Breadth-First Search Algo

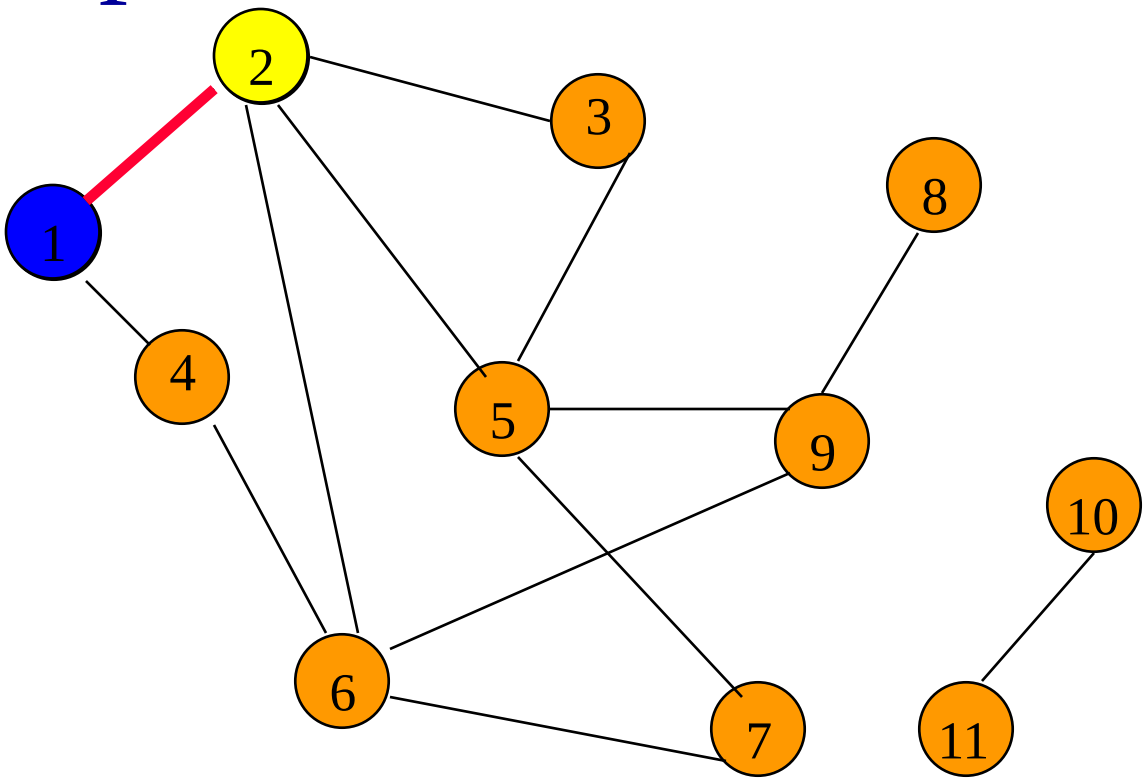
- Searching in nonempty graph beginning at a given vertex

getBreadthFirstSearch(originVertex)
Mark originVertex *as visited*
OutputQueue.enqueue(originVertex)
WorkingQueue.enqueue(originVertex)
while (! WorkingQueue.isEmpty())
{ frontVertex = WorkingQueue.dequeue()
while (frontVertex *has an unvisited neighbor*)
{ nextNeighbor = *next unvisited neighbor of* frontVertex
Mark nextNeighbor *as visited*
OutputQueue.enqueue(nextNeighbor)
WorkingQueue.enqueue(nextNeighbor)
} }
return OutputQueue

Depth-First Search

```
depthFirstSearch(v)
{
   Label vertex v as reached.
   for (each unreached vertex u
                              adjacenct from v)
      depthFirstSearch(u);
}
```

Depth-First Search Example

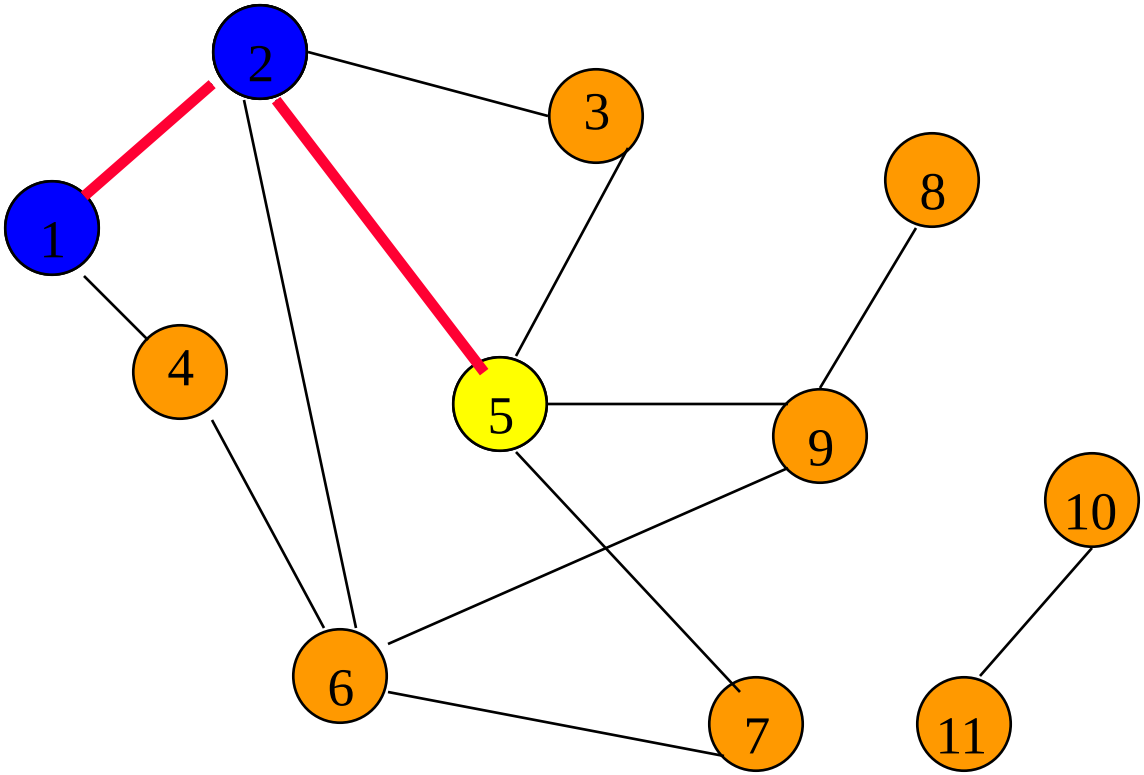


Start search at vertex 1.

Label vertex 1 and do a depth first search from either 2 or 4.

Suppose that vertex 2 is selected.

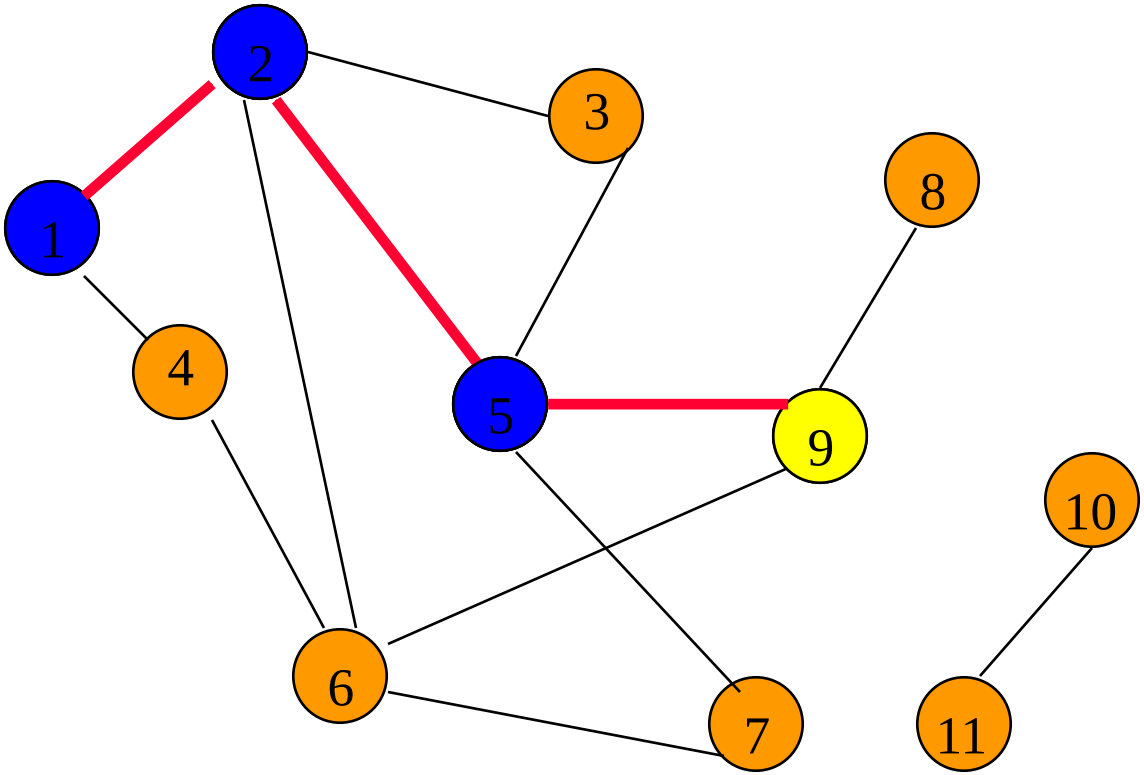
Depth-First Search Example



Label vertex 2 and do a depth first search from either 3, 5, or 6.

Suppose that vertex 5 is selected.

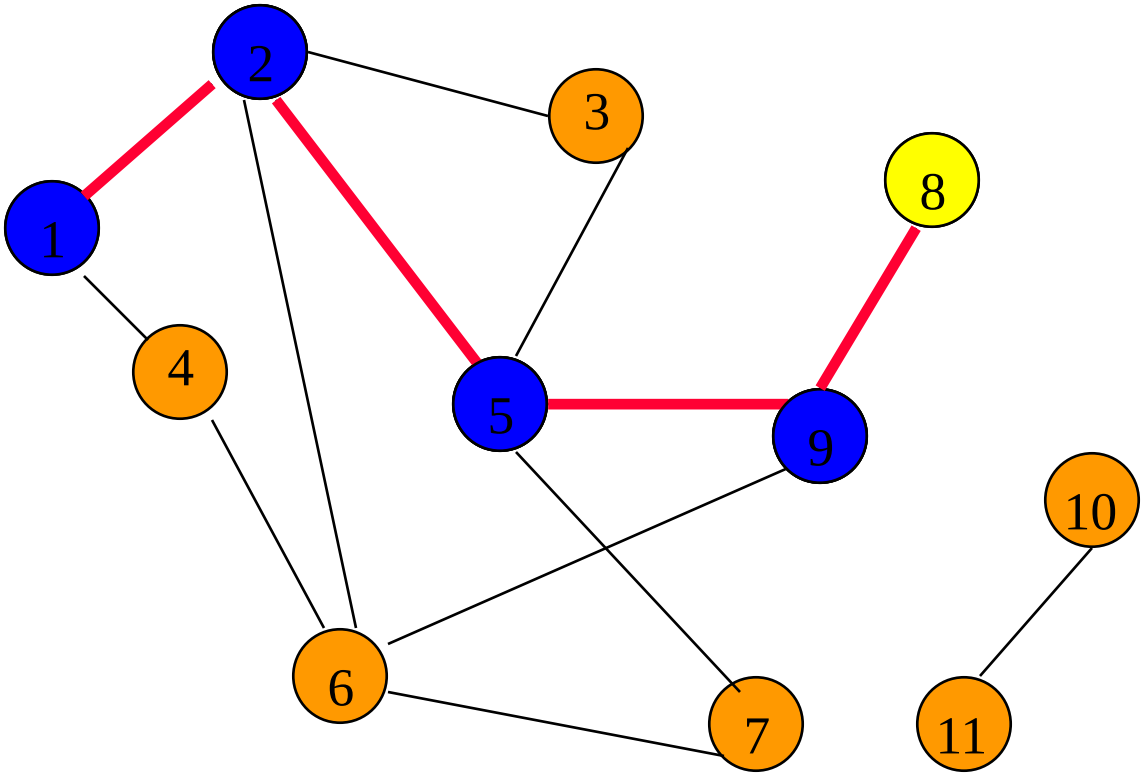
Depth-First Search Example



Label vertex 5 and do a depth first search from either 3, 7, or 9.

Suppose that vertex 9 is selected.

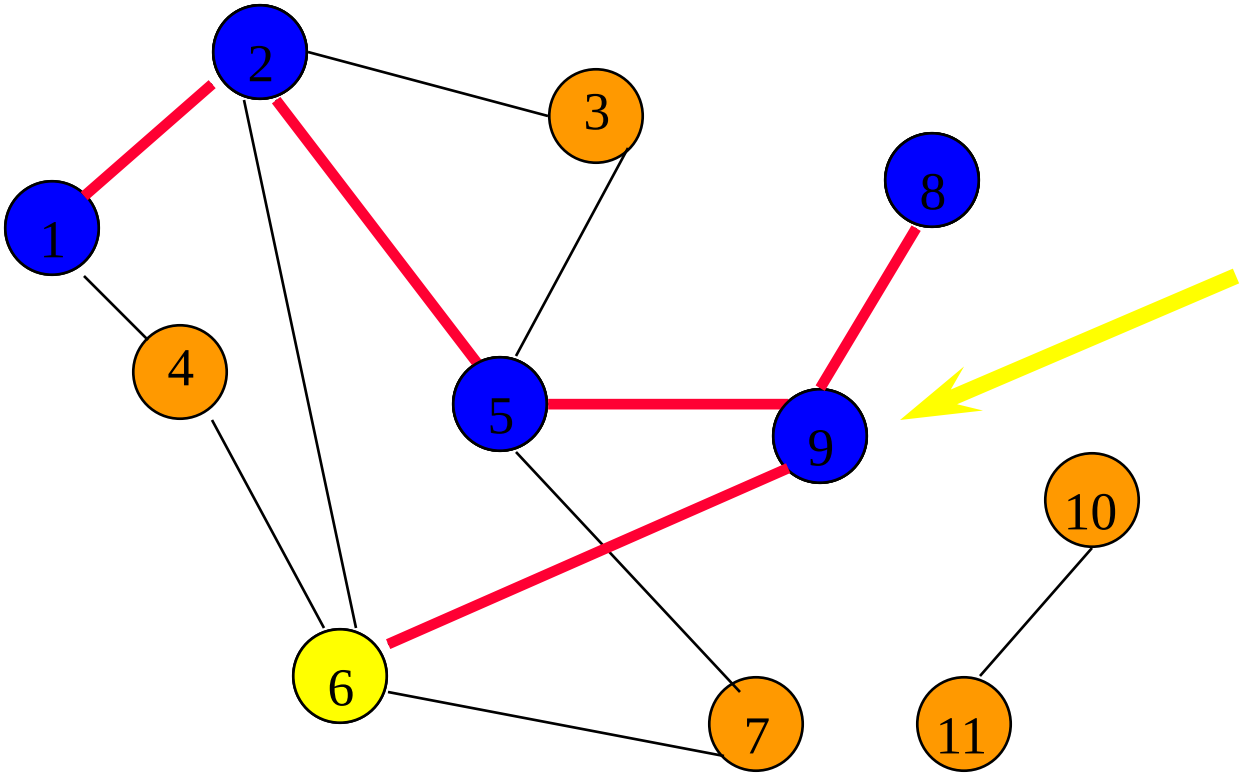
Depth-First Search Example



Label vertex 9 and do a depth first search from either 6 or 8.

Suppose that vertex 8 is selected.

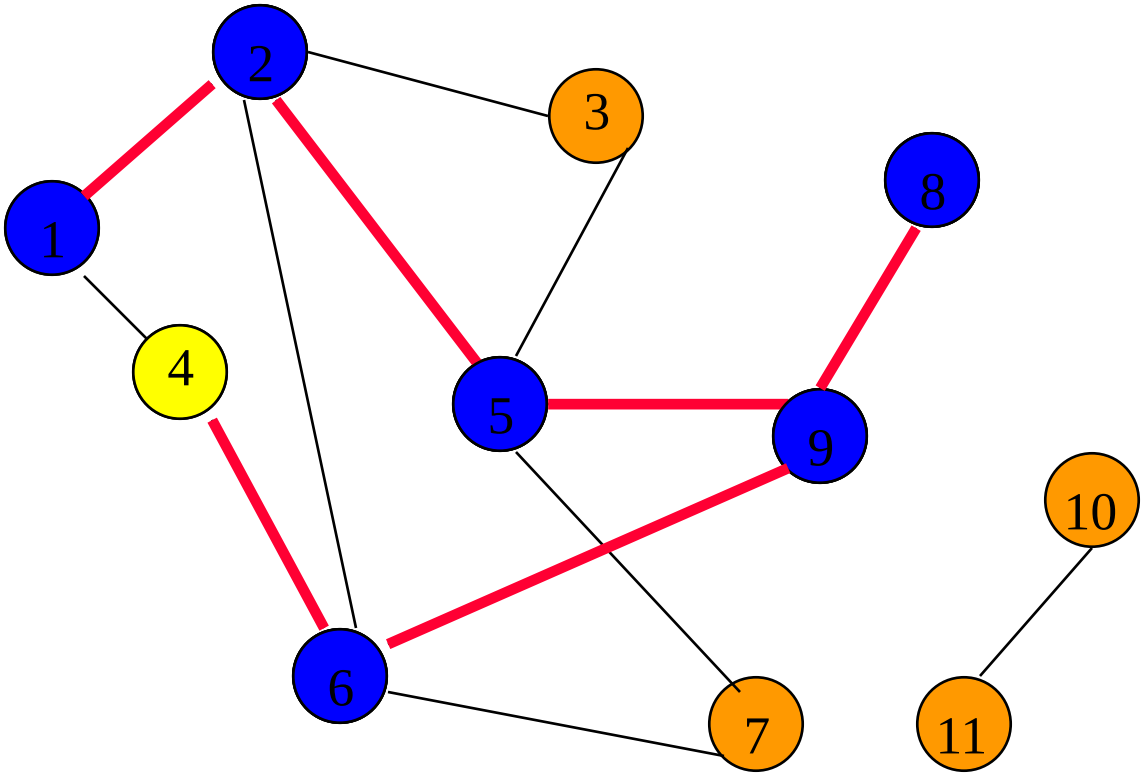
Depth-First Search Example



Label vertex 8 and return to vertex 9.

From vertex 9 do a dfs(6).

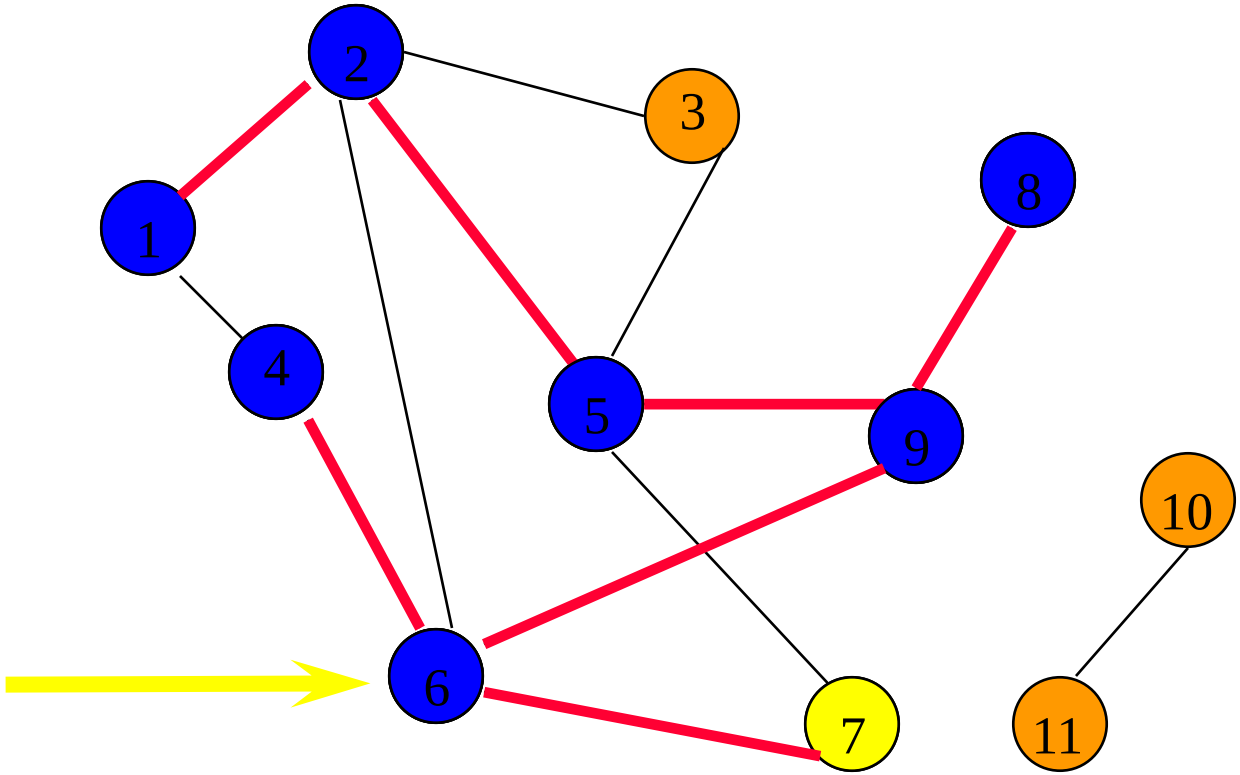
Depth-First Search Example



Label vertex 6 and do a depth first search from either 4 or 7.

Suppose that vertex 4 is selected.

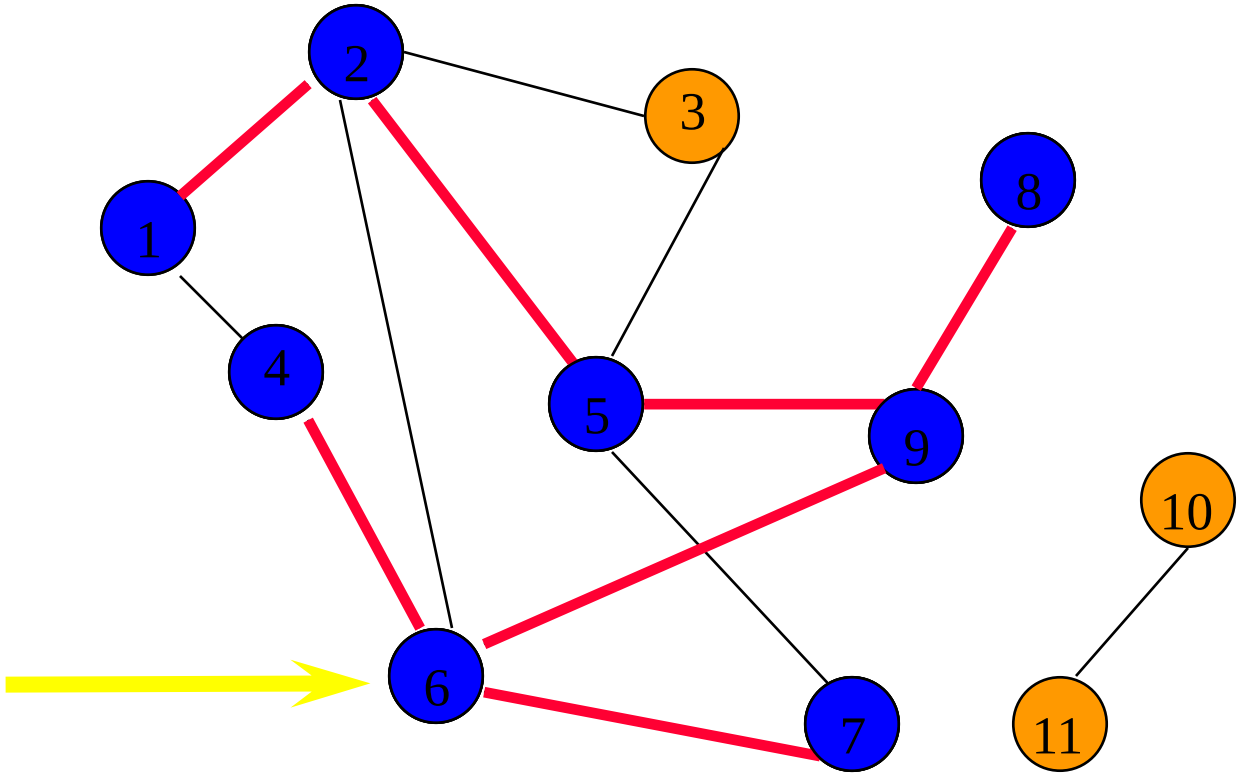
Depth-First Search Example



Label vertex 4 and return to 6.

From vertex 6 do a dfs(7).

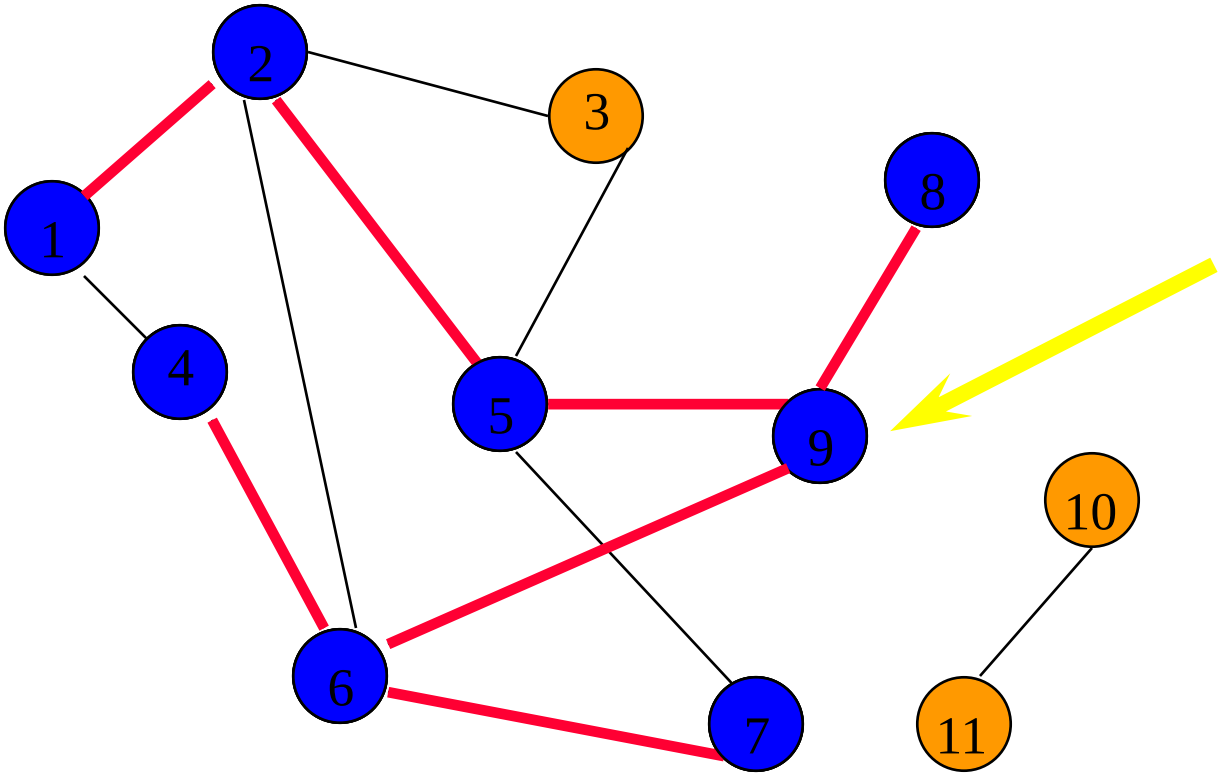
Depth-First Search Example



Label vertex 7 and return to 6.

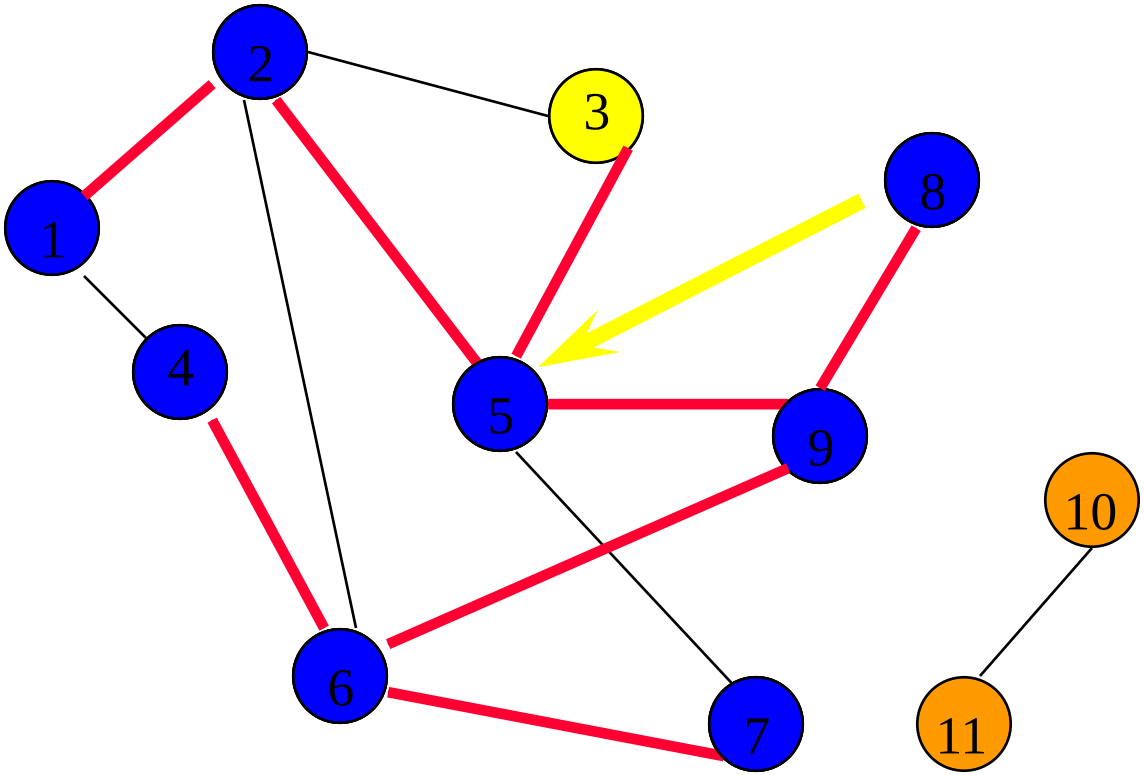
Return to 9.

Depth-First Search Example



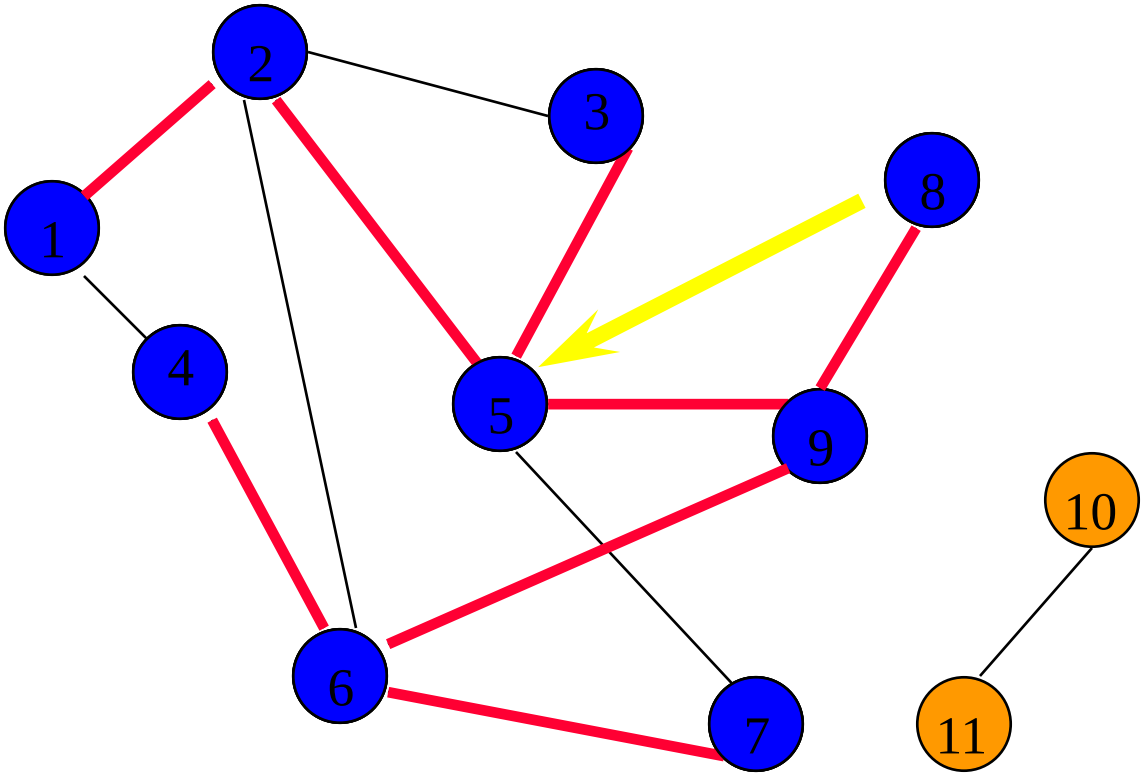
Return to 5.

Depth-First Search Example



Do a dfs(3).

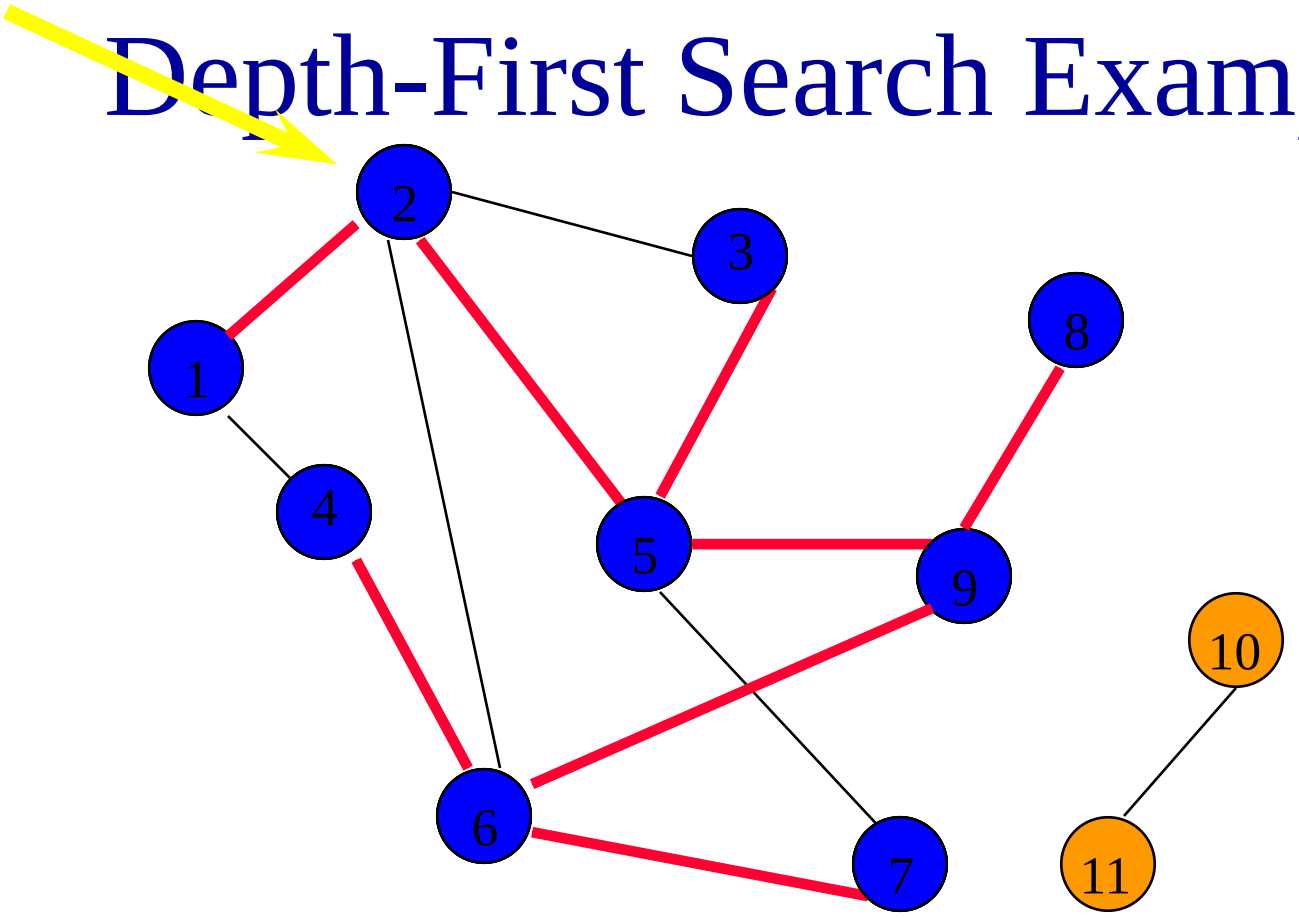
Depth-First Search Example



Label 3 and return to 5.

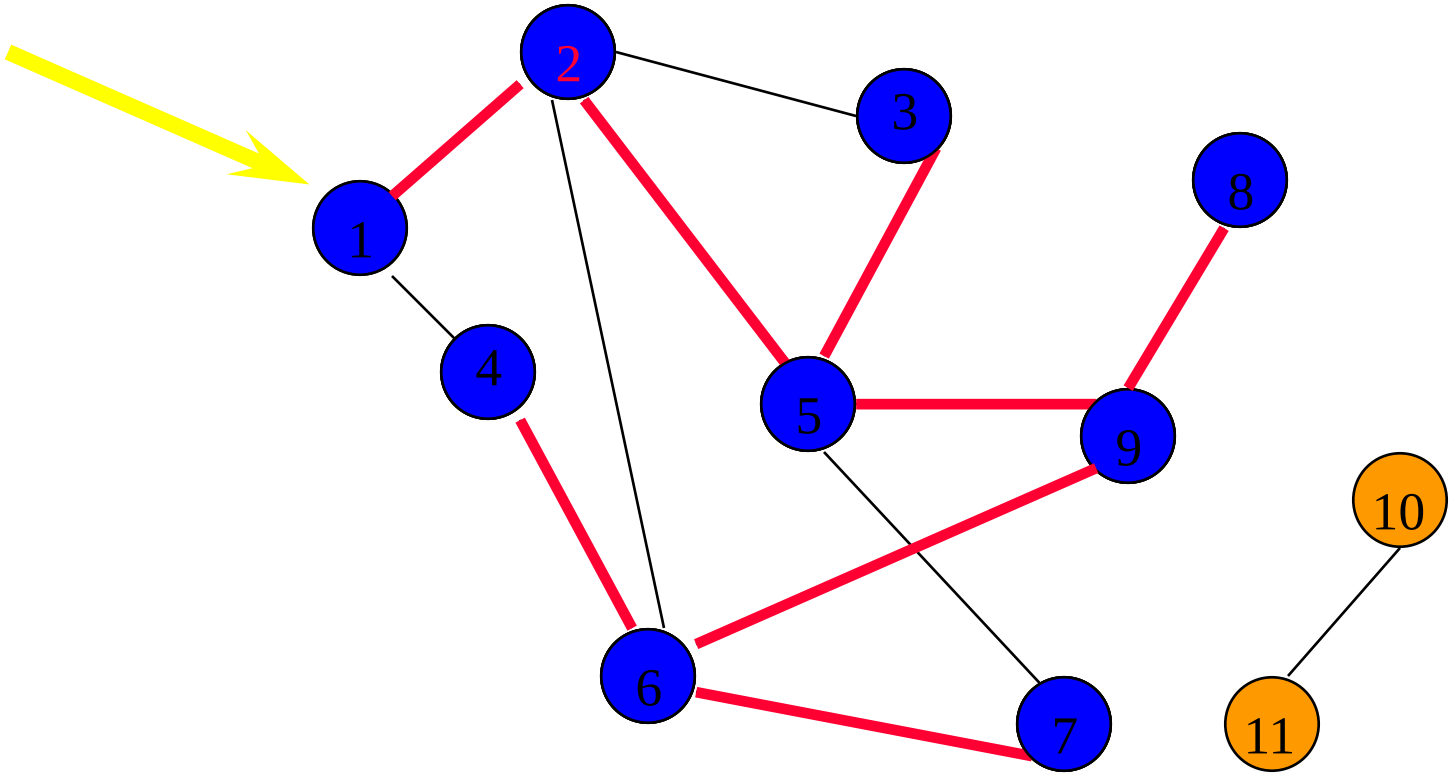
Return to 2.

Depth-First Search Example



Return to 1.

Depth-First Search Example



Return to invoking method.

Depth-First Search Properties

- Same complexity as BFS.
- Same properties with respect to path finding, connected components, and spanning trees.
- Edges used to reach unlabeled vertices define a depth-first spanning tree when the graph is connected.
- There are problems for which bfs is better than dfs and vice versa.

Depth-First Search Algo

```
Algorithm getDepthFirstSearch(originVertex)
   Mark originVertex as visited
   WorkingStack.push(originVertex)
   while (! WorkingStack.isEmpty())
   {
      topVertex = WorkingStack.peek()
      if (topVertex has an unvisited neighbor)
      {     nextNeighbor = next unvisited neighbor of topVertex
            Mark nextNeighbor as visited
                  WorkingStack.push(nextNeighbor)
      }
      else // all neighbors are visited

            WorkingStack.pop()

   }
```