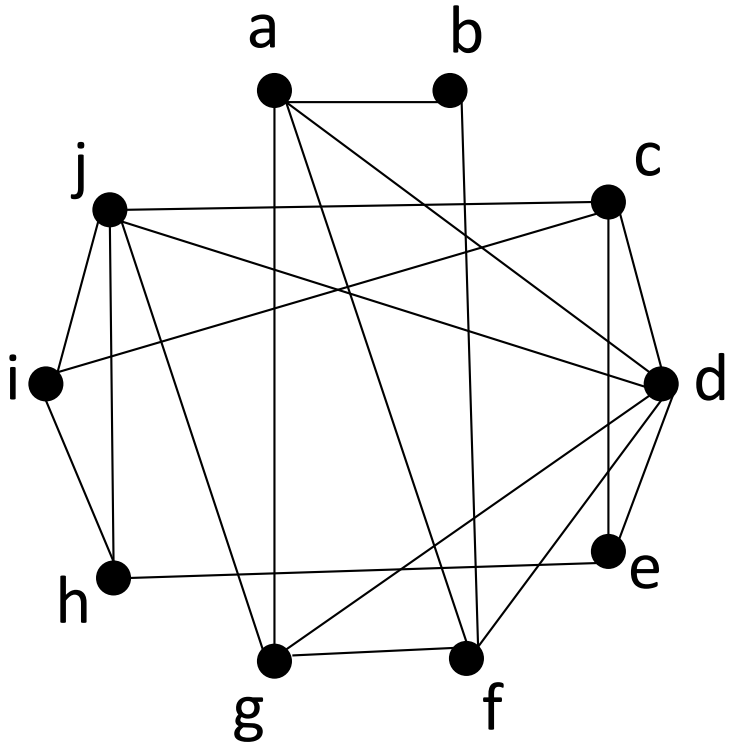


Graph coloring

- Assign each vertex a “color” so that no pair of adjacent vertices have the same color
  - Colors are typically numbers 1, 2, etc
- Goal is to use as few colors as possible

What is the fewest number of colors that can be used to color the graph below?

A. 2
B. 3
C. 4
D. 5
E. 6



What is the fewest number of colors that can be used to color the graph below?

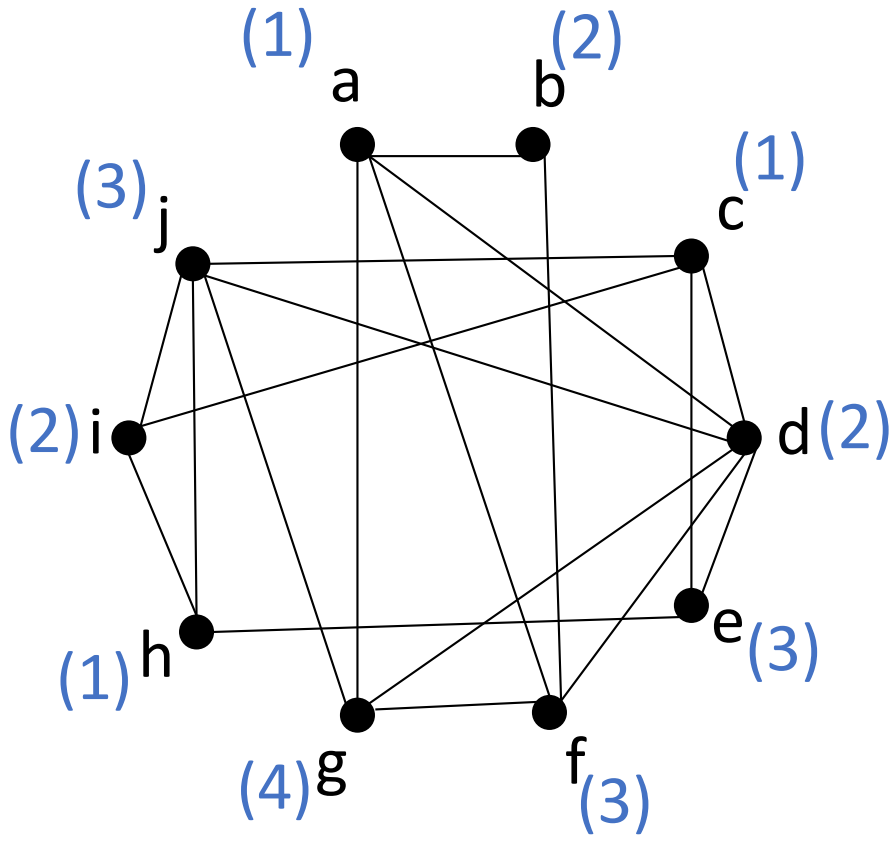
A. 2

B. 3

C. 4

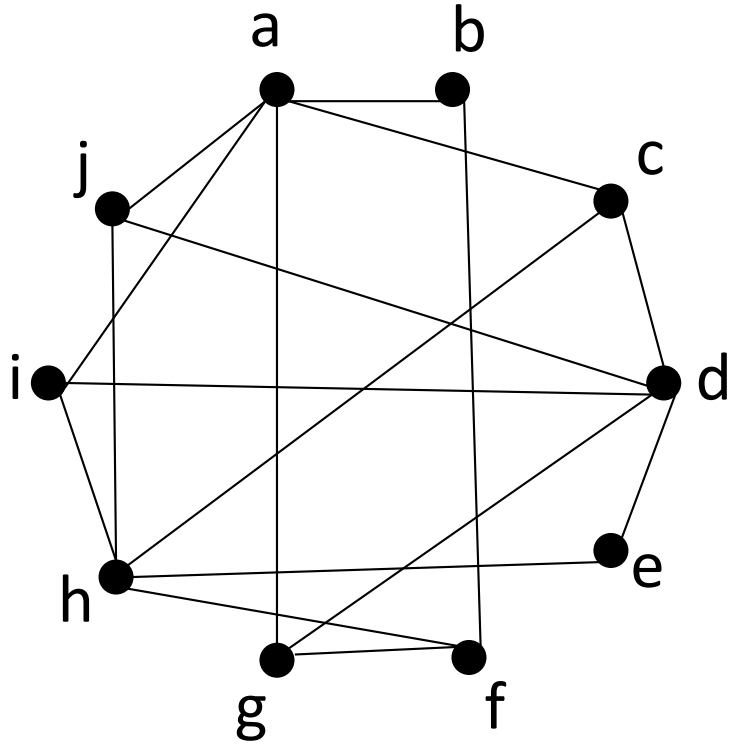
D. 5

E. 6



Is this graph 2-colorable?

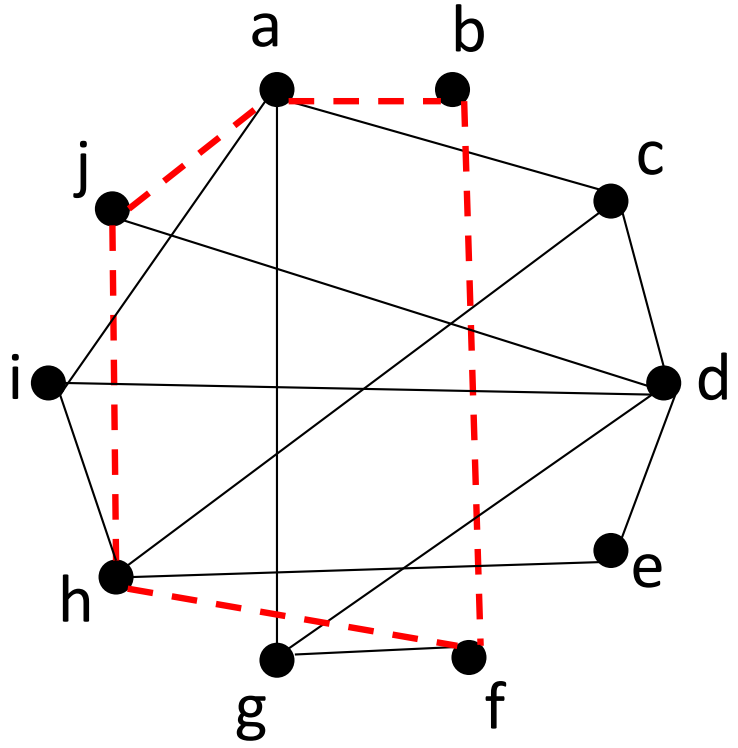
A. Yes
B. No



Is this graph 2-colorable?

A. Yes

B. No



Give an algorithm to determine if a graph is 2-colorable

Consider the adjacency matrix representation of a directed graph. Give an O(n) algorithm to determine if it has a *universal sink*, a vertex with incoming edges from every other vertex and no outgoing edges.