

## Project 6: Supplemental Figure 1

The example below gives the resulting DFS spanning tree for the Depth-first search algorithm on the right.

This assumes that you have read sections 9.6.1 and 9.6.4 in the book.

From Project 6, question 3:

```
void Graph::DFS(Vertex v,
unsigned &time){
    v.visited = true;
    v.discovered = ++time;
    foreach Veretex w adjacent to v
        if(!w.visited)
            dfs(w, time);
    v.explored = ++time;
}
```

a)

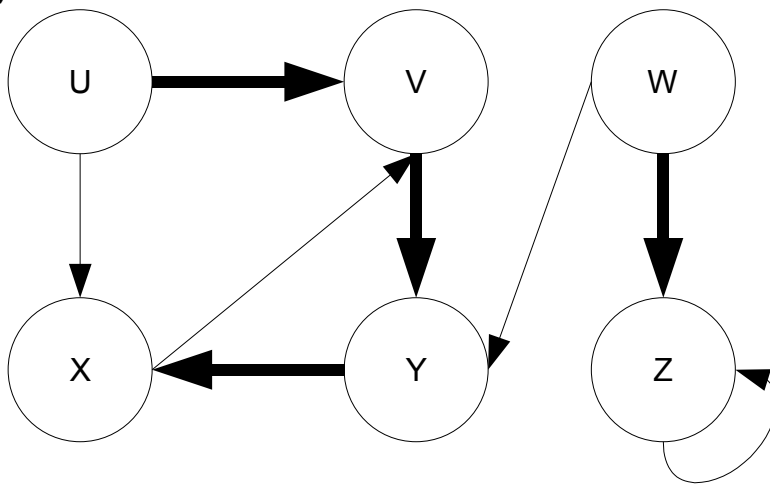


Figure (a), a simple directed graph. Note that the bold arrows indicate which edges will be used as tree edges in the DFS spanning tree, the remaining edges must be added as either forward, back or cross edges.

b)

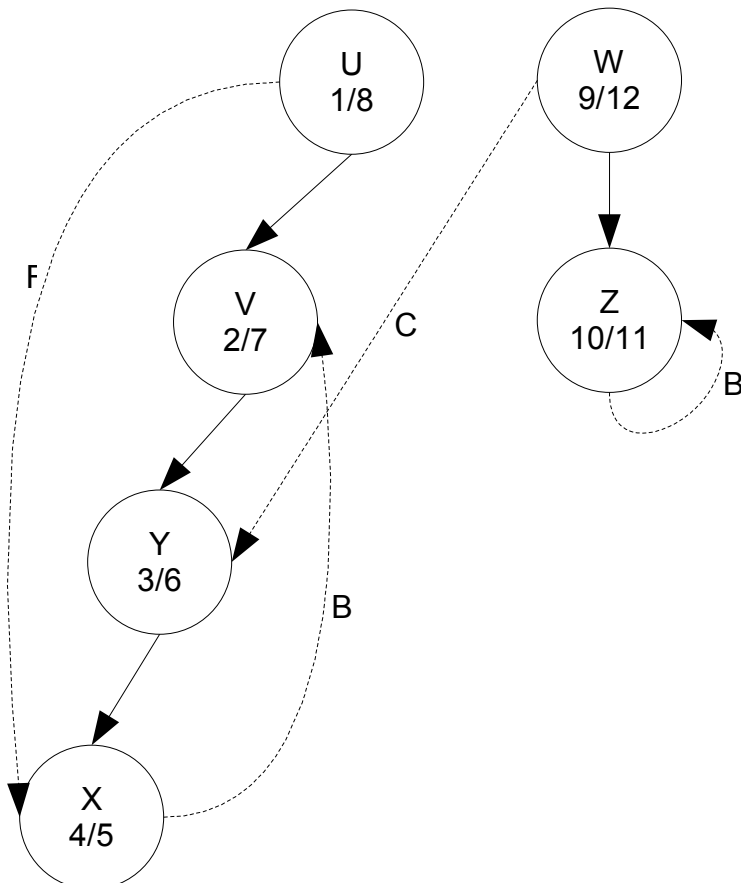


Figure (b), the resulting DFS spanning tree for the graph in figure (a). Unlabeled edges are tree edges, all other edges are labeled forward (F), back (B), or cross (c) edges.

Note that each node also has two integers assigned by our algorithm above, the first number is the order the node was discovered, the second is when the node was fully explored.

For any two given nodes, we can use these numbers, and only these numbers, to determine what type of edge is connected them.

Also for question 3, assume the the two nodes are given as an ordered pair (u,v) were u has an edge pointing to v.