Graphs, Part 5

Depth-First-Search

- One of the simplest algorithms for graph searching
- Very efficient
- Many applications
  - Topological sort
  - Strongly connected components

Depth-First Search

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```

Define the predecessor sub-graph of G as $G_\pi = (V_\pi, E_\pi)$, where

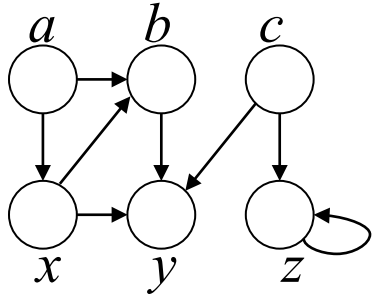
$$V_\pi = \{v \in V : v.\pi \neq NIL\} \cup \{s\}$$

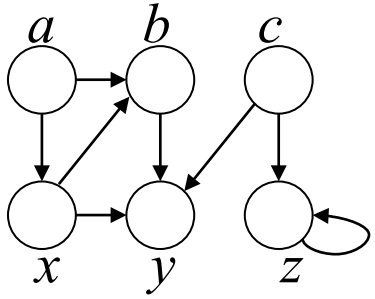
$$E_\pi = \{(v.\pi, v) : v \in V_\pi - \{s\}\}$$

G_π defines the Depth-first forest.

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



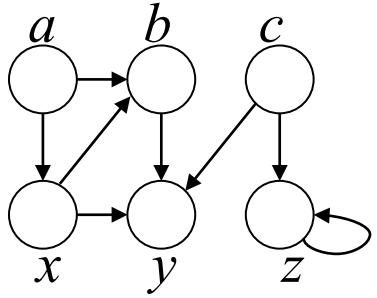


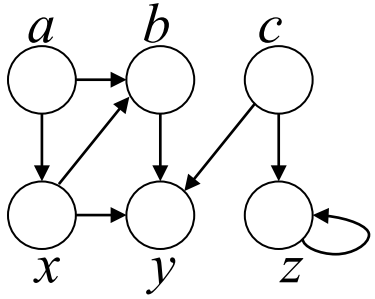
Graph G, adjacency list alphabetical

DFS(G)

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



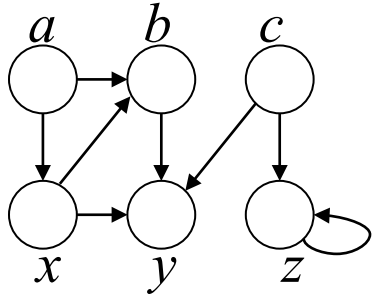


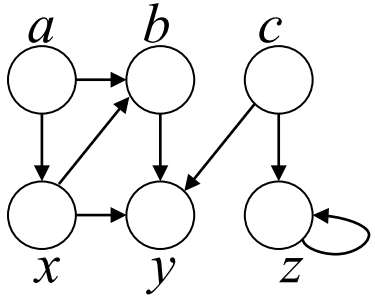
Graph G, adjacency list alphabetical

DFS(G): Lines 1-5

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



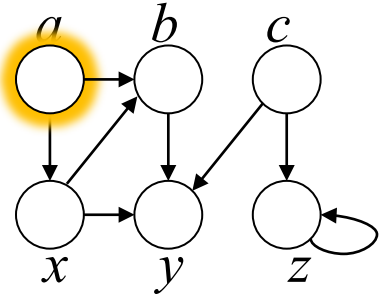


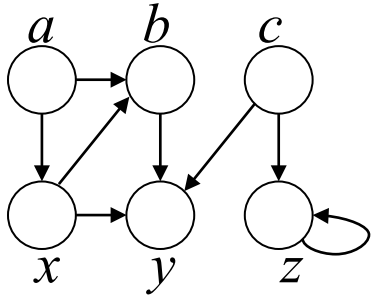
Graph G, adjacency list alphabetical

DFS(G): a is WHITE

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



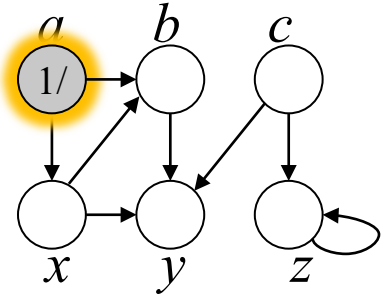


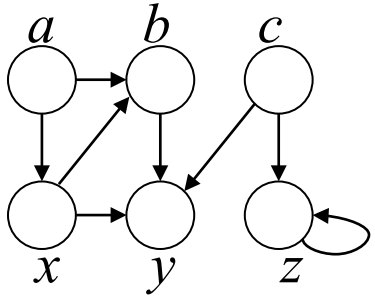
Graph G, adjacency list alphabetical

DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



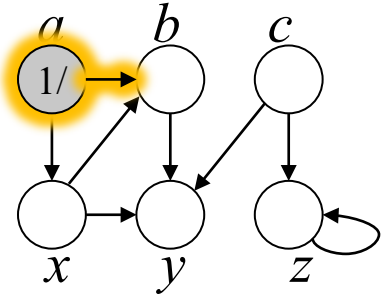


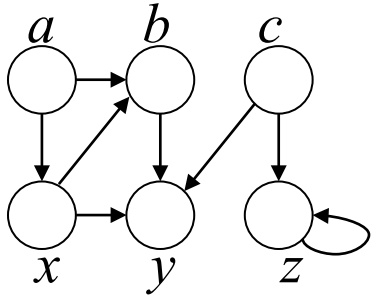
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Lines 1-3
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



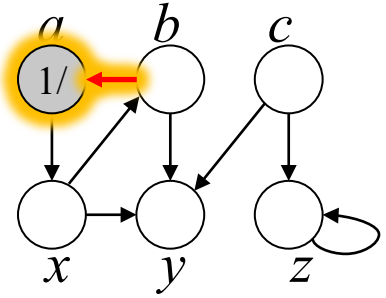


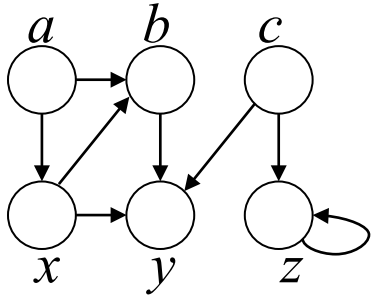
Graph G, adjacency list alphabetical

DFS-Visit(G, a): b is WHITE
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



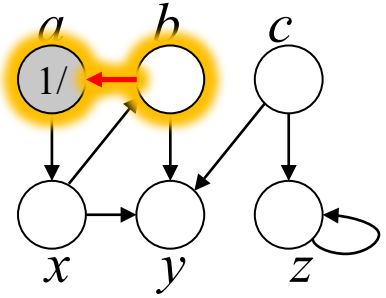


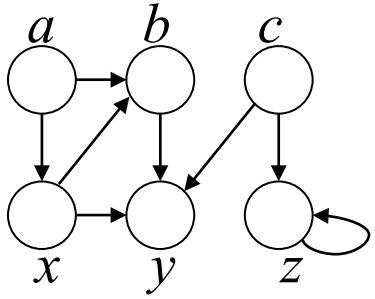
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Line 6
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



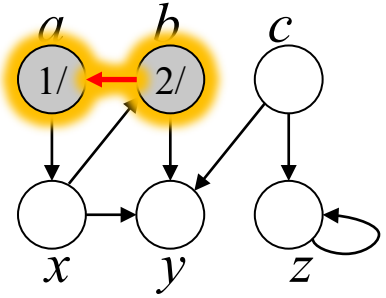


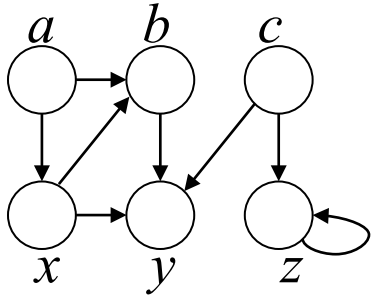
Graph G, adjacency list alphabetical

DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



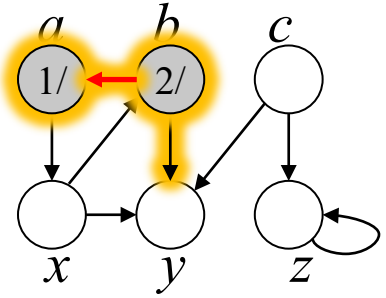


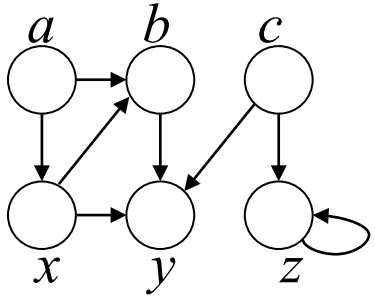
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Lines 1-3
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



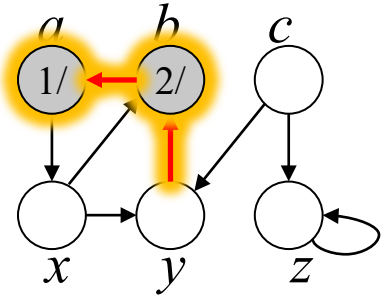


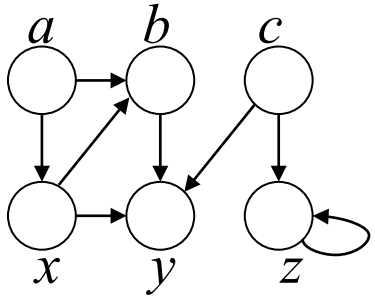
Graph G, adjacency list alphabetical

DFS-Visit(G, b): y is WHITE
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



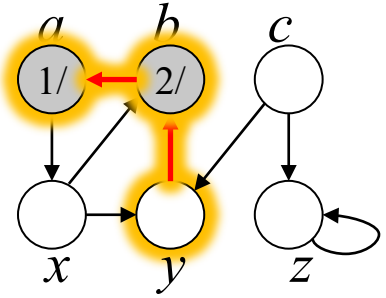


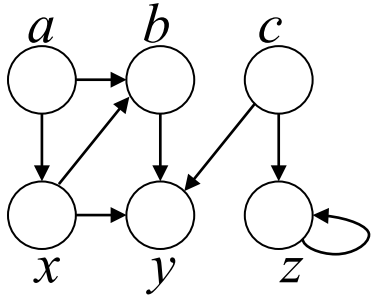
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Line 6
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



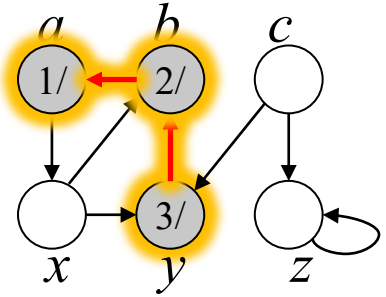


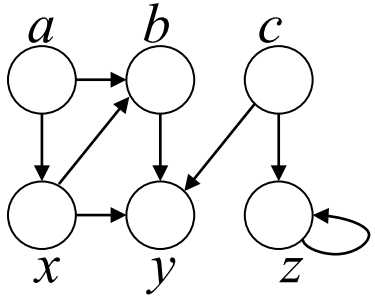
Graph G, adjacency list alphabetical

DFS-Visit(G, y)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



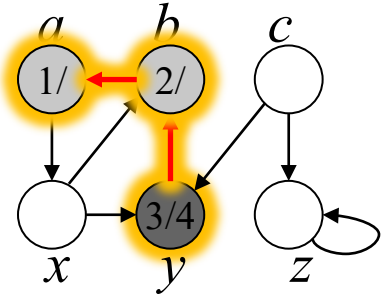


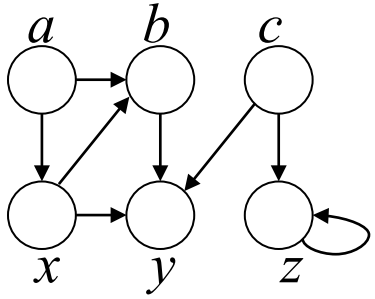
Graph G, adjacency list alphabetical

DFS-Visit(G, y): Lines 1-3
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



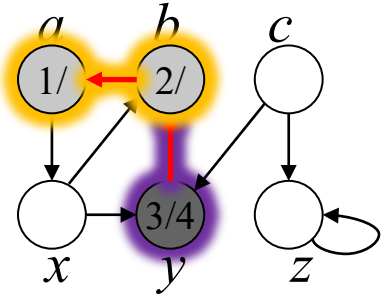


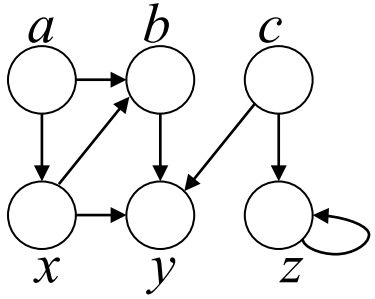
Graph G, adjacency list alphabetical

DFS-Visit(G, y): Lines 9-11
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



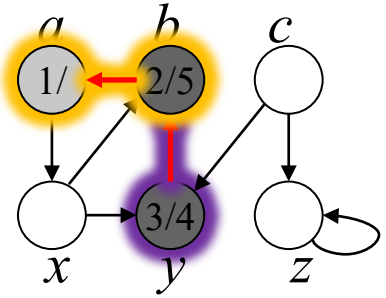


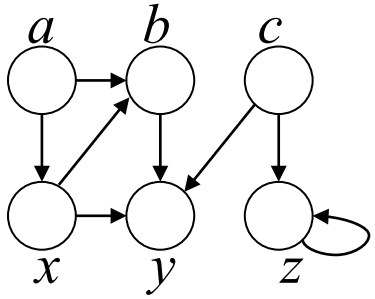
Graph G, adjacency list alphabetical

DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



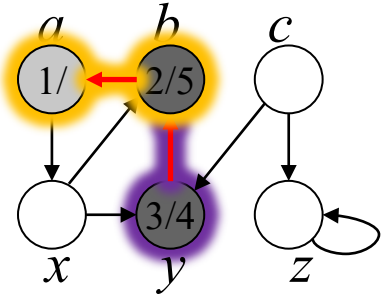


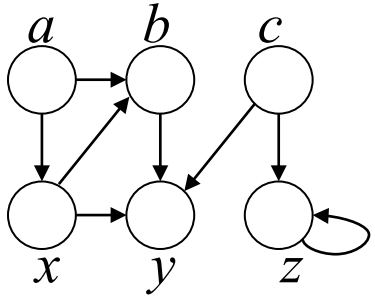
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Lines 9-11
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



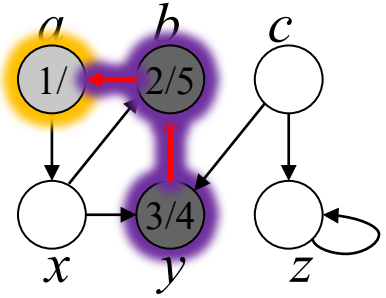


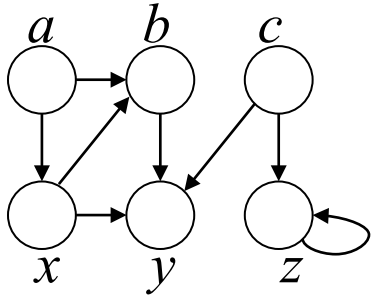
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Lines 9-11
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



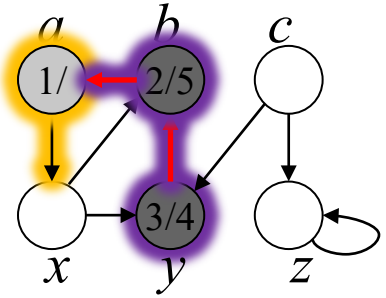


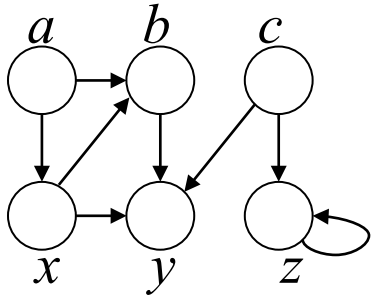
Graph G, adjacency list alphabetical

DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



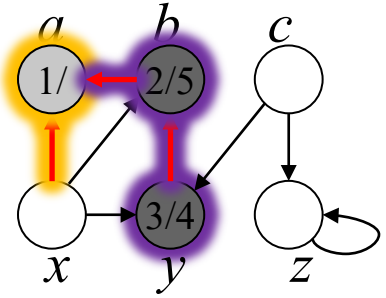


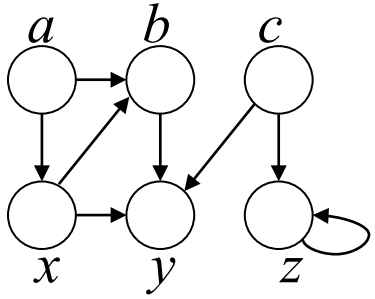
Graph G, adjacency list alphabetical

DFS-Visit(G, a): x is WHITE
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



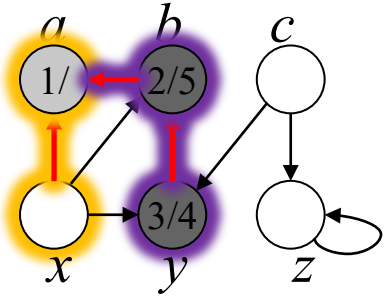


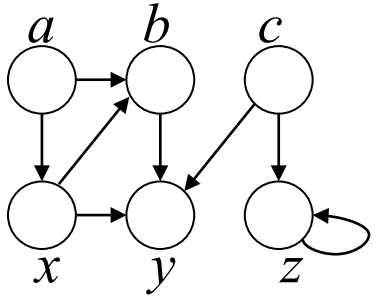
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Line 6
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



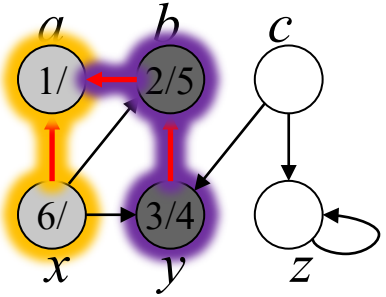


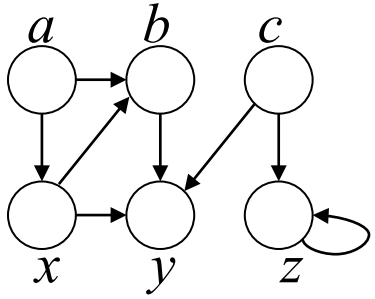
Graph G, adjacency list alphabetical

DFS-Visit(G, x)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



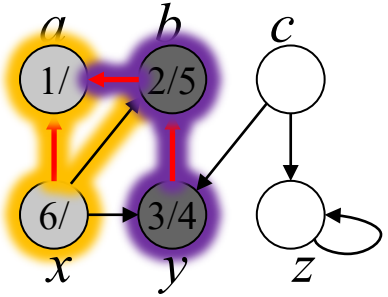


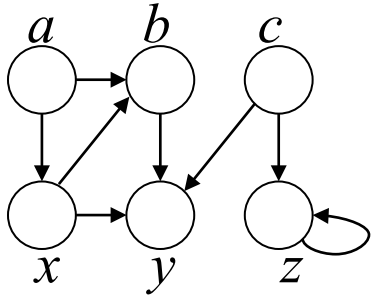
Graph G, adjacency list alphabetical

DFS-Visit(G, x): Lines 1-3
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



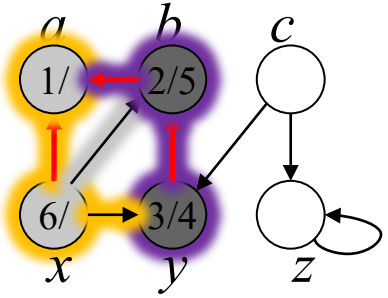


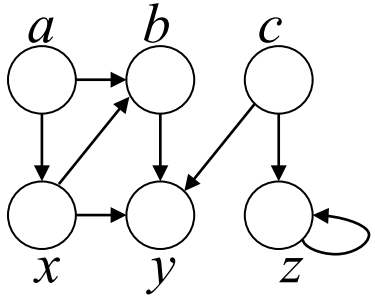
Graph G, adjacency list alphabetical

DFS-Visit(G, x): b is BLACK
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



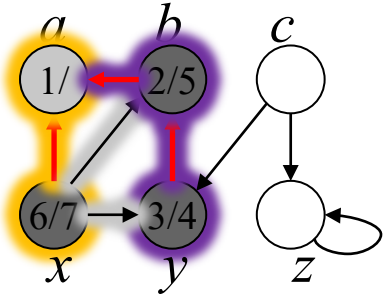


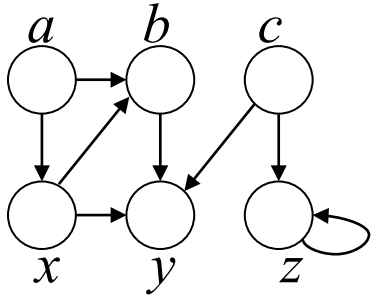
Graph G, adjacency list alphabetical

DFS-Visit(G, x): y is BLACK
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



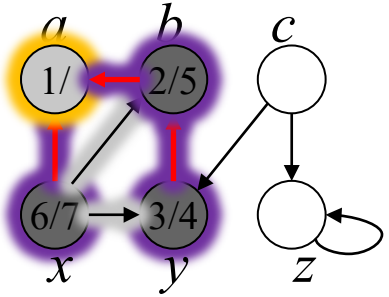


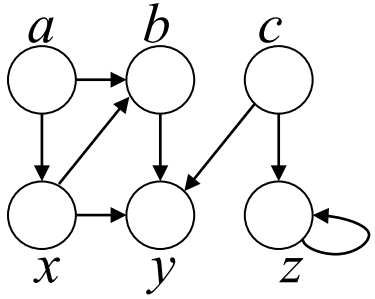
Graph G, adjacency list alphabetical

DFS-Visit(G, x): Lines 9-11
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



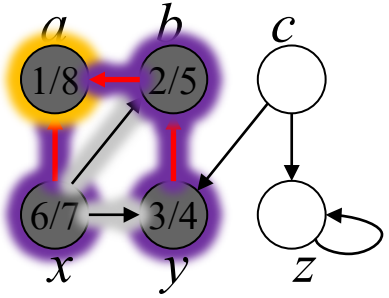


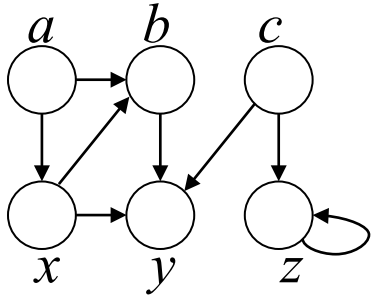
Graph G, adjacency list alphabetical

DFS-Visit(G, a)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



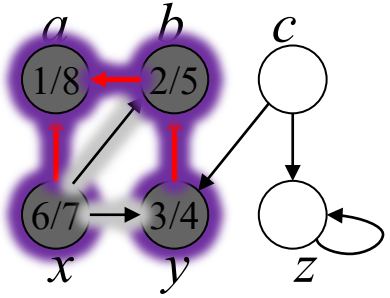


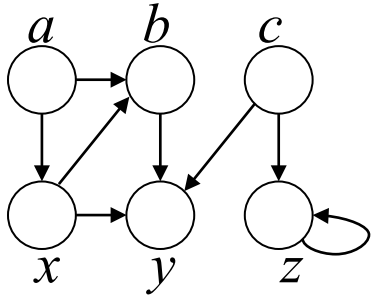
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Lines 9-11
DFS(G)

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



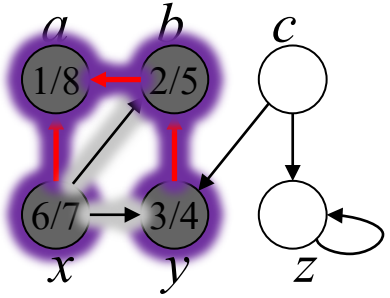


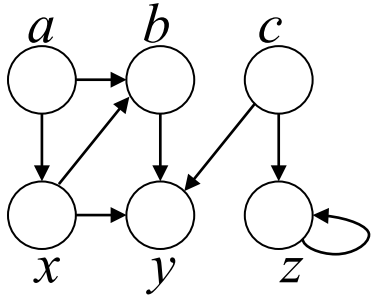
Graph G, adjacency list alphabetical

DFS(G)

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



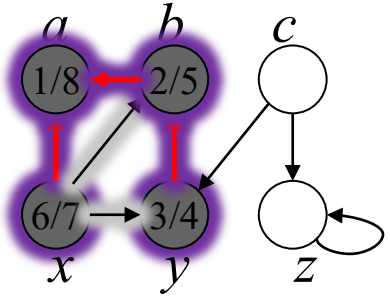


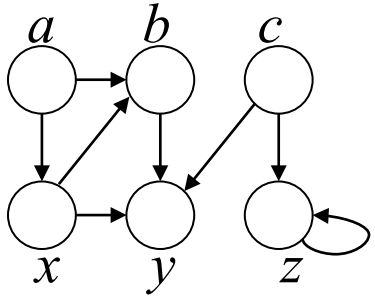
Graph G, adjacency list alphabetical

DFS(G): b is BLACK

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



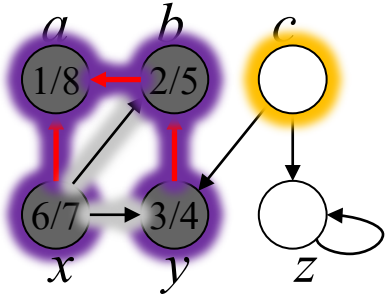


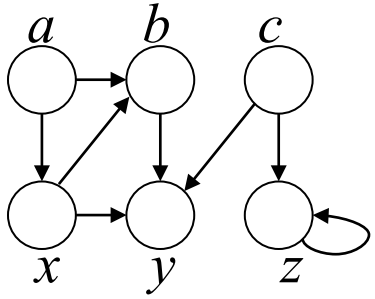
Graph G, adjacency list alphabetical

DFS(G): c is WHITE

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



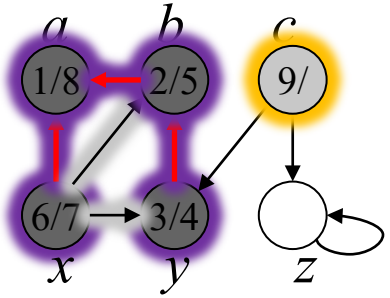


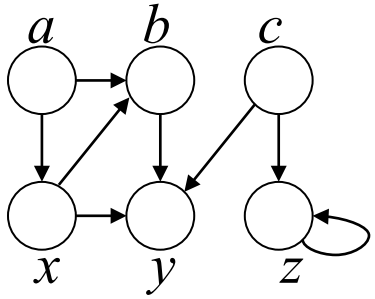
Graph G, adjacency list alphabetical

DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



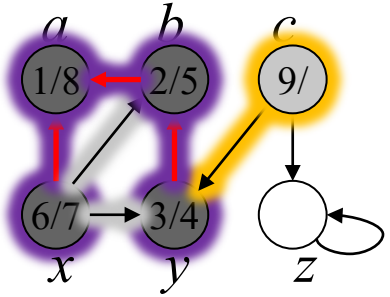


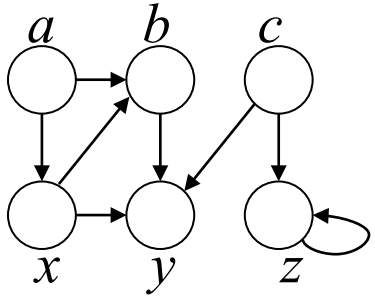
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Lines 1-3
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



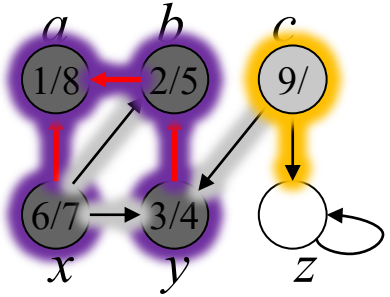


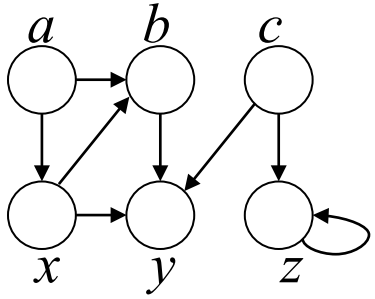
Graph G, adjacency list alphabetical

DFS-Visit(G, c): y is BLACK
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



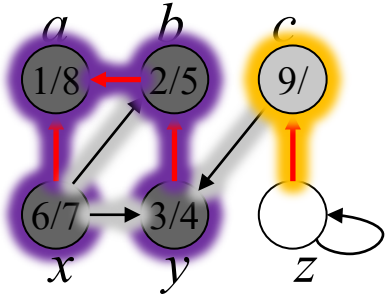


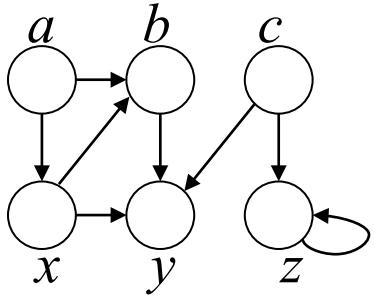
Graph G, adjacency list alphabetical

DFS-Visit(G, c): z is WHITE
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



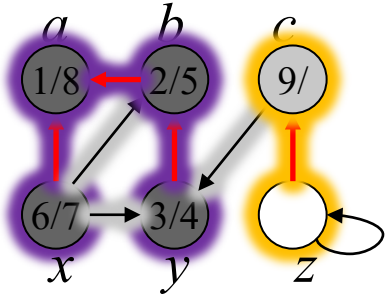


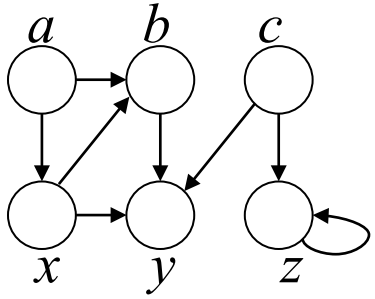
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Line 6
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



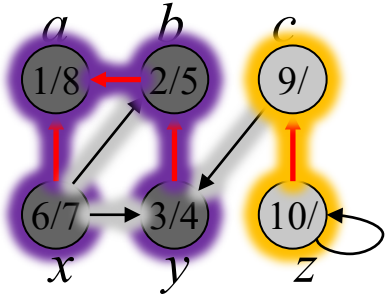


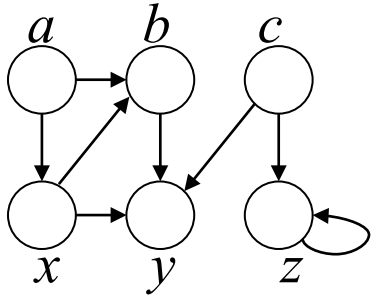
Graph G, adjacency list alphabetical

DFS-Visit(G, z)
DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



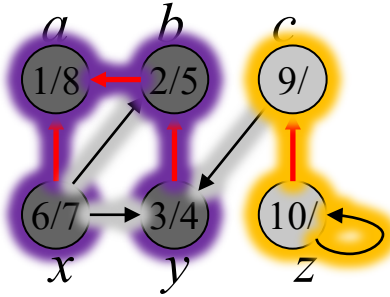


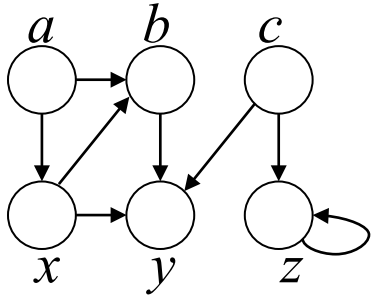
Graph G, adjacency list alphabetical

DFS-Visit(G, z): Lines 1-3
DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



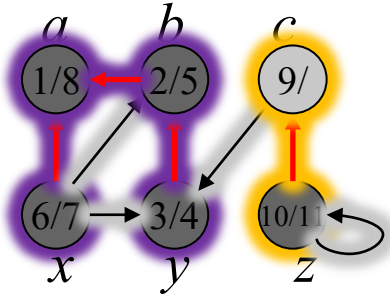


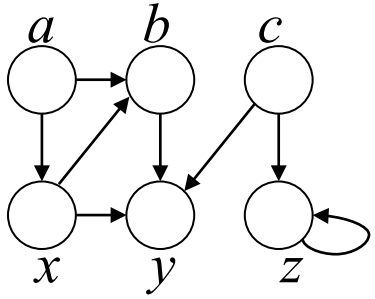
Graph G, adjacency list alphabetical

DFS-Visit(G, z): z is GRAY
DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



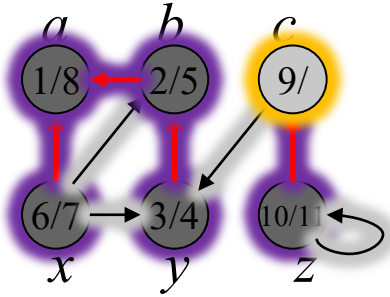


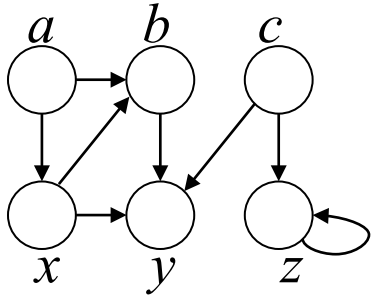
Graph G, adjacency list alphabetical

DFS-Visit(G, z): Lines 9-11
DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



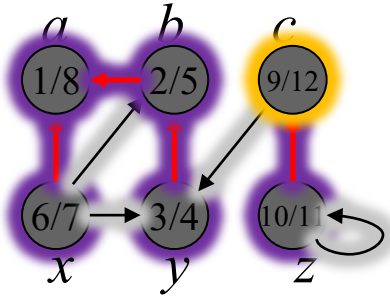


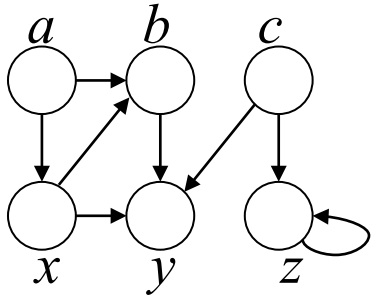
Graph G, adjacency list alphabetical

DFS-Visit(G, c)
DFS(G)

Running Example (DFS for directed graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



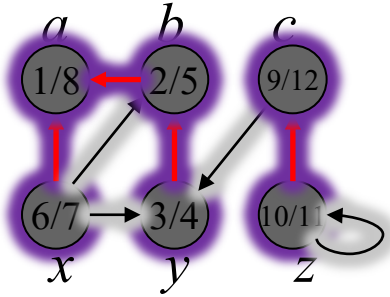


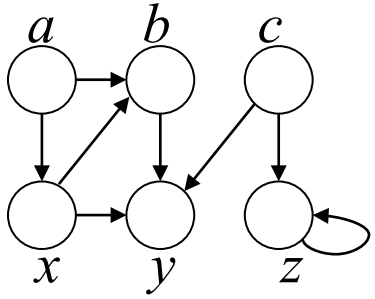
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Lines 9-11
DFS(G)

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



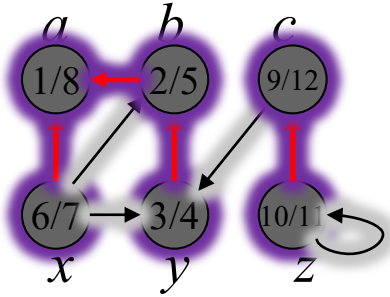


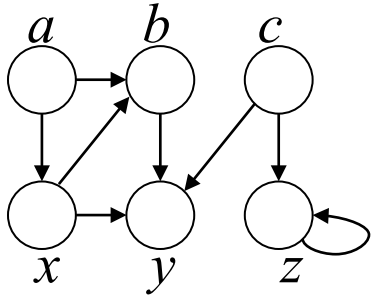
Graph G, adjacency list alphabetical

DFS(G)

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



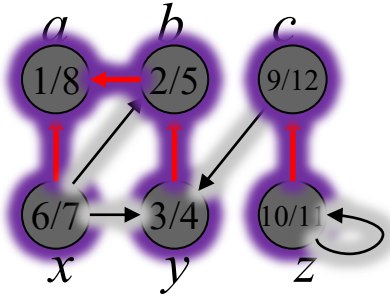


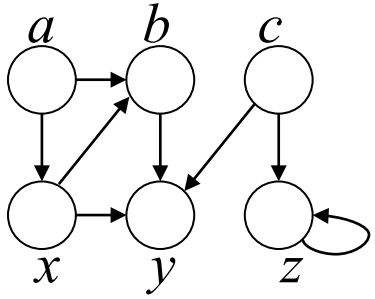
Graph G, adjacency list alphabetical

DFS(G): x is black

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



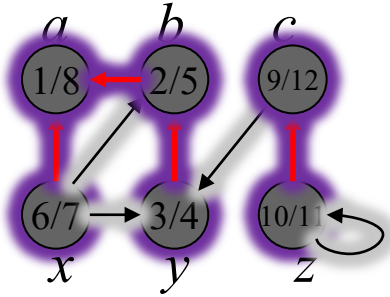


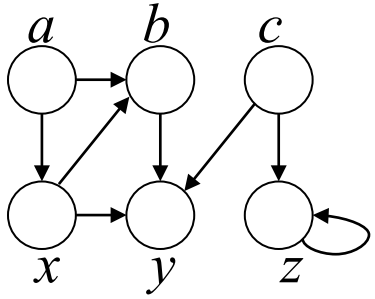
Graph G, adjacency list alphabetical

DFS(G): y is black

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



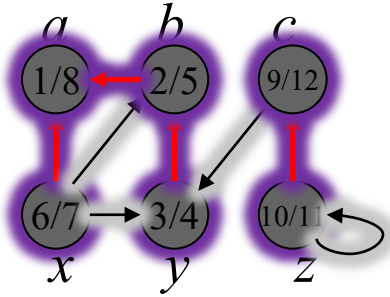


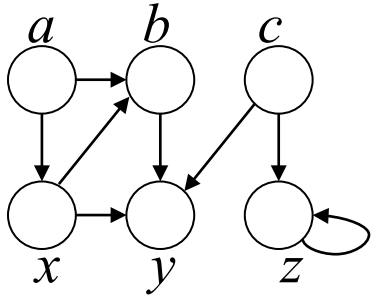
Graph G, adjacency list alphabetical

DFS(G): z is black

Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```

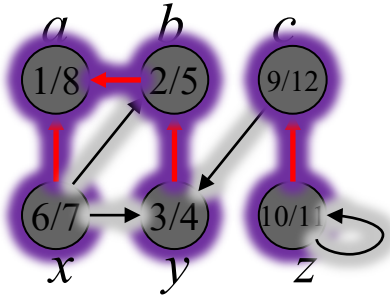


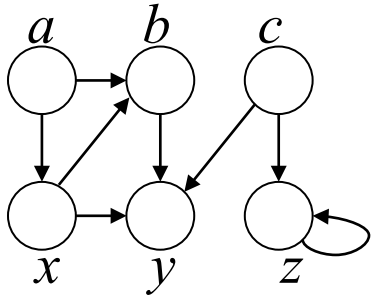


Graph G, adjacency list alphabetical

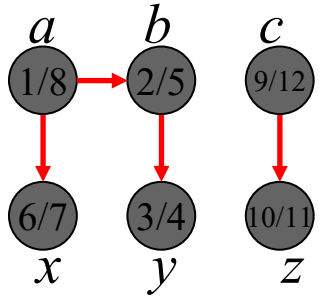
Running Example (DFS for directed graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```

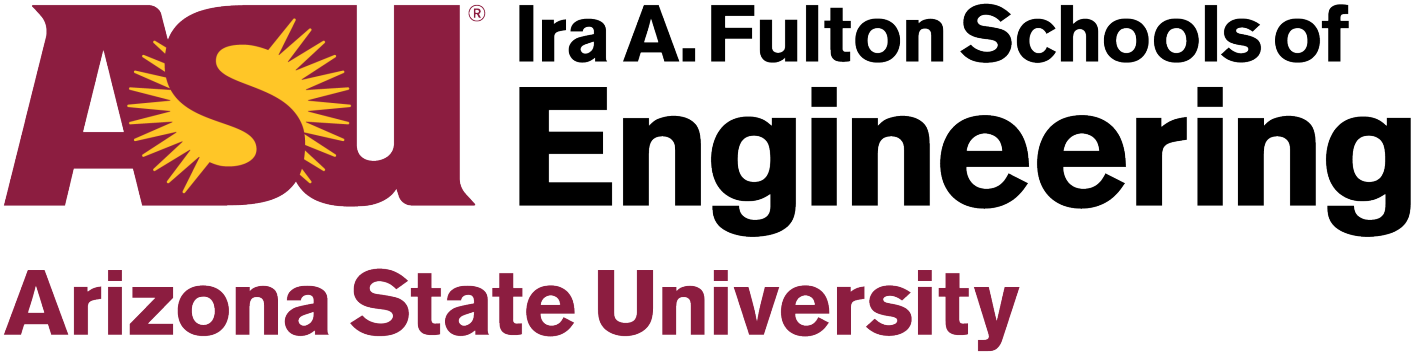




Graph G, adjacency list alphabetical



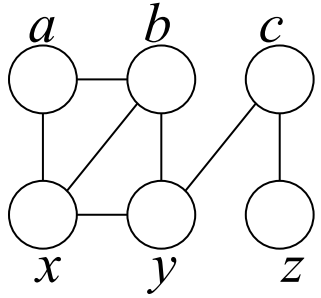
DFS Frost

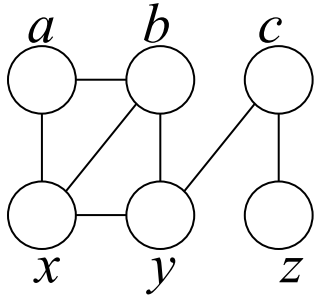

Ira A. Fulton Schools of
Engineering
Arizona State University

Graphs, Part 6

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



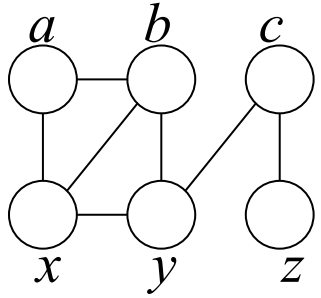


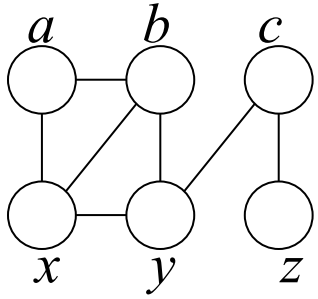
Graph G, adjacency list alphabetical

DFS(G)

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



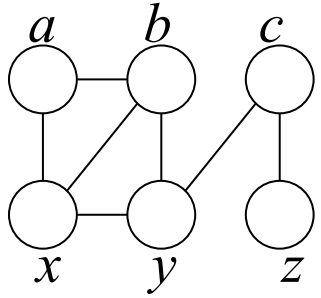


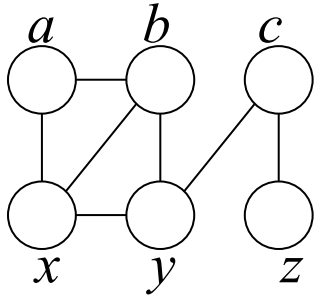
Graph G, adjacency list alphabetical

DFS(G): Lines 1-5

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



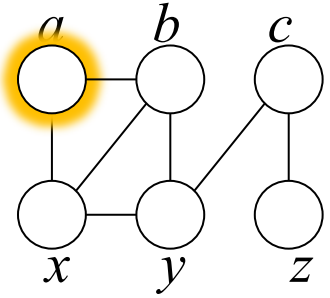


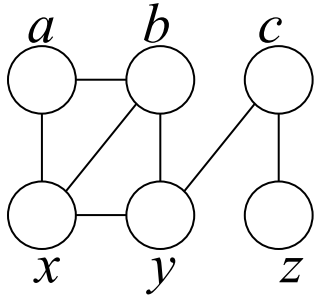
Graph G, adjacency list alphabetical

DFS(G): a is WHITE

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



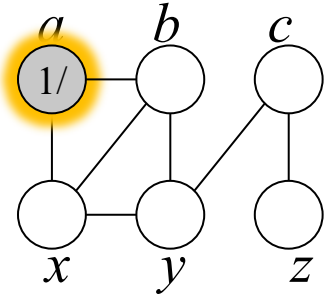


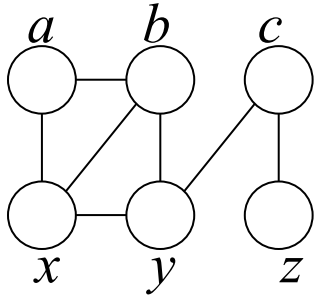
Graph G, adjacency list alphabetical

DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



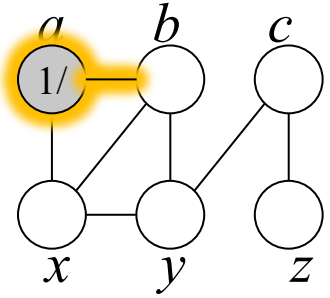


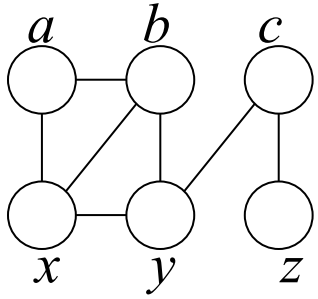
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Lines 1-3
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



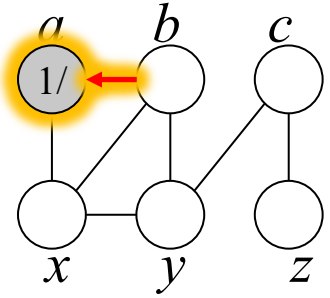


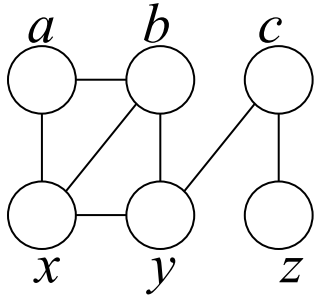
Graph G, adjacency list alphabetical

DFS-Visit(G, a): b is WHITE
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



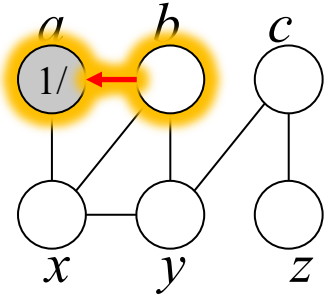


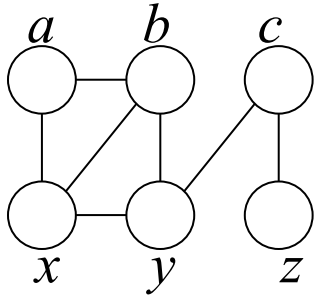
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Line 6
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



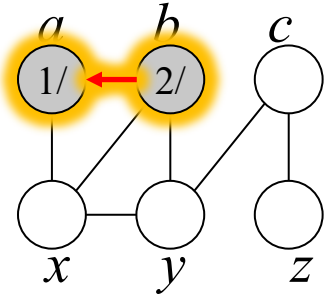


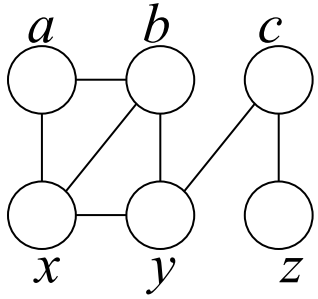
Graph G, adjacency list alphabetical

DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



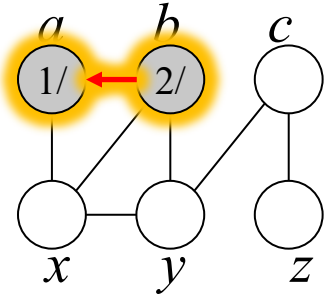


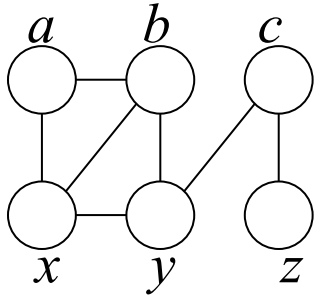
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Lines 1-3
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



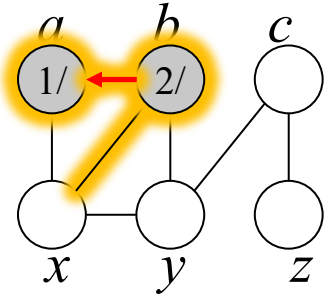


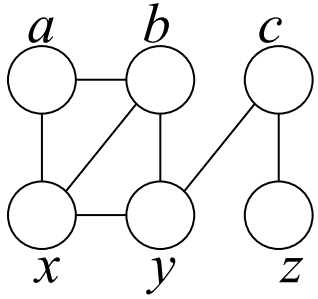
Graph G, adjacency list alphabetical

DFS-Visit(G, b): a is GREY
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



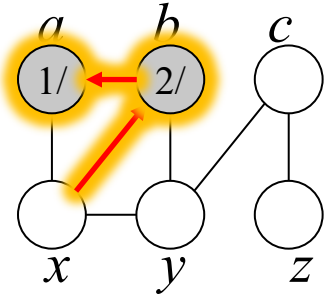


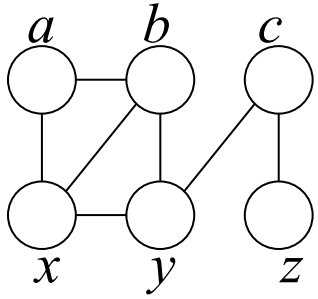
Graph G, adjacency list alphabetical

DFS-Visit(G, b): x is WHITE
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



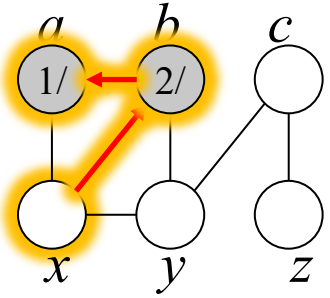


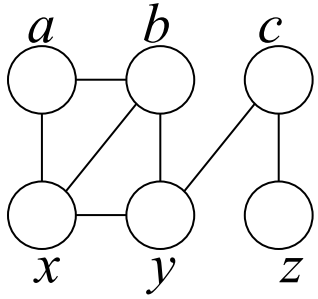
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Line 6
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



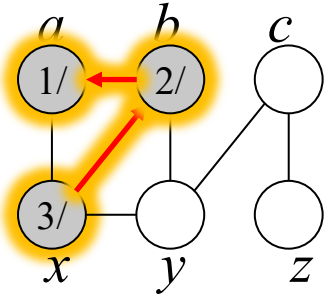


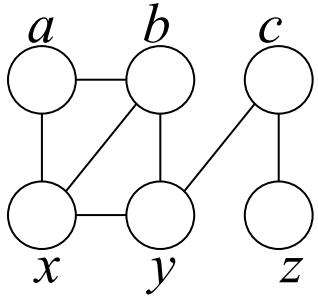
Graph G, adjacency list alphabetical

DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



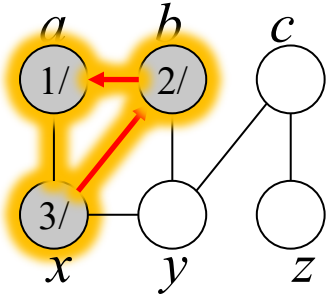


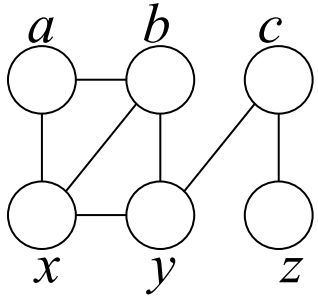
Graph G, adjacency list alphabetical

DFS-Visit(G, x): Lines 1-3
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



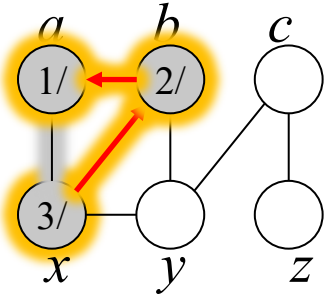


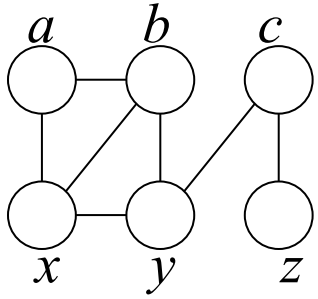
Graph G, adjacency list alphabetical

DFS-Visit(G, x): a is GRAY
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



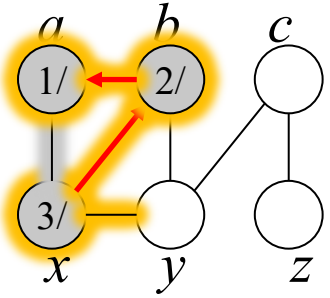


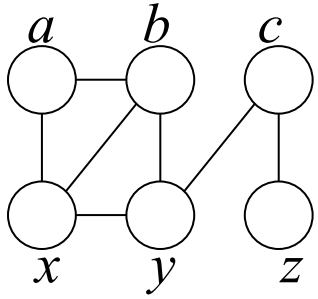
Graph G, adjacency list alphabetical

DFS-Visit(G, x): b is GRAY
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



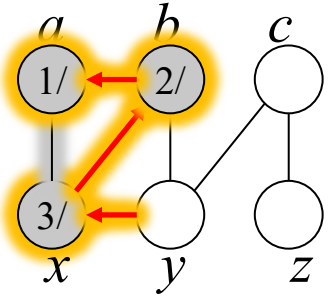


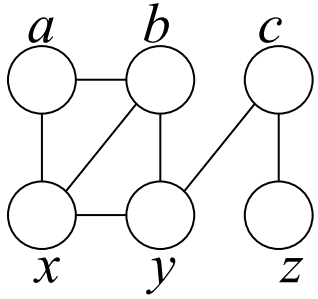
Graph G, adjacency list alphabetical

DFS-Visit(G, x): y is WHITE
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



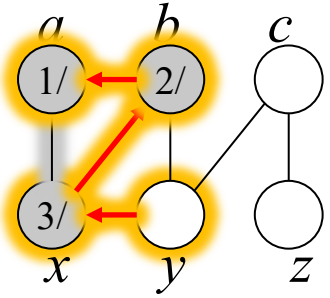


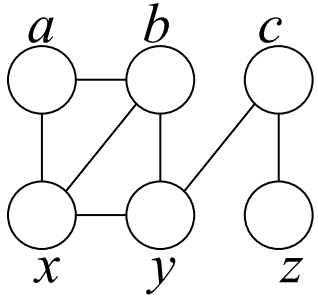
Graph G, adjacency list alphabetical

DFS-Visit(G, x): Line 6
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



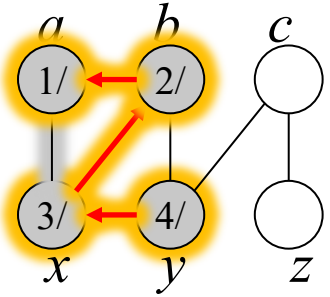


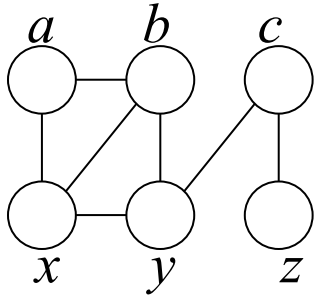
Graph G, adjacency list alphabetical

DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



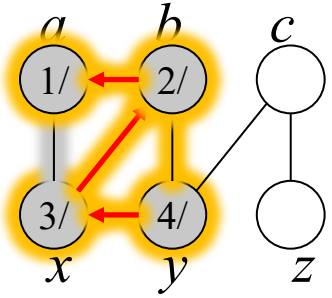


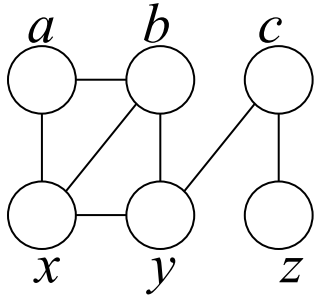
Graph G, adjacency list alphabetical

DFS-Visit(G, y): Lines 1-3
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



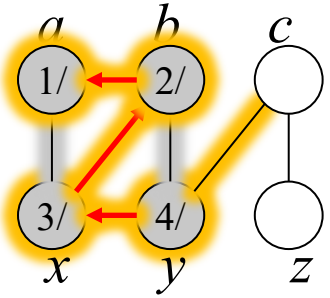


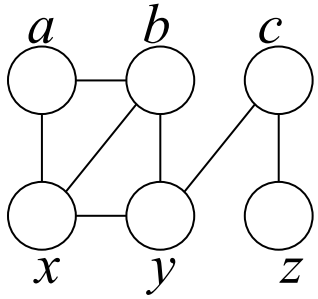
Graph G, adjacency list alphabetical

DFS-Visit(G, y): b is GRAY
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



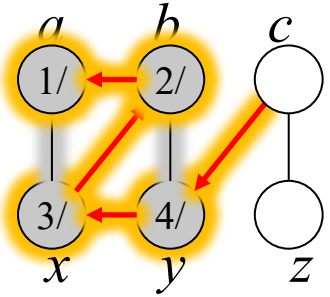


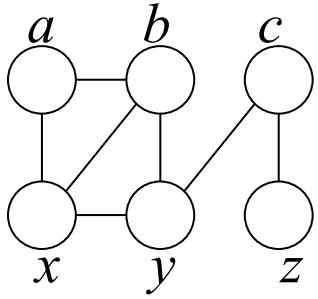
Graph G, adjacency list alphabetical

DFS-Visit(G, y): c is WHITE
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



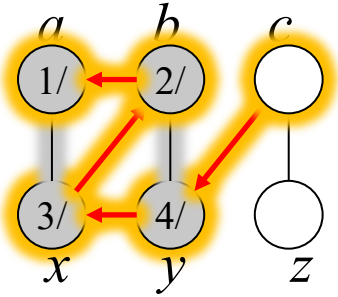


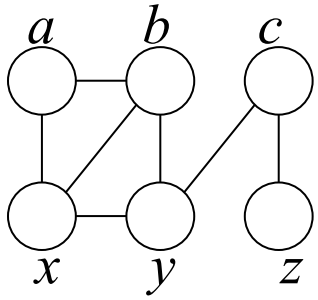
Graph G, adjacency list alphabetical

DFS-Visit(G, y): Line 6
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



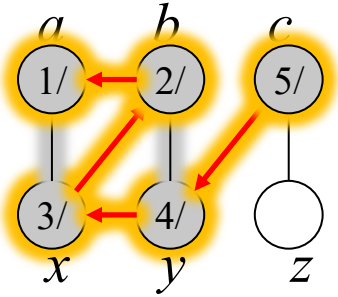


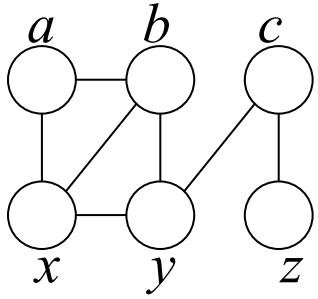
Graph G, adjacency list alphabetical

DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



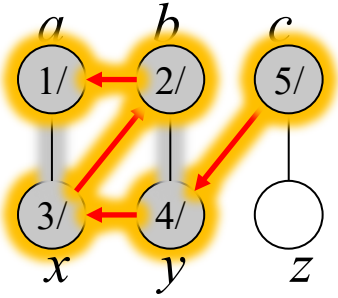


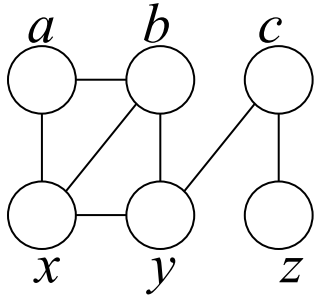
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Lines 1-3
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



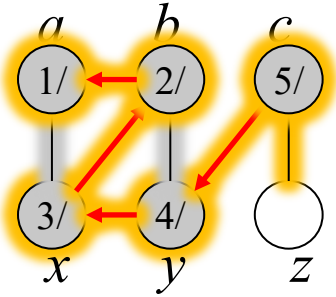


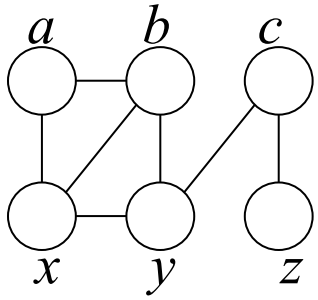
Graph G, adjacency list alphabetical

DFS-Visit(G, c): y is GRAY
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



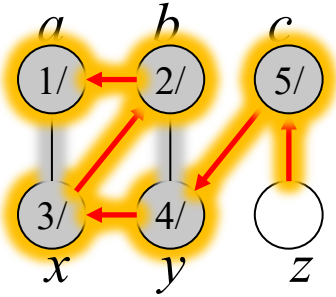


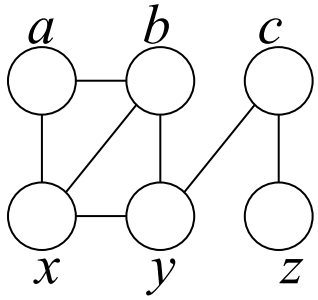
Graph G, adjacency list alphabetical

DFS-Visit(G, c): z is WHITE
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



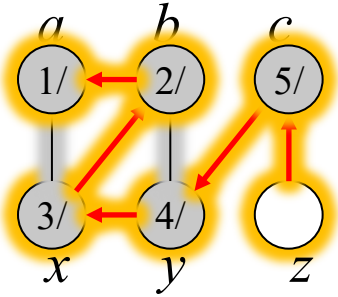


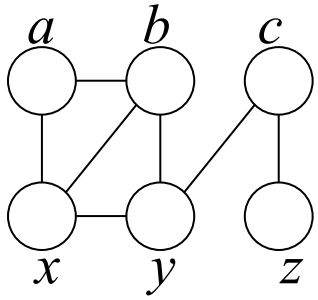
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Line 6
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



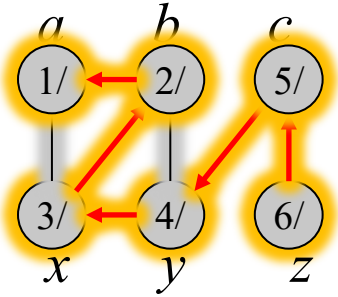


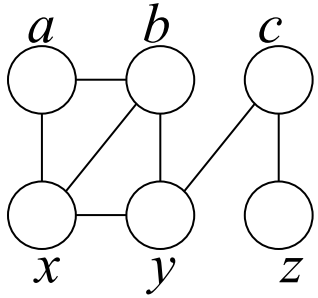
Graph G, adjacency list alphabetical

DFS-Visit(G, z)
DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



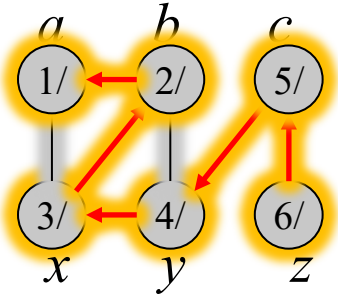


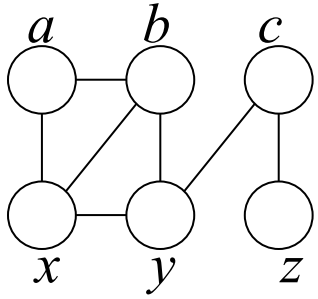
Graph G, adjacency list alphabetical

DFS-Visit(G, z): Lines 1-3
DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



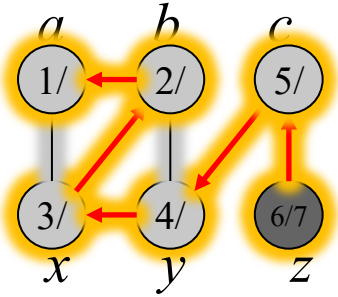


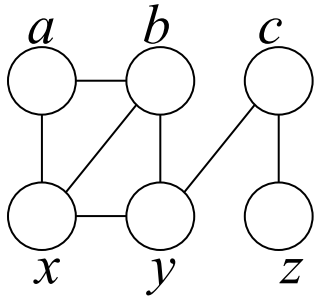
Graph G, adjacency list alphabetical

DFS-Visit(G, z): c is GRAY
DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



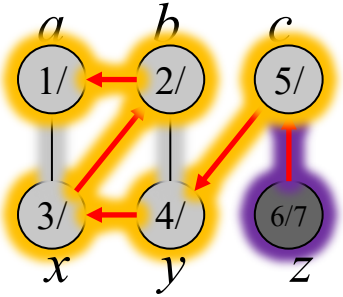


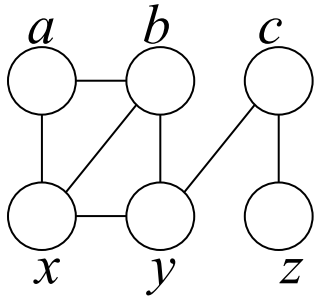
Graph G, adjacency list alphabetical

DFS-Visit(G, z): Lines 9-11
DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



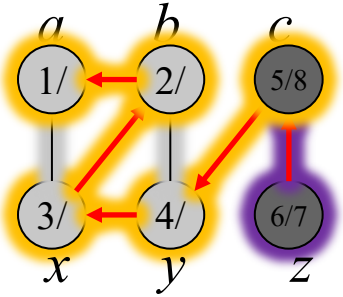


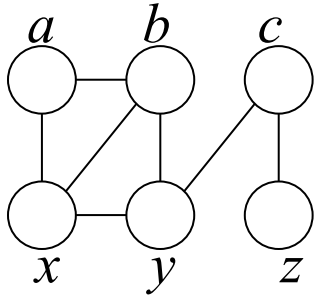
Graph G, adjacency list alphabetical

DFS-Visit(G, c)
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



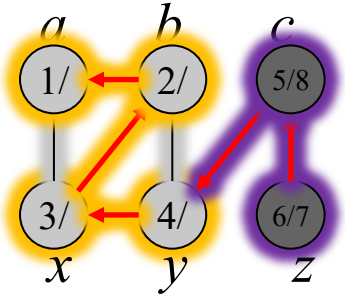


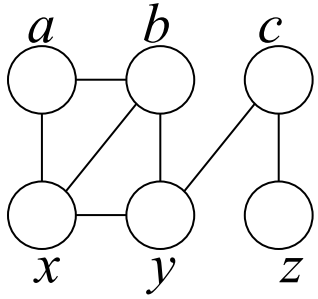
Graph G, adjacency list alphabetical

DFS-Visit(G, c): Lines 9-11
DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



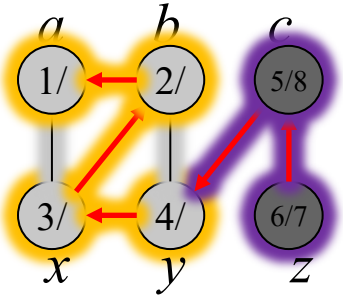


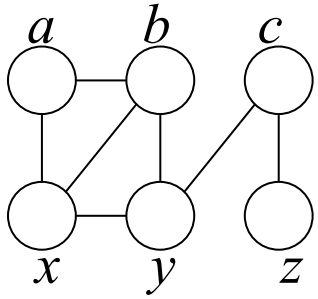
Graph G, adjacency list alphabetical

DFS-Visit(G, y)
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



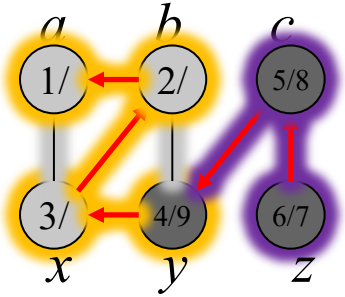


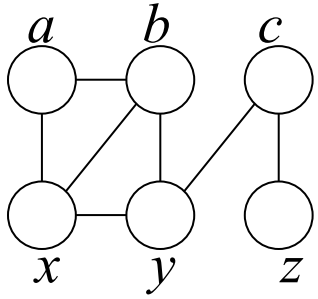
Graph G, adjacency list alphabetical

DFS-Visit(G, y): x is GRAY
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



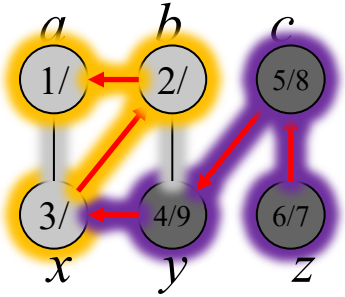


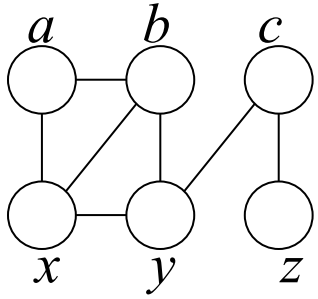
Graph G, adjacency list alphabetical

DFS-Visit(G, y): Lines 9-11
DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



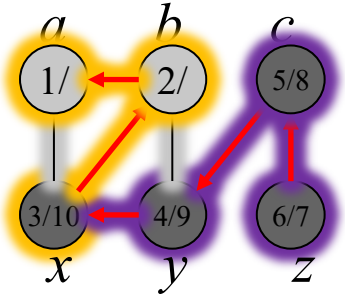


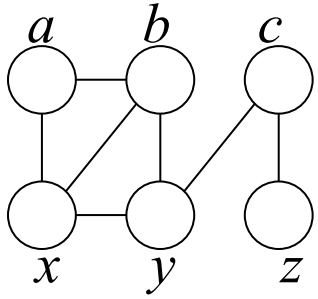
Graph G, adjacency list alphabetical

DFS-Visit(G, x)
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



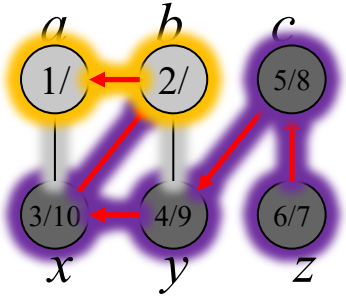


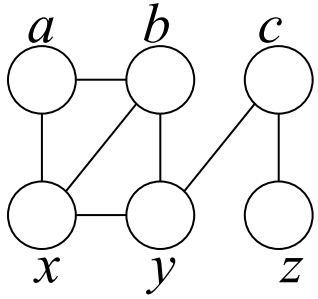
Graph G, adjacency list alphabetical

DFS-Visit(G, x): Lines 9-11
DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



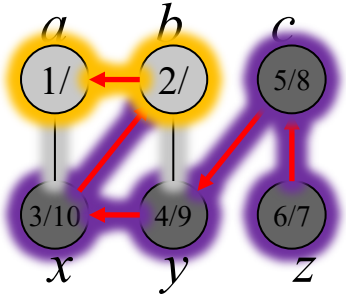


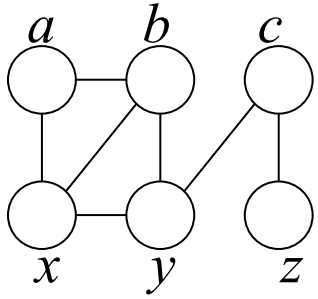
Graph G, adjacency list alphabetical

DFS-Visit(G, b)
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



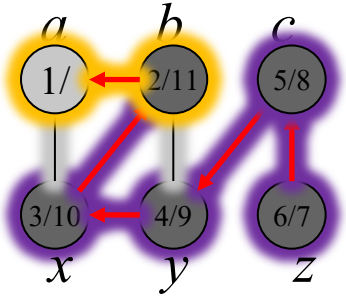


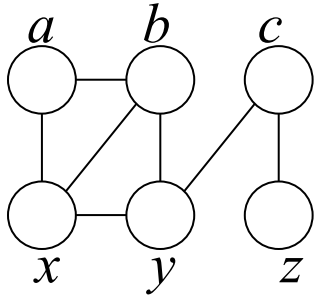
Graph G, adjacency list alphabetical

DFS-Visit(G, b): y is BLACK
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



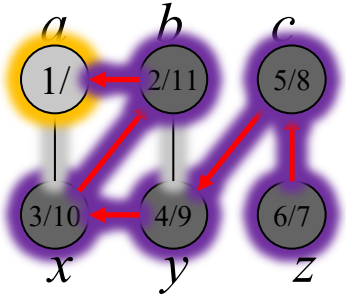


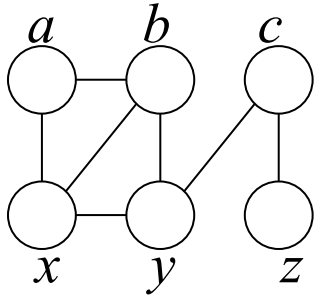
Graph G, adjacency list alphabetical

DFS-Visit(G, b): Lines 9-11
DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



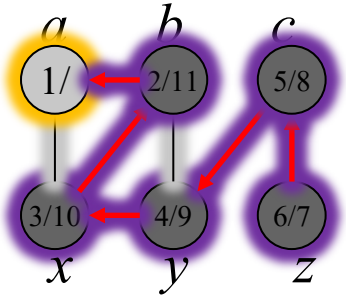


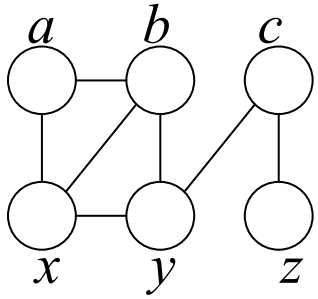
Graph G, adjacency list alphabetical

DFS-Visit(G, a)
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



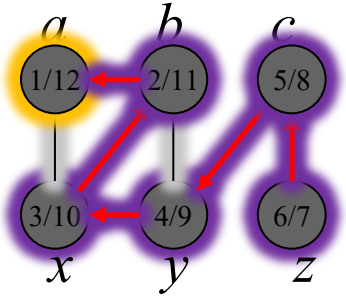


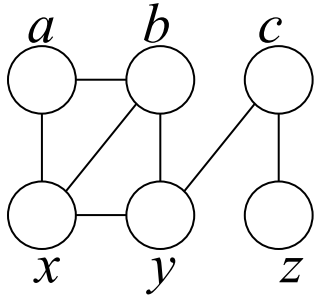
Graph G, adjacency list alphabetical

DFS-Visit(G, a): x is BLACK
DFS(G)

Running Example (DFS for undirected graph)

```
DFS-Visit(G, u)
1   time = time + 1;
2   u.d = time;
3   u.color = GRAY;
4   for each vertex v∈G.Adj[u]
5       if (v.color == WHITE){
6           v.π = u
7           DFS-Visit(G, v)
8       }
9   time = time + 1;
10  u.f = time;
11  u.color = BLACK;
```



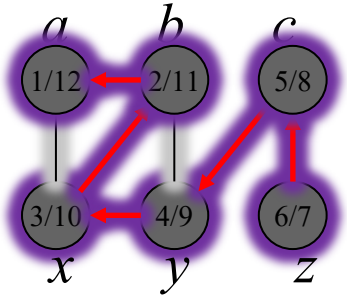


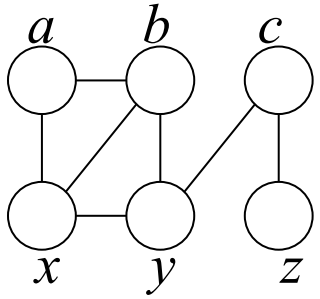
Graph G, adjacency list alphabetical

DFS-Visit(G, a): Lines 9-11
DFS(G)

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



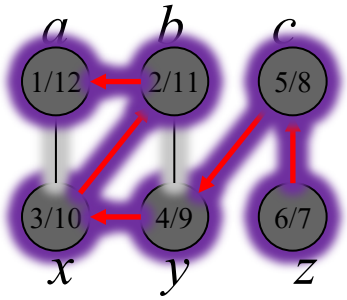


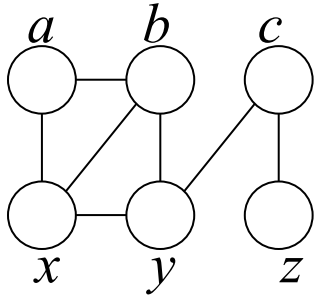
Graph G, adjacency list alphabetical

DFS(G)

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



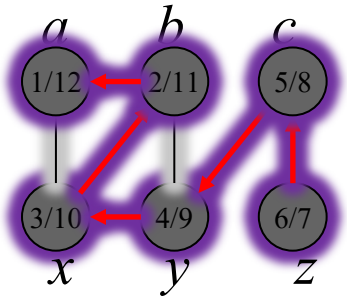


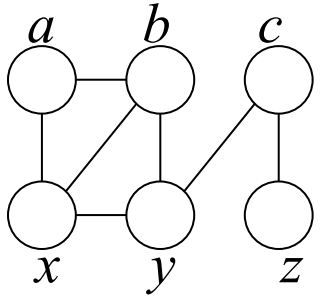
Graph G, adjacency list alphabetical

DFS(G): b is BLACK

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



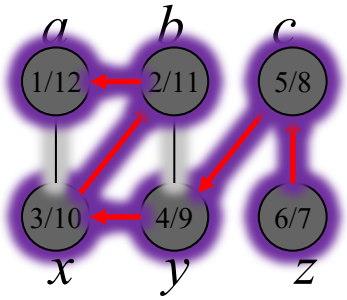


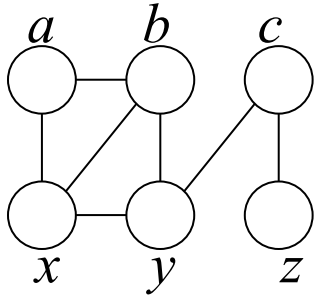
Graph G, adjacency list alphabetical

DFS(G): c is BLACK

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



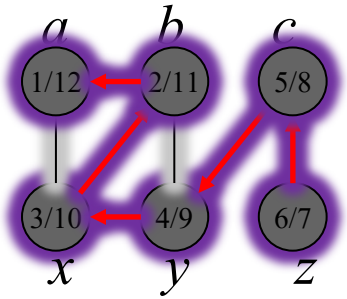


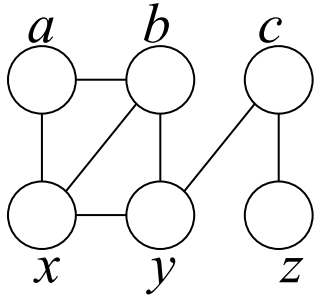
Graph G, adjacency list alphabetical

DFS(G): x is BLACK

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



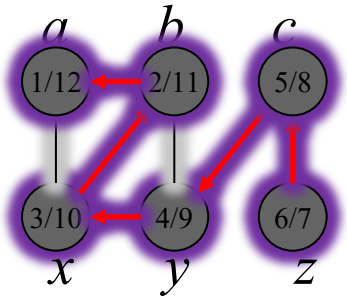


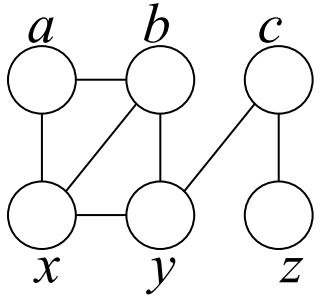
Graph G, adjacency list alphabetical

DFS(G): y is BLACK

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```



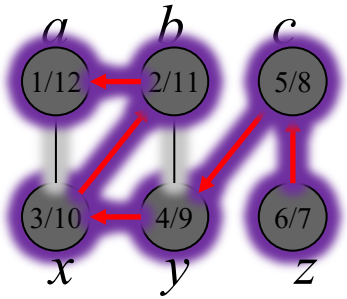


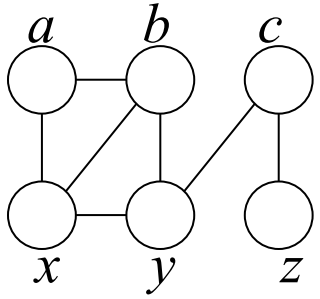
Graph G, adjacency list alphabetical

DFS(G): z is BLACK

Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```

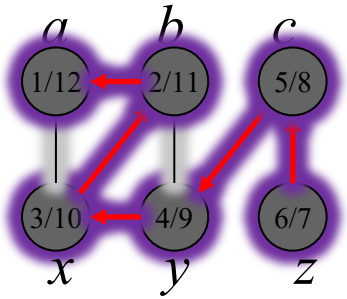


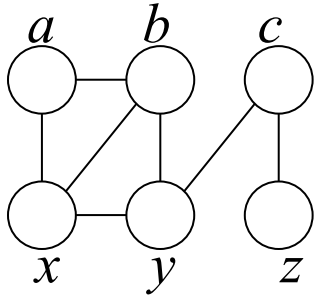


Graph G, adjacency list alphabetical

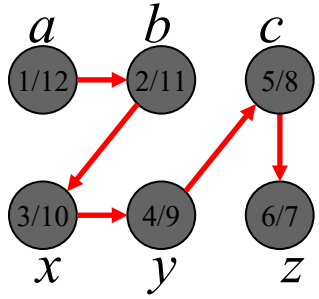
Running Example (DFS for undirected graph)

```
DFS(G)
1   for each vertex u ∈ G.V{
2       u.color = WHITE;
3       u.π = NIL;
4   }
5   time = 0;
6   for each vertex u ∈ G.V{
7       if (u.color == WHITE){
8           DFS-Visit(G, u);
9       };
10  }
```

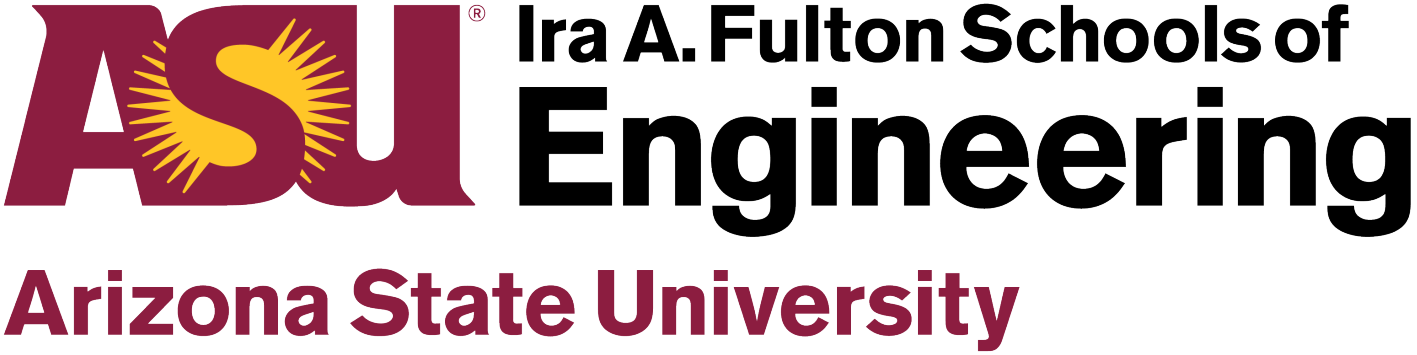




Graph G, adjacency list alphabetical



DFS Frost


Ira A. Fulton Schools of
Engineering
Arizona State University