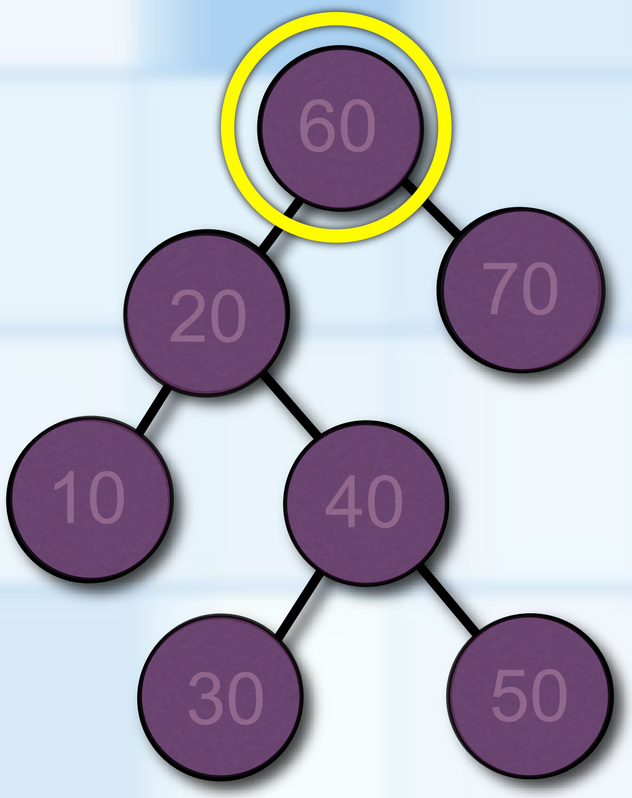
Graph Traversals and Ordering

Pearson

Depth-First Traversal

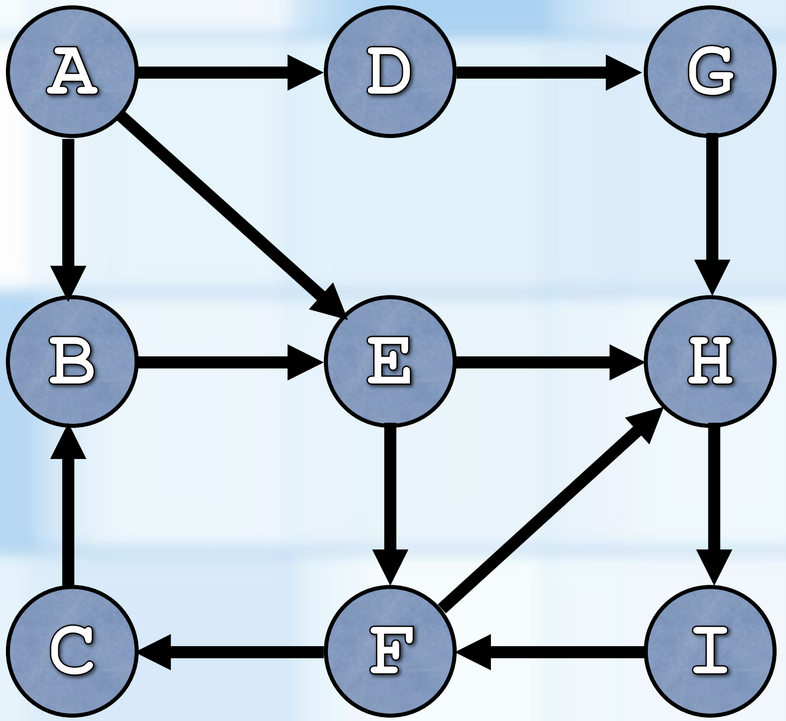
Follows a path that goes as deeply into the graph as possible before following other paths



Pearson

Depth-First Traversal

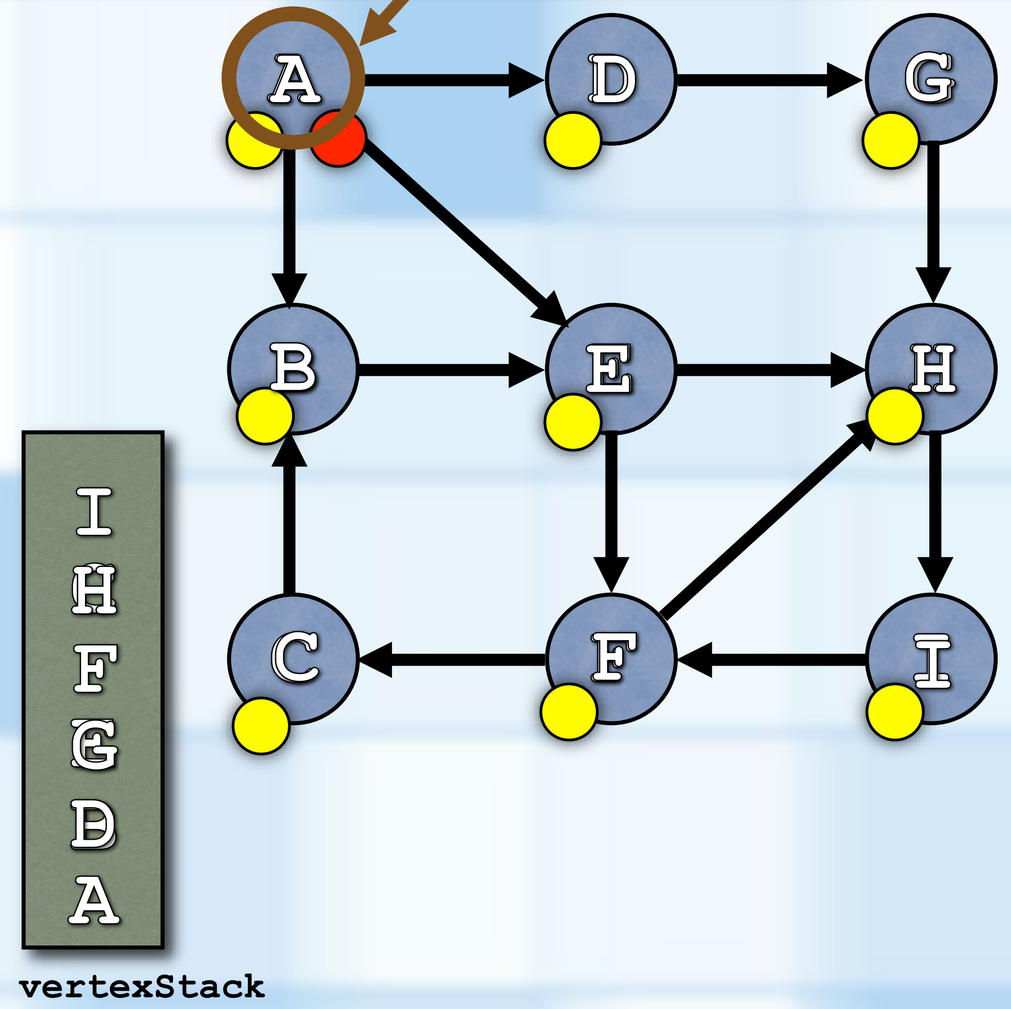
Follows a path that goes as deeply into the graph as possible before following other paths



Depth-First Traversal

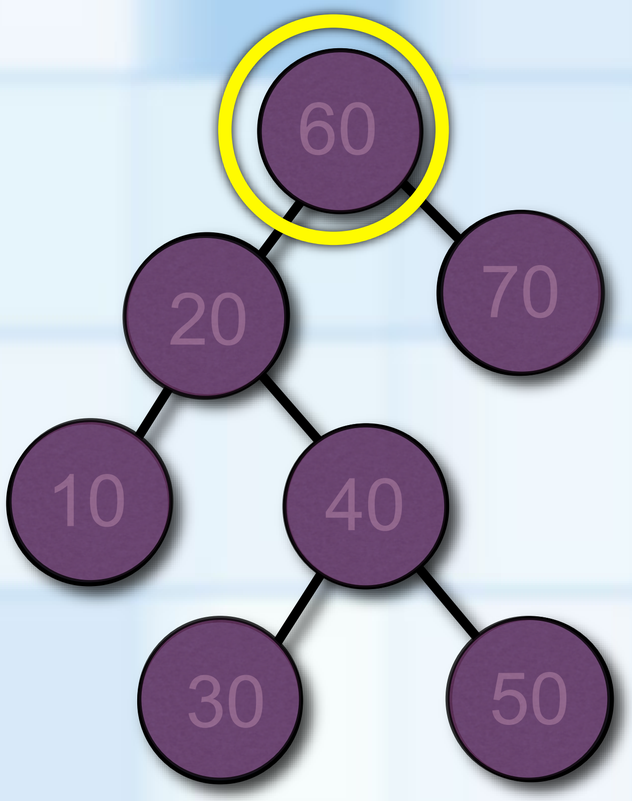
Algorithm getDepthFirstTraversal(originVertex)

```
traversalOrder = queue for the resulting traversal order
vertexStack = stack to hold vertices as they are visited

Mark originVertex as visited
traversalOrder.enqueue(originVertex)
vertexStack.push(originVertex)

while (!vertexStack.isEmpty())
{
   topVertex = vertexStack.peek()
   if (topVertex has an unvisited neighbor)
   {
       nextNeighbor = next unvisited neighbor of topVertex
        Mark nextNeighbor as visited
        traversalOrder.enqueue(nextNeighbor)
        vertexStack.push(nextNeighbor)
   }
   else
       vertexStack.pop()
}
return traversalOrder
```



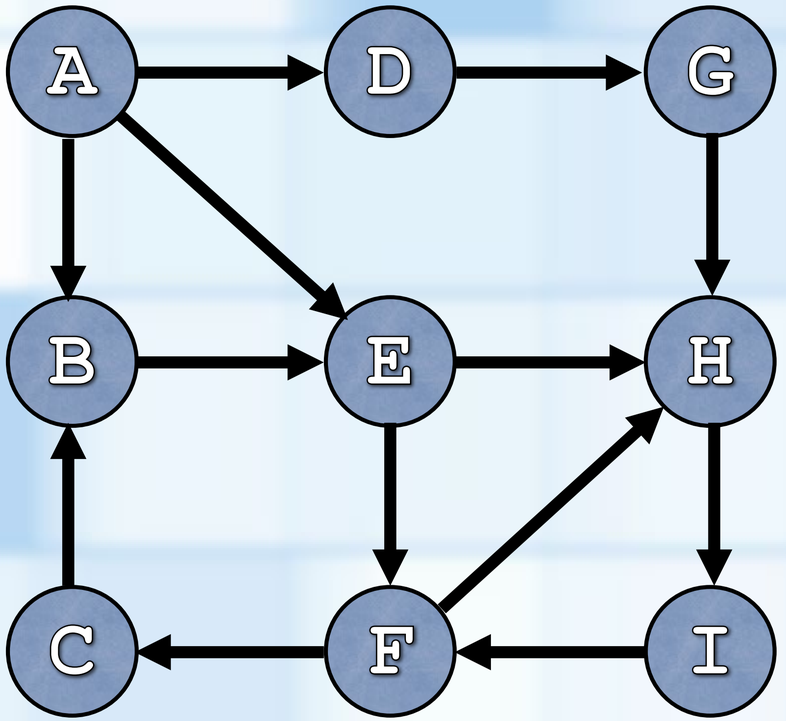
Breadth-First Traversal

Follows a path that explores an entire level before moving to the next level



Breadth-First Traversal

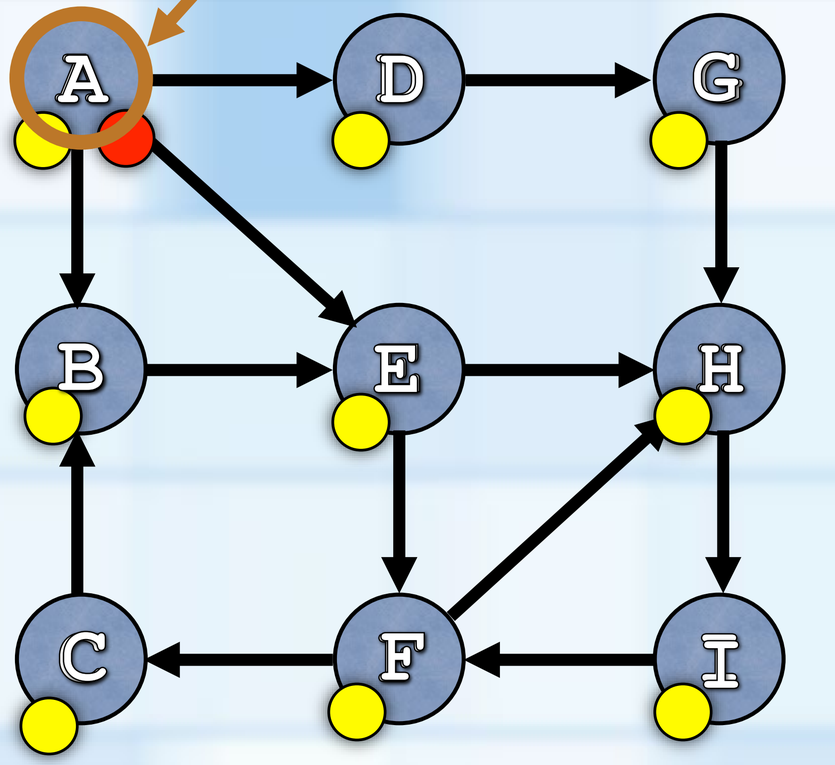
Follows a path that explores an entire level before moving to the next level



Pearson

Breadth-First Traversal

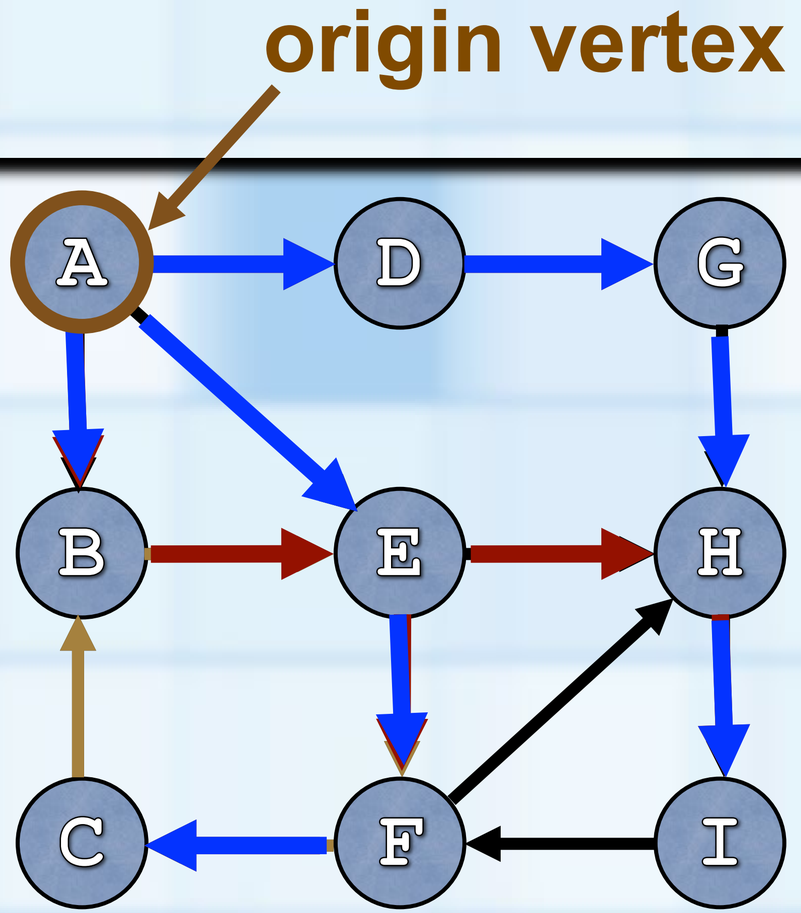
```
Algorithm getBreadthFirstTraversal(originVertex)
traversalOrder = queue for the resulting traversal order vertexQueue
= queue to hold vertices as they are visited

Mark originVertex as visited
traversalOrder.enqueue(originVertex)
vertexQueue.enqueue(originVertex)

while (!vertexQueue.isEmpty())
{
   frontVertex = vertexQueue.peekFront()
   vertexQueue.dequeue()
   while (frontVertex has a neighbor)
   {
       nextNeighbor = next neighbor of frontVertex
      if (nextNeighbor is not visited)
      {
          Mark nextNeighbor as visited
          traversalOrder.enqueue(nextNeighbor)
          vertexQueue.enqueue(nextNeighbor)
      }
   }
}
return traversalOrder
```



vertexQueue: A B E D F H G C I

traversalOrder:

Topological Order



- **Order of vertices in a directed graph without cycles**
  - Topological ordering is not possible with a directed graph with cycle