

# INTRODUCTION to ARTIFICIAL INTELLIGENCE

## 2013-2014 SPRING SEMESTER

### LABORATORY MANUAL

#### Experiment 4

##### Depth-First Search

###### Idea:

- Starting at a node, follow a path all the way until you cannot move any further
- Then backtrack and try another branch
- Do this until all nodes have been visited
- Similar to finding a route in a maze

###### Implementation:

Assume you are given a digraph  $G = (V, E)$

- The same algorithm works for undirected graphs but the resulting structure imposed on the graph is different

We use 4 auxiliary arrays

- `color[u]`
  - white – undiscovered
  - gray – discovered but not yet processed
  - black – finished processing
- `pred[u]`, which points to the predecessor of `u`
  - The vertex that discovered `u`
- 2 timestamps: Purpose will be explained later
  - `d[u]`: Time at which the vertex was discovered  
Not to be confused with distance of `u` in BFS!
  - `f[u]`: Time at which the processing of the vertex was finished

```

DFS(G, s){
  for each u in V {                                // Initialization
    color[u] = white;
    pred[u] = NULL;
  } //end-for

  time = 0;
  for each u in V
    if (color[u] == white)                        // Found an undiscovered vertex
      DFSVisit(u);                               // Start a new search there
} // end-DFS

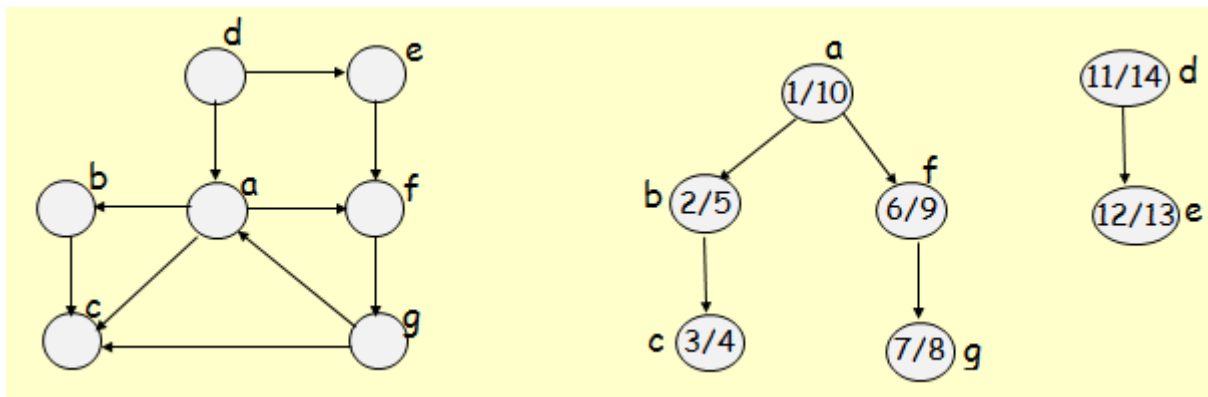
DFSVisit(u){                                       // Start a new search at u
  color[u] = gray;                               // Mark u visited
  d[u] = ++time;
  for each v in Adj[u] {
    if (color[v] == white){                      // if neighbor v undiscovered
      pred[v] = u;                             // ... set its predecessor
      DFSVisit(v);                             // ...visit v
    } //end-if
  } //end-for

  color[u] = black;                             // we are done with u
  f[u] = ++time;
} //end-DFSVisit

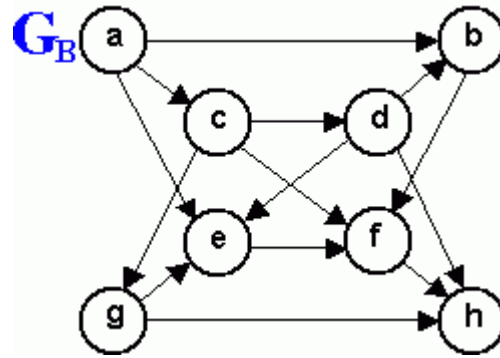
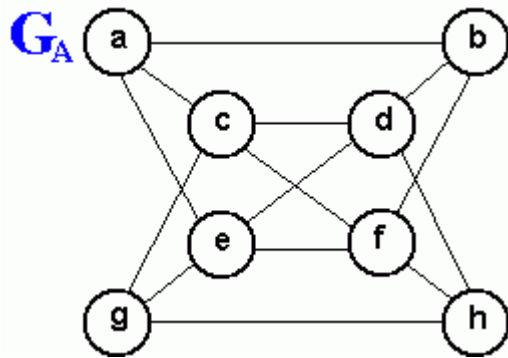
```

**Fig. 1.** Pseudo code of DFS algorithm

Example:



## Exercises



1. Consider a depth-first traversal of the undirected graph  $G_A$  shown above, starting from vertex a. List the order in which the nodes are visited. Apply all the steps of the algorithm given in Fig. 1. and explain each step in your report briefly.
2. Repeat exercise 1 for the directed graph  $G_B$ .
3. Write the program codes of DFS algorithm according to pseudo code given in Fig. 1. No other programs are evaluated during the lab.