NCKU Programming Contest Training Course

Course 5

2014/02/15

施冠榕

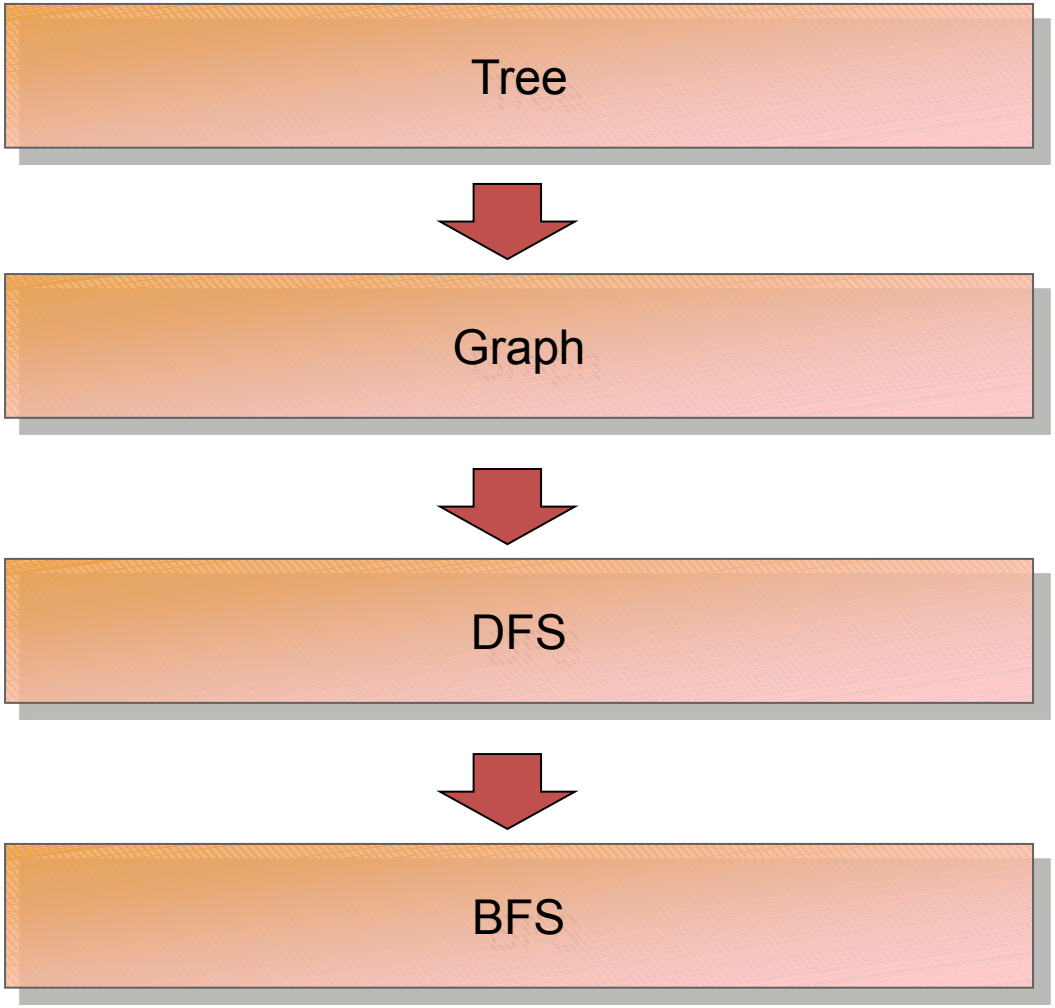
steven68680@gmail.com

http://myweb.ncku.edu.tw/~f74991081/Course5.zip

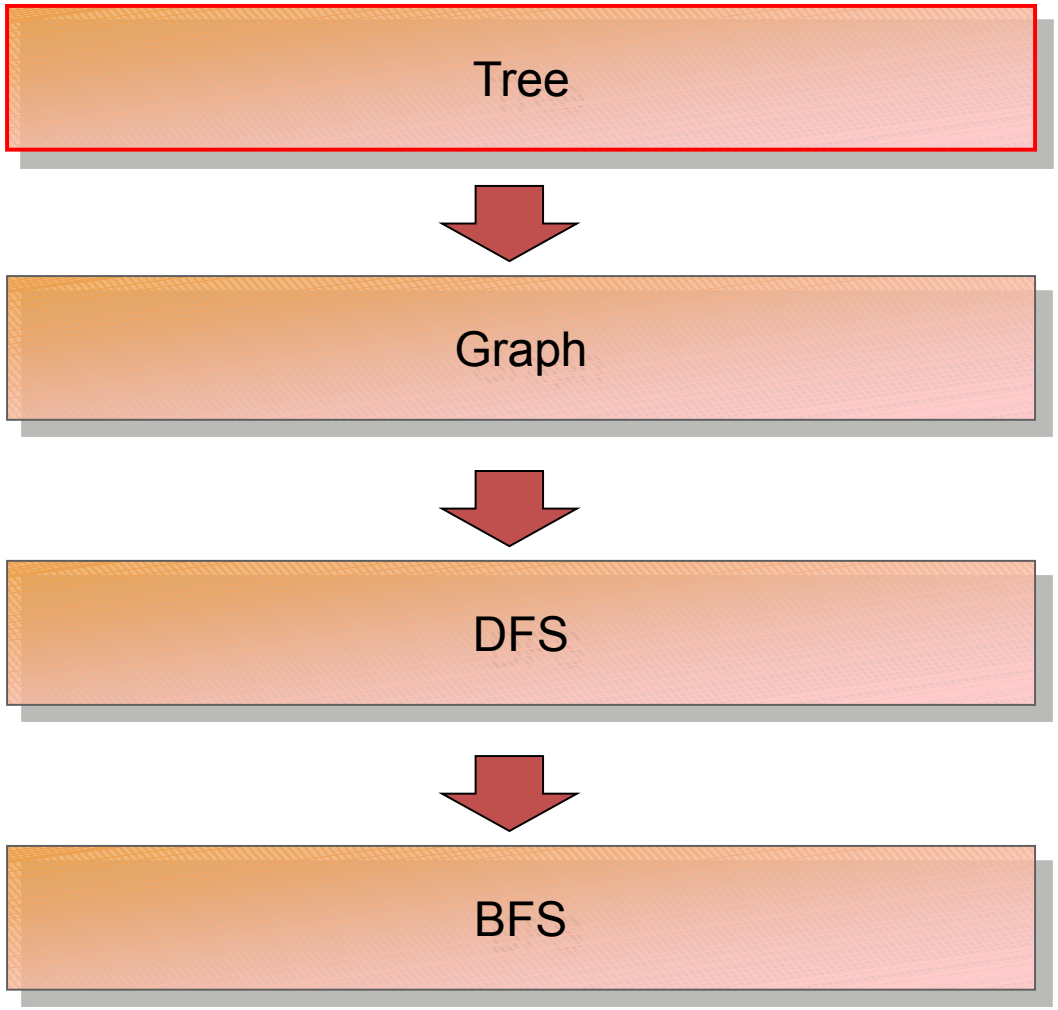
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Made By louis7340,Rongz

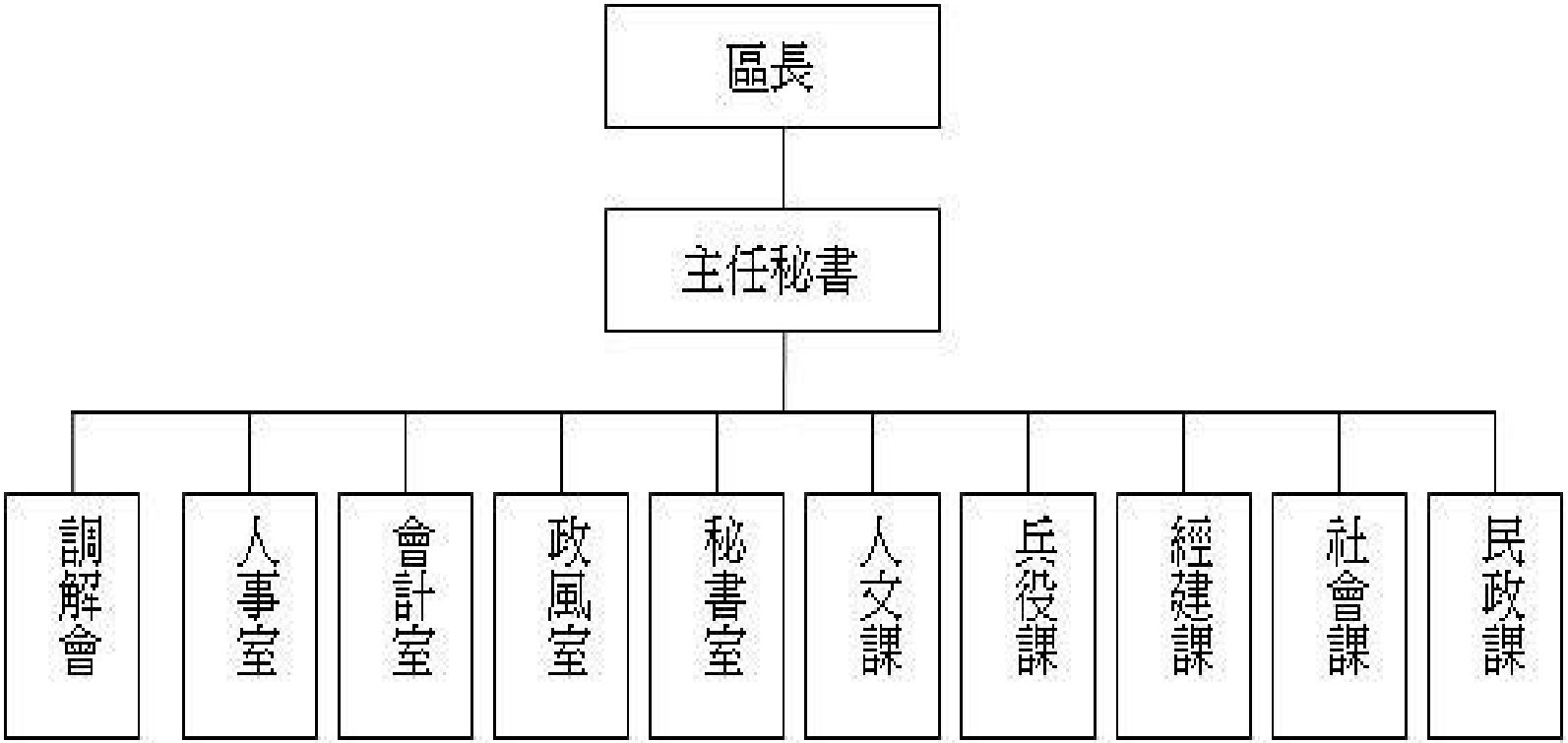
Outline



Outline

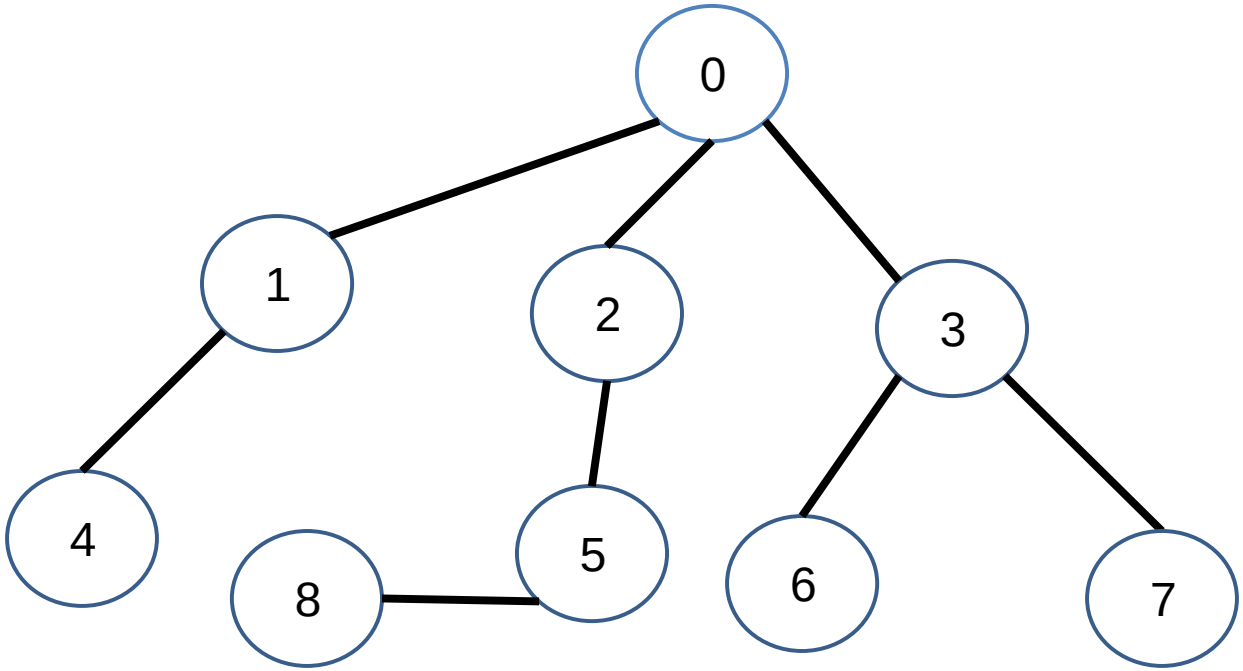


Made By louis7340,Rongz


區長
主任秘書
調解會
人事室
會計室
政風室
秘書室
人文課
兵役課
經建課
社會課
民政課

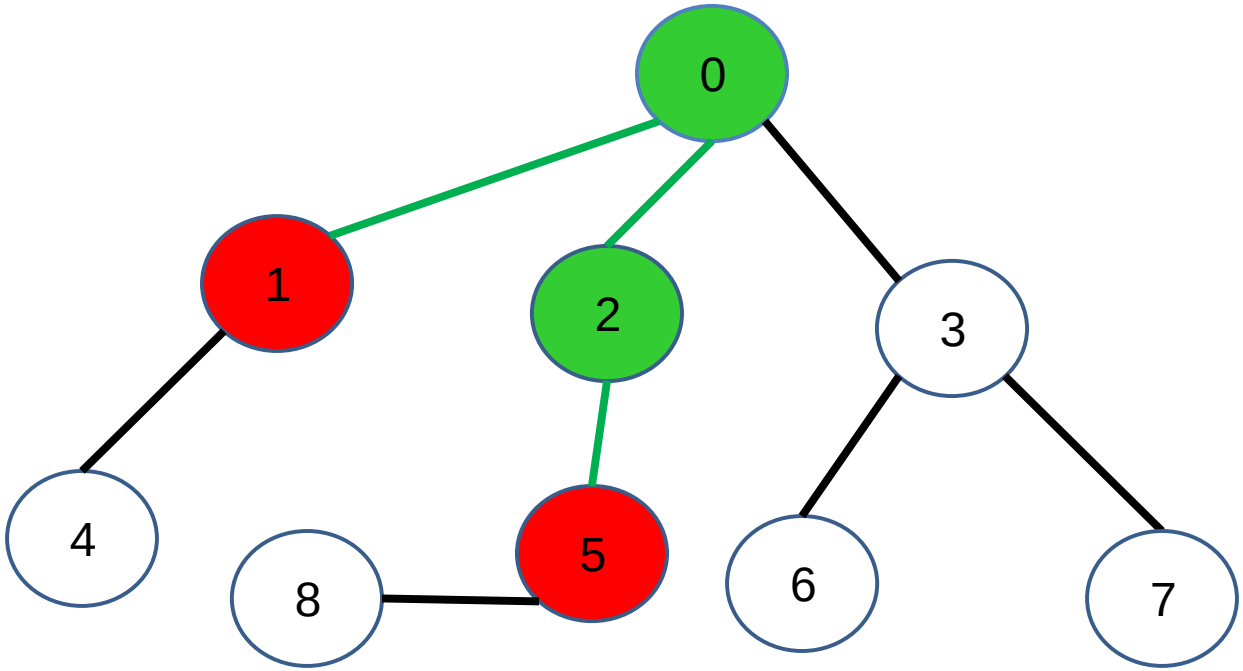
Tree

- 由 node 組成的資料結構



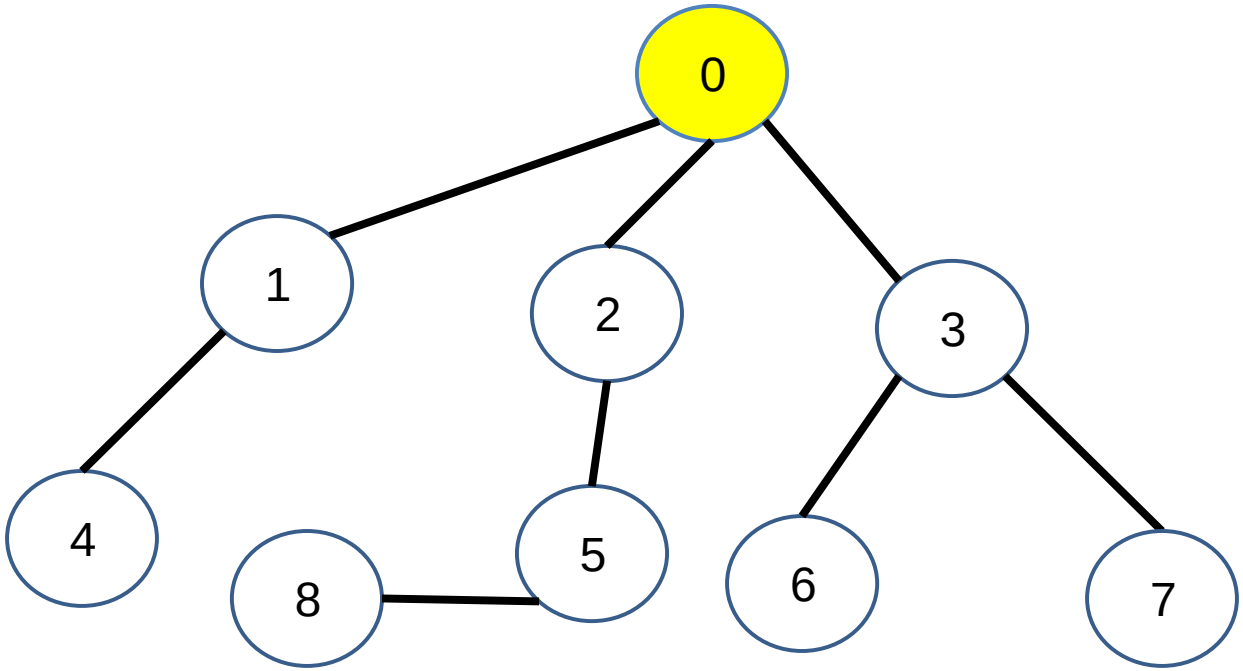
Tree

- 由 node 組成的資料結構
- 每兩個點之間必有一個路徑，且必只有一個路徑



Tree

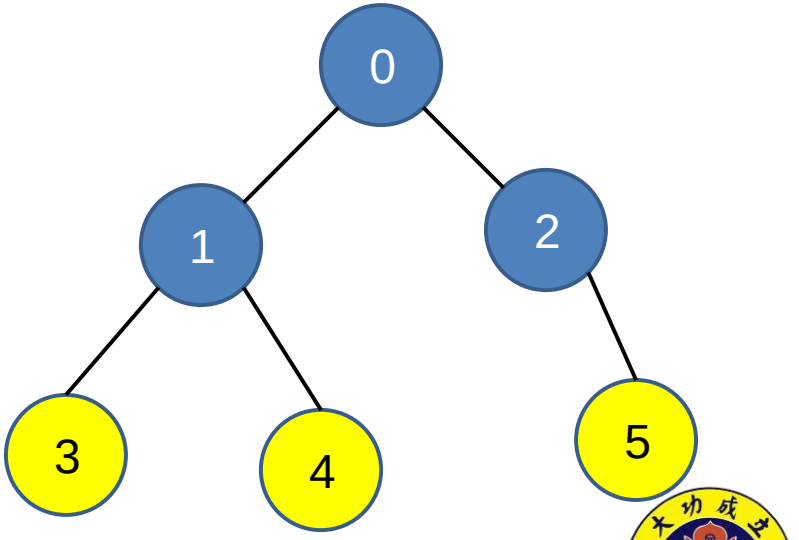
- 由 node 組成的資料結構
- 每兩個點之間必有一個路徑，且必只有一個路徑
- 每個樹都有一個 root



Tree

Relations: (using node 0 as root)

- **Leaf :** 沒有孩子的 **node**

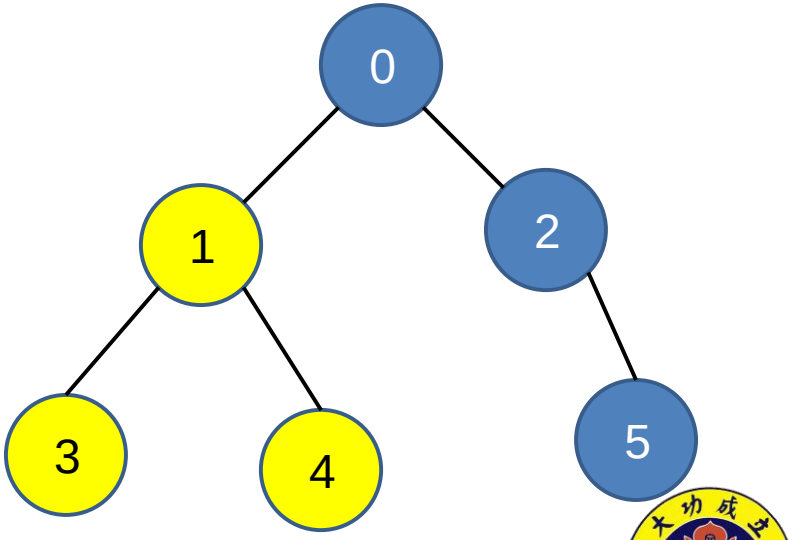


Tree

Relations: (using node 0 as root)

- **Leaf :** 沒有孩子的 **node**
- **Parent/Children :** 上面的 node –Parent (1)

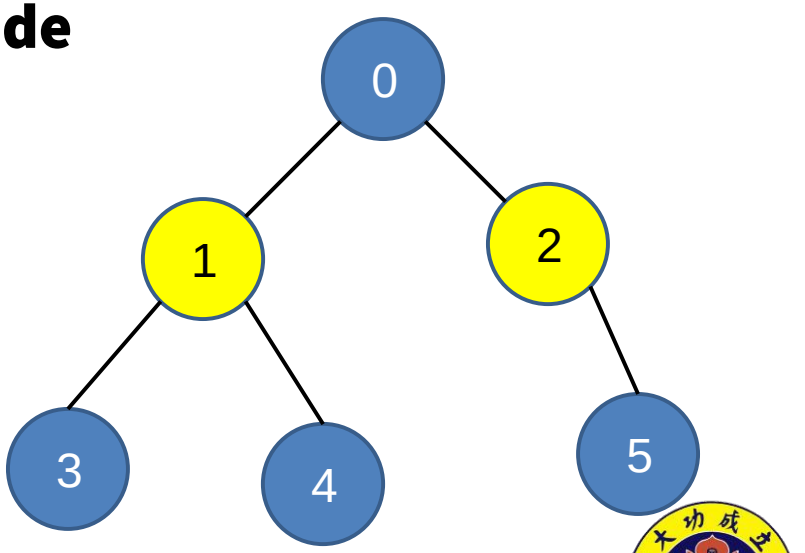
  下面的 node (3,4)



Tree

Relations: (using node 0 as root)

- **Leaf :** 沒有孩子的 **node**
- **Parent/Children :** 上面的 node –Parent
  下面的 node
- **Sibling:** 有一樣 **parent** 的 **node**



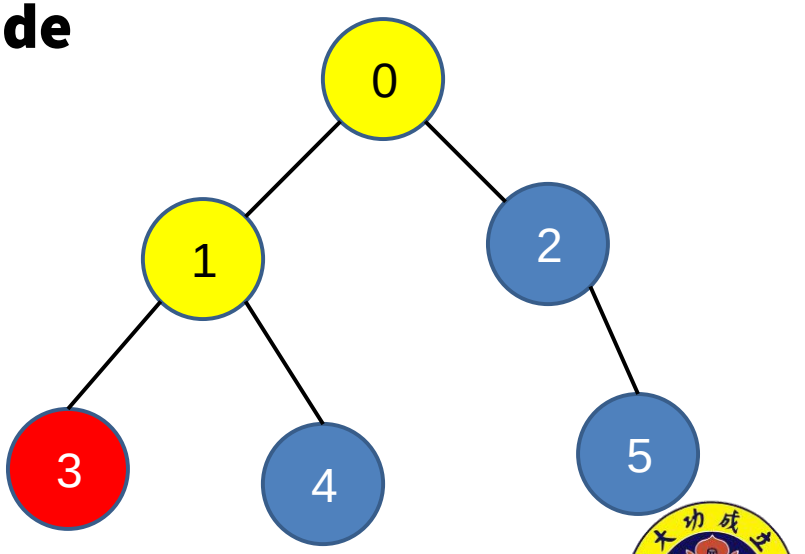
Tree

Relations: (using node 0 as root)

- **Leaf :** 沒有孩子的 **node**
- **Parent/Children :** 上面的 node –Parent

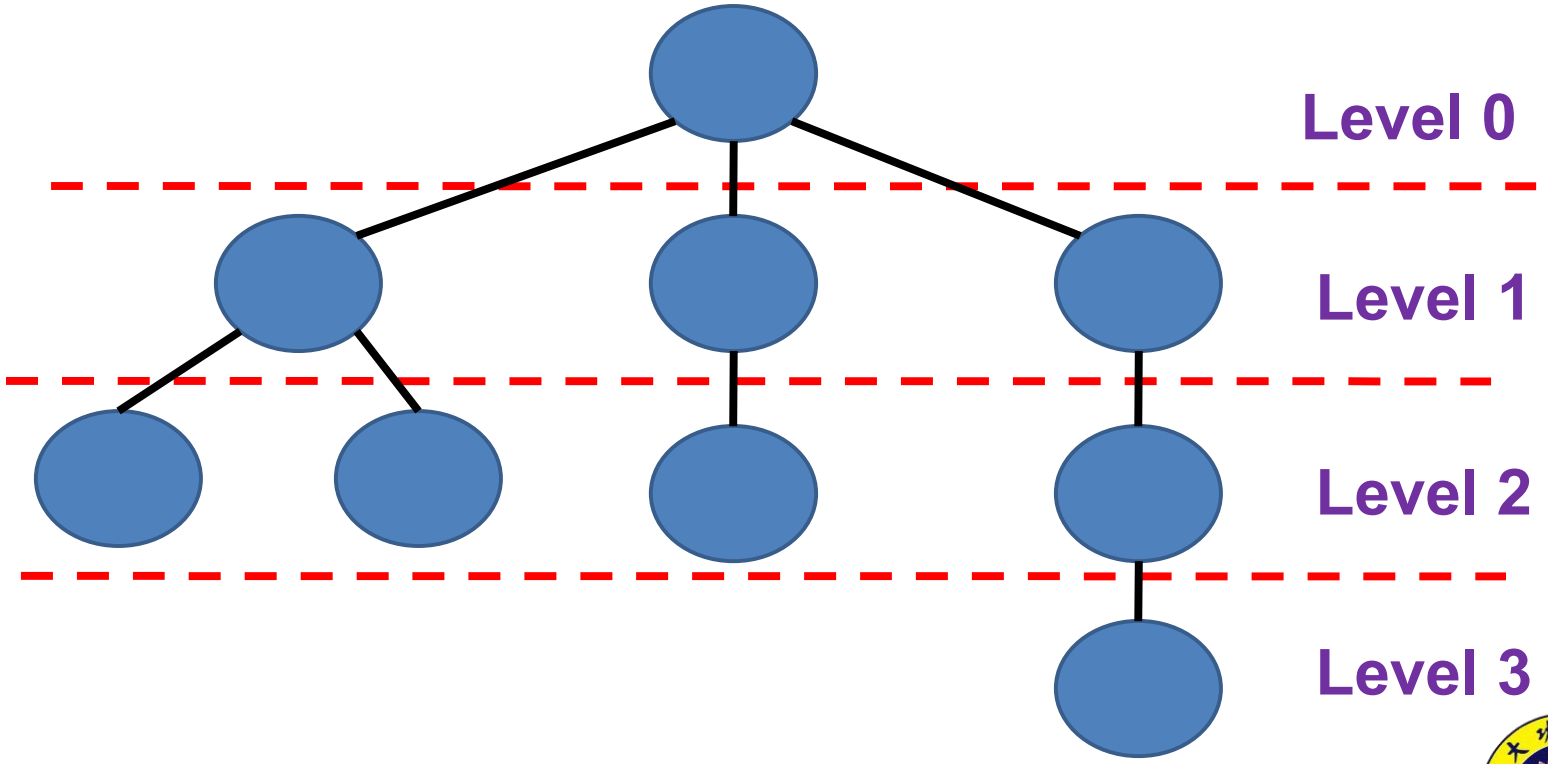
  下面的 node
- **Sibling:** 有一樣 **parent** 的 **node**
- **Ancestor:** 到 **root** 上的 **node**

  (3's ancestor: 0,1)



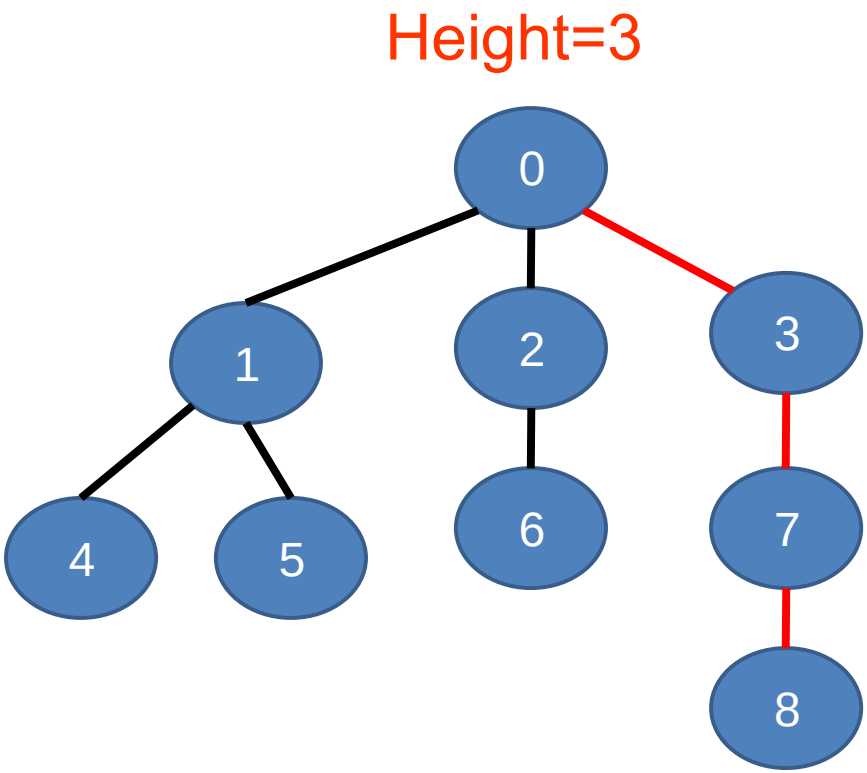
Tree

- **Tree Level:** 從根到 **root** 的距離



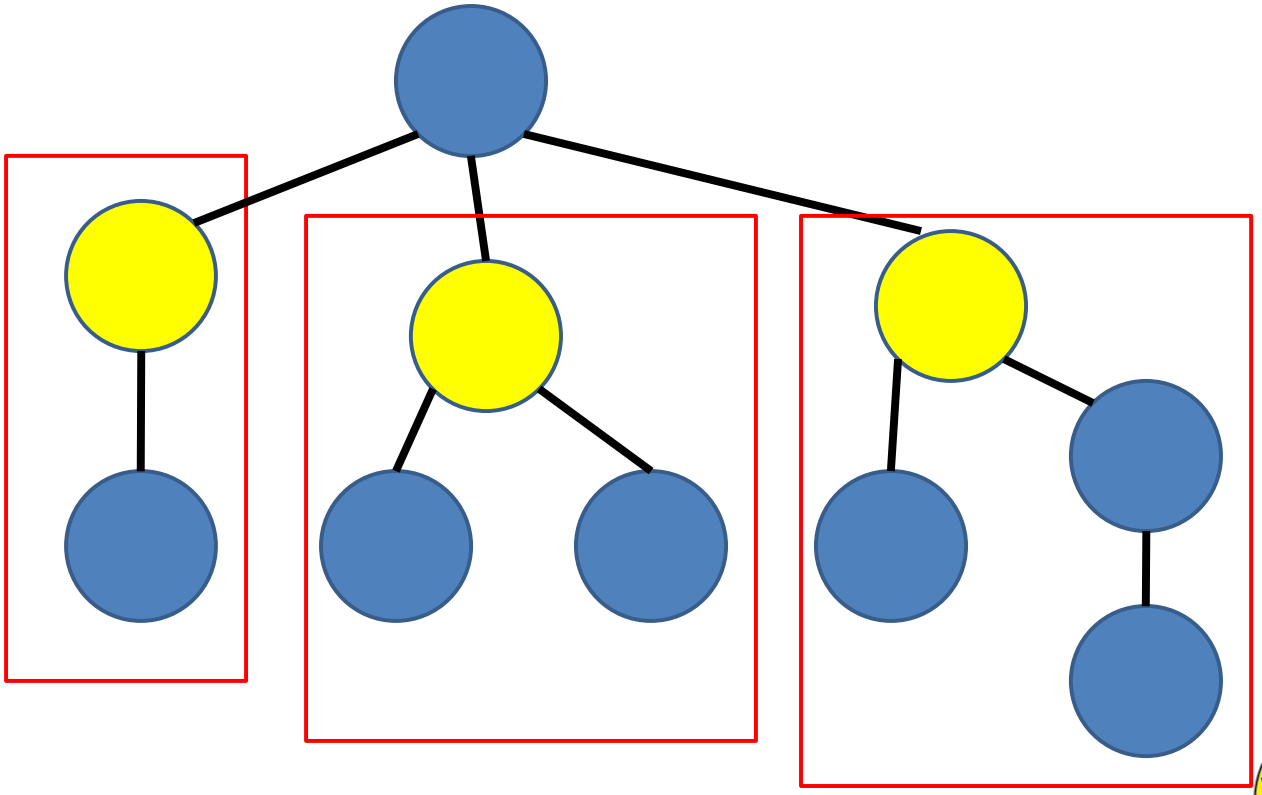
Tree

- **Tree Height(Depth):** the maximum distance from the root.



Tree

- **SubTree:** nodes that are not the root can form a subtree (become a root)



Tree

- **Feature:**

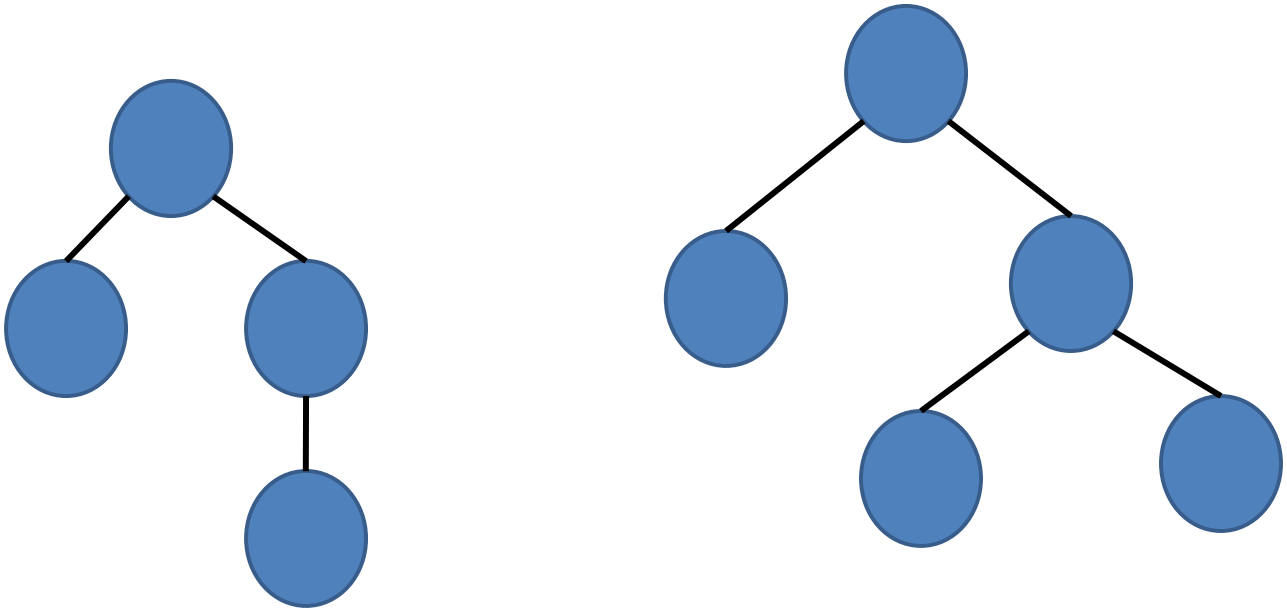
  -no cycle

  -every pair of nodes are connected

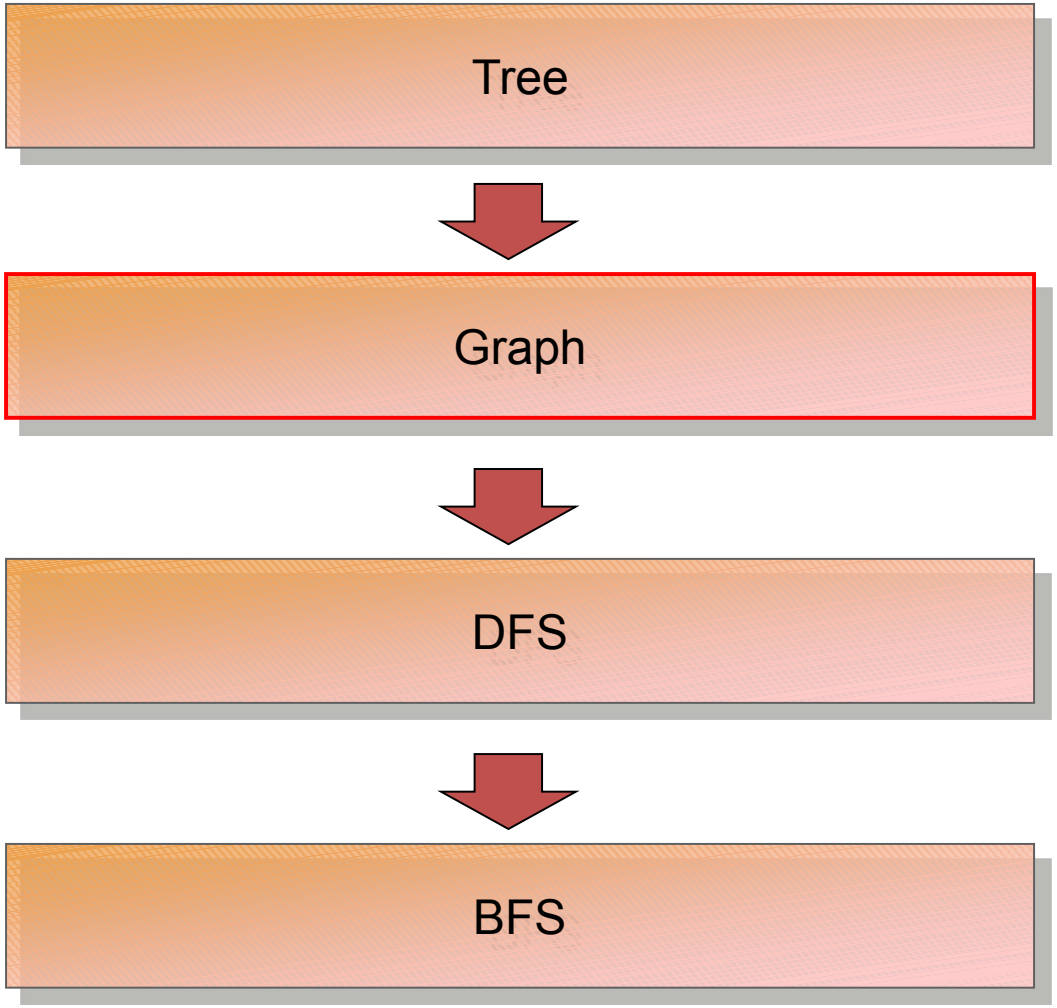
  -every pair of nodes has only one path

Tree

- **Forest:** n trees are disjoint ($n \geq 0$)

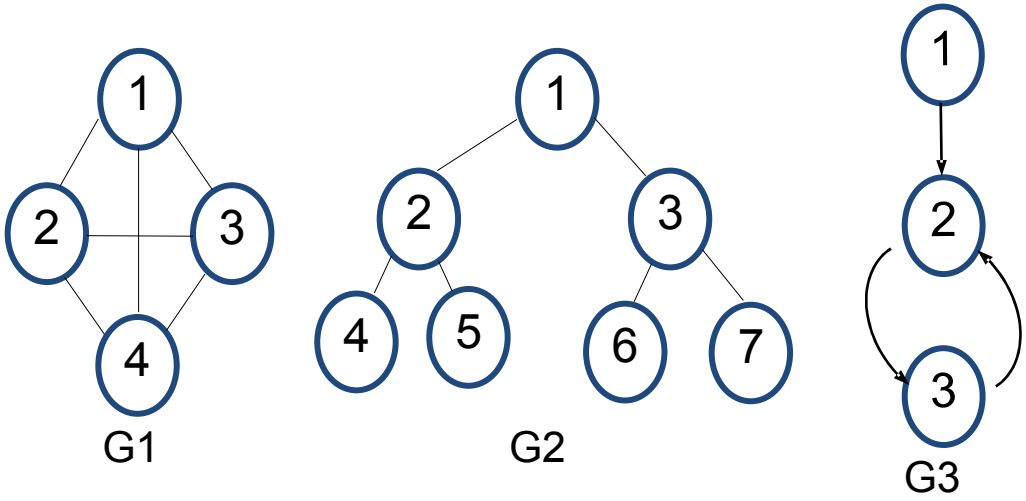


Outline



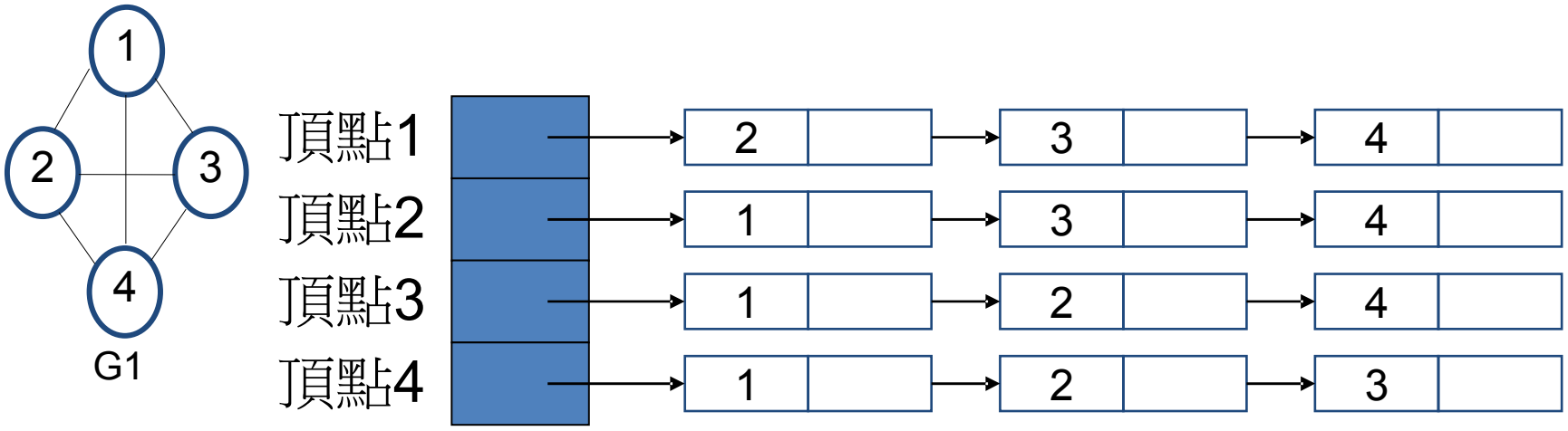
Graph

- Undirected Graph – G1, G2
- Directed Graph – G3
  - deg: in-degree and out-degree



Graph

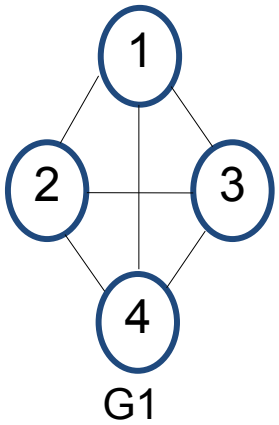
- Representation
  - adjacent list



Made By louis7340

Graph

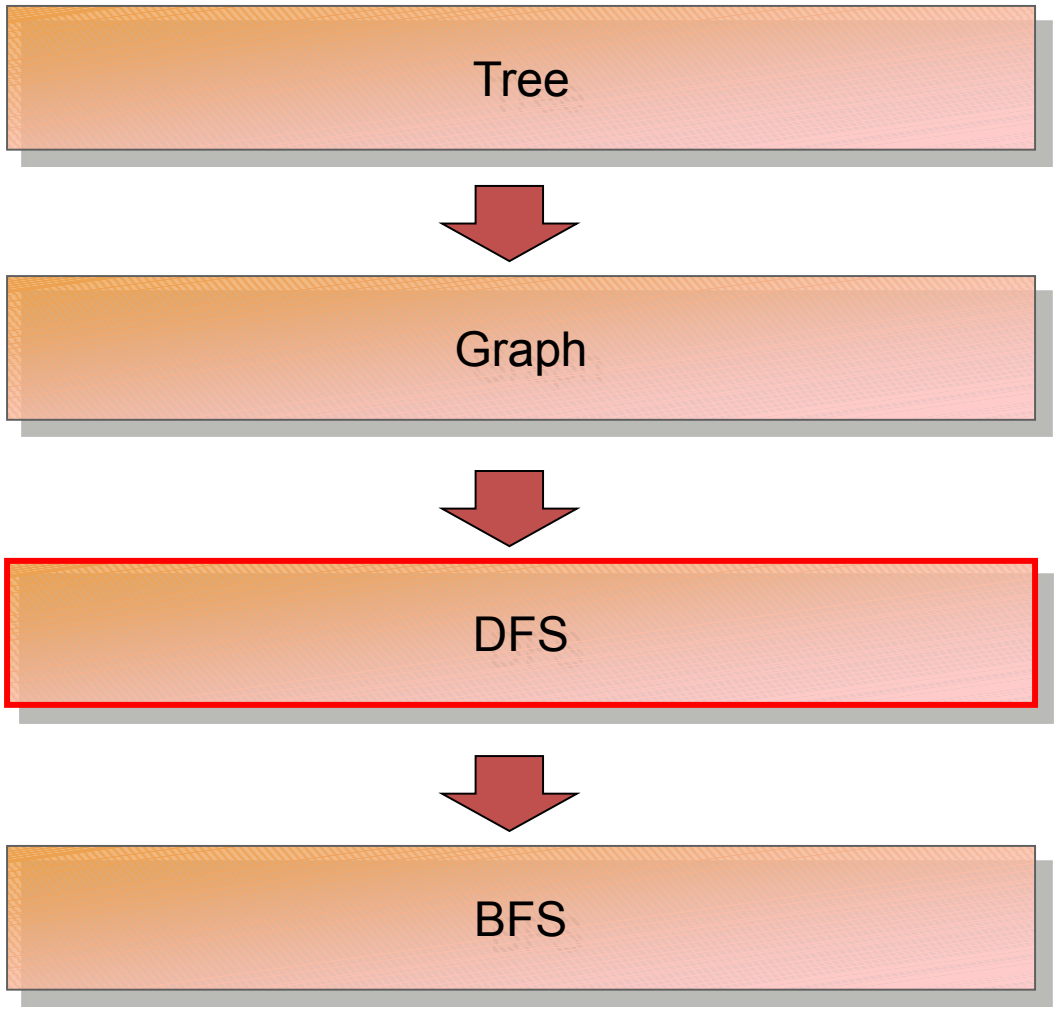
- Representation
  - adjacent matrix



G1

	1	2	3	4
1	0	1	1	1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0

Outline



DFS

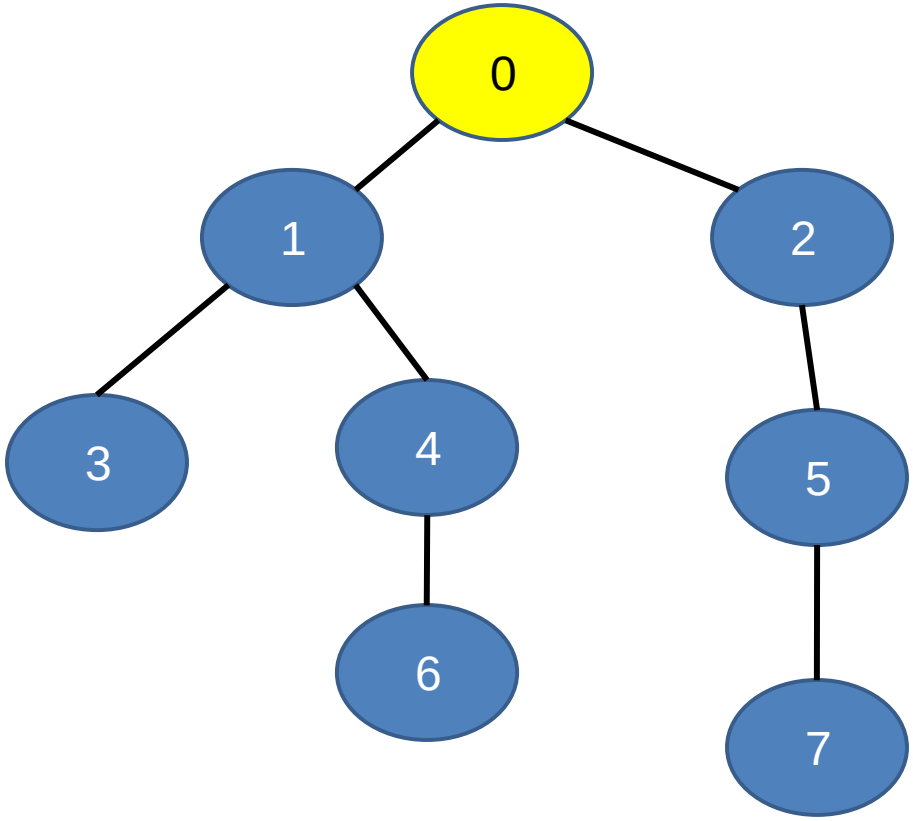
(D)epth-(F)irst-(S)earch

DFS

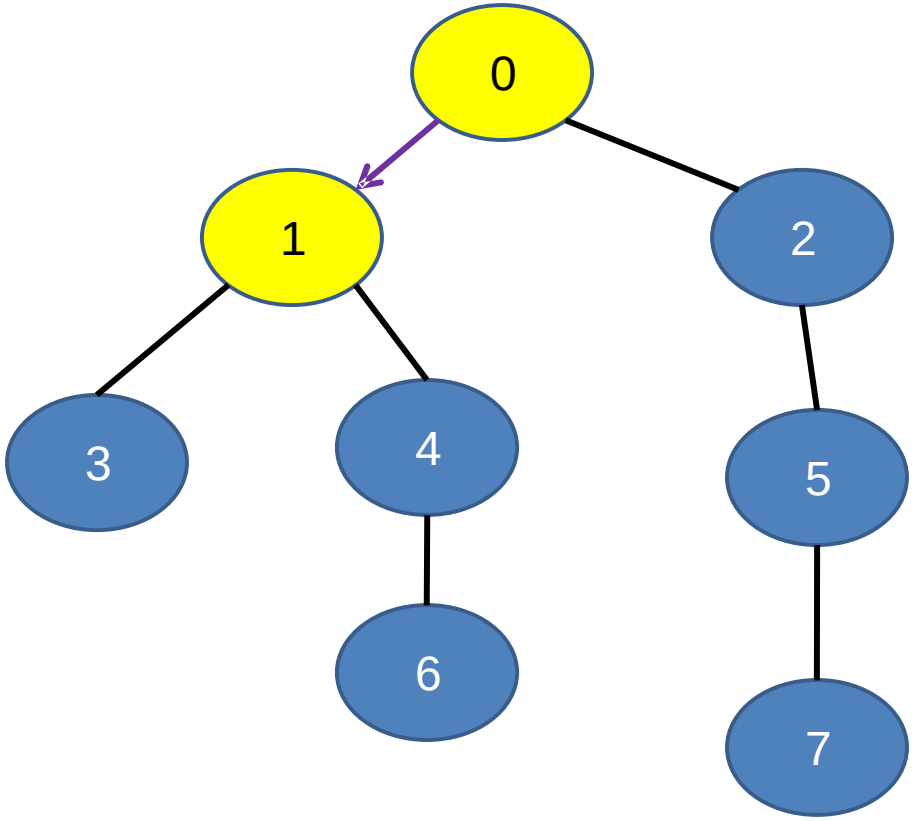
(D)epth (F)irst (S)earch

Stack

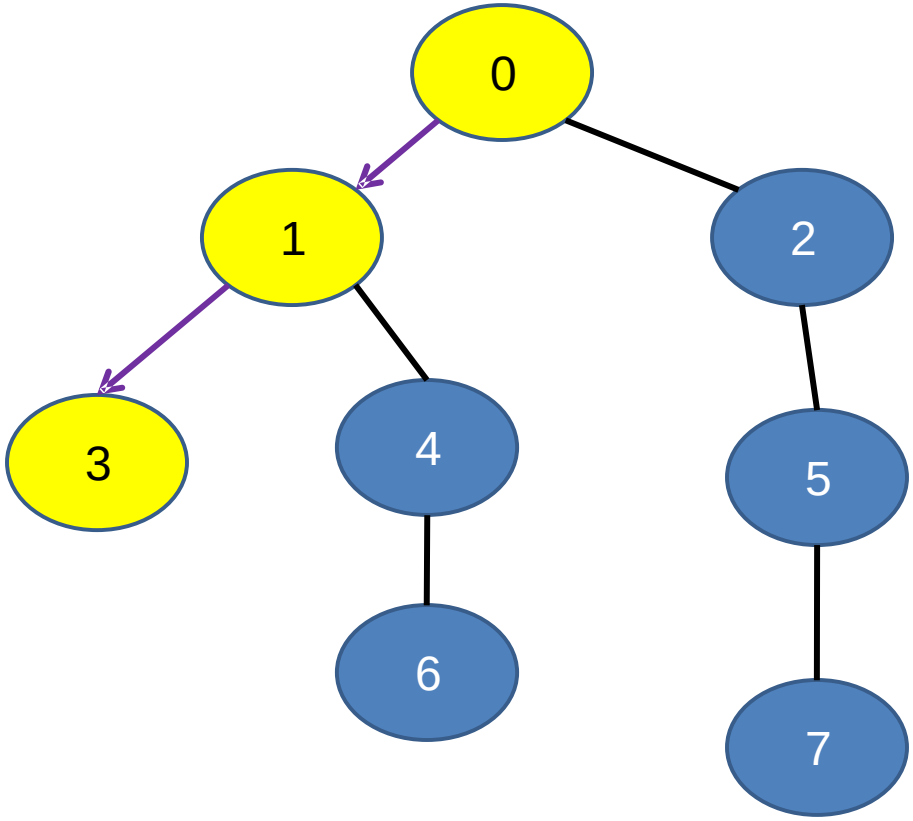
DFS



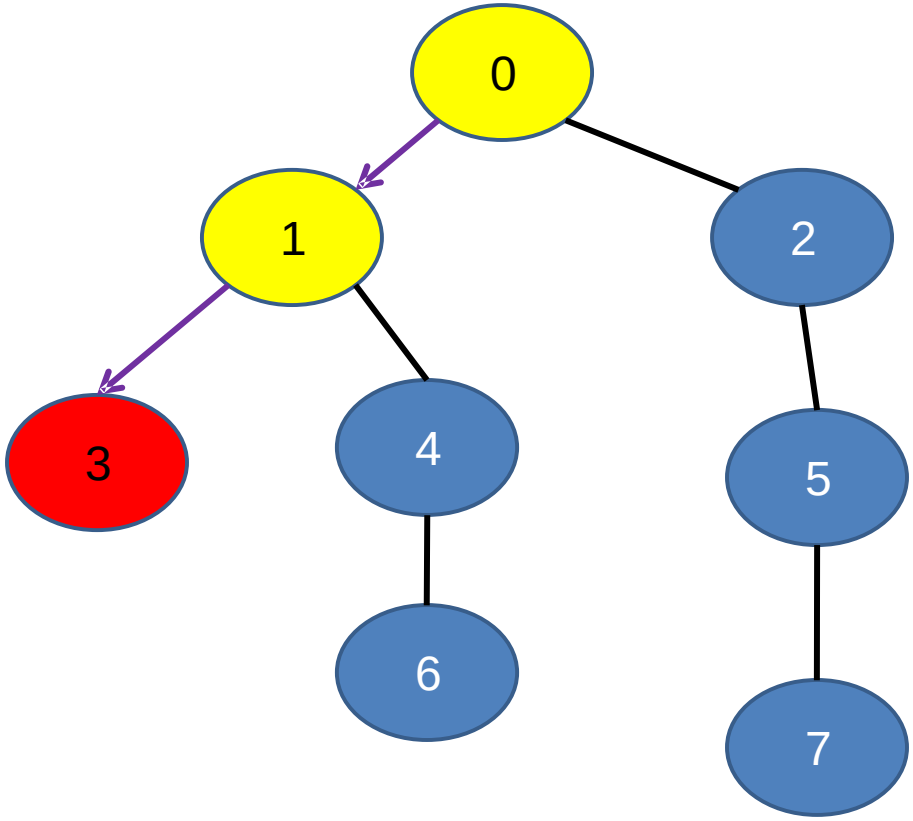
DFS



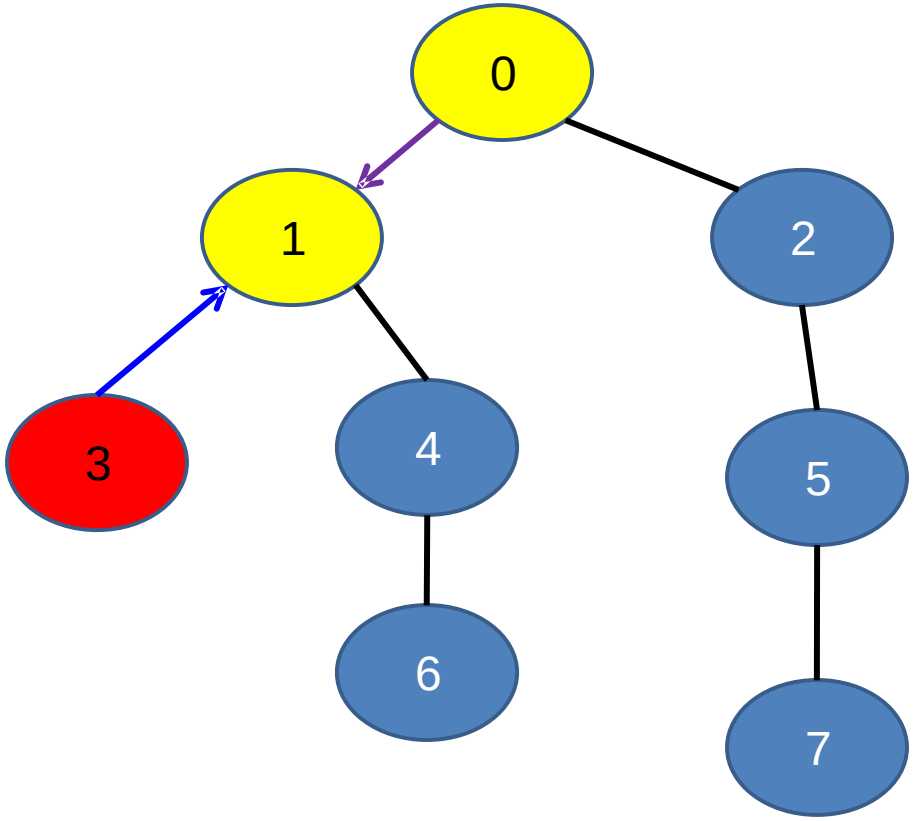
DFS



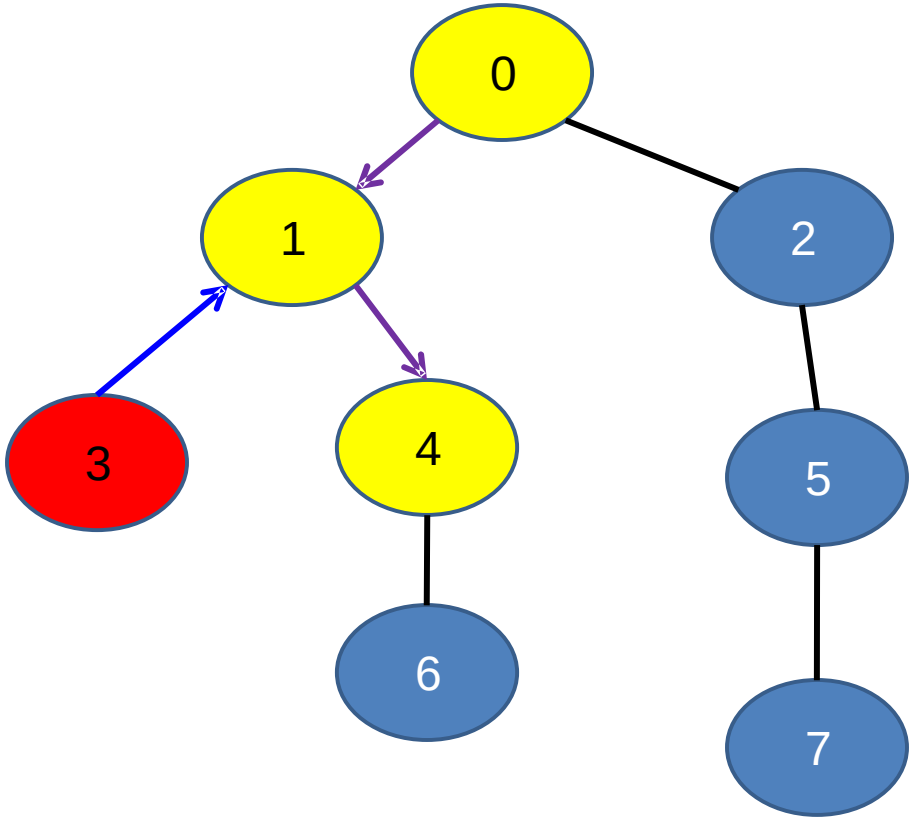
DFS



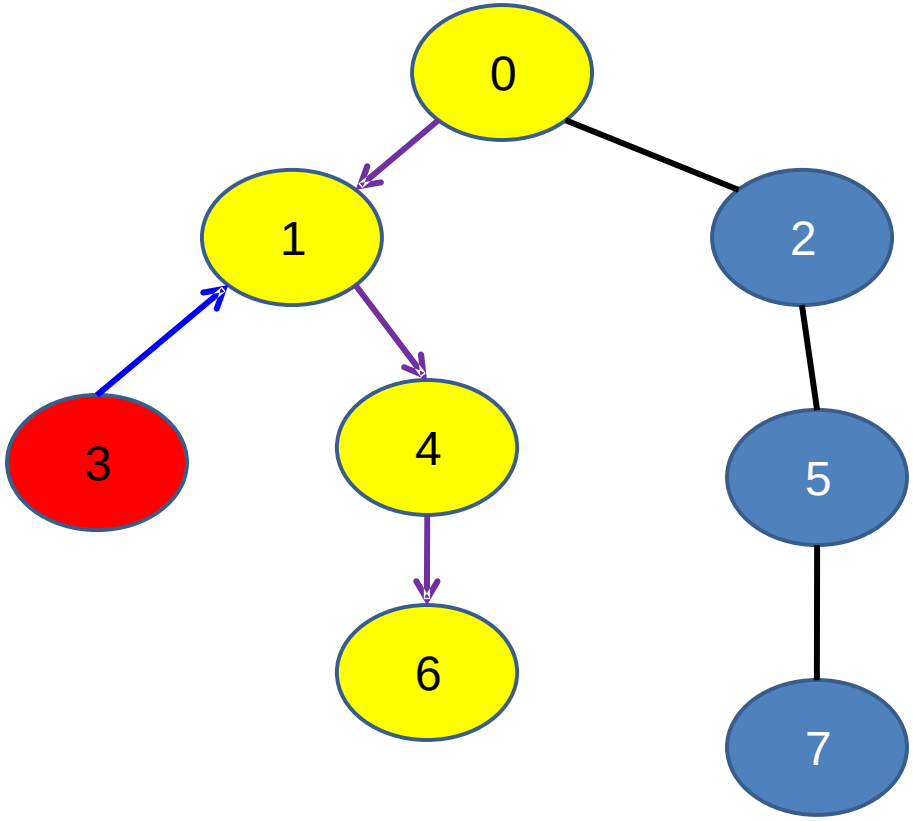
DFS



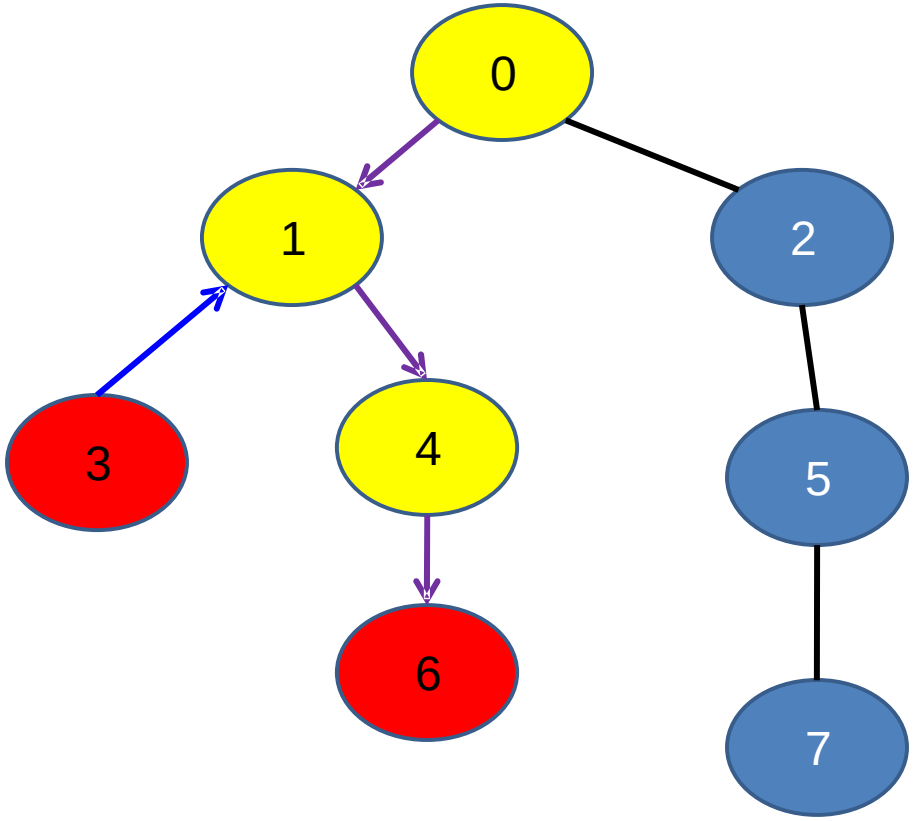
DFS



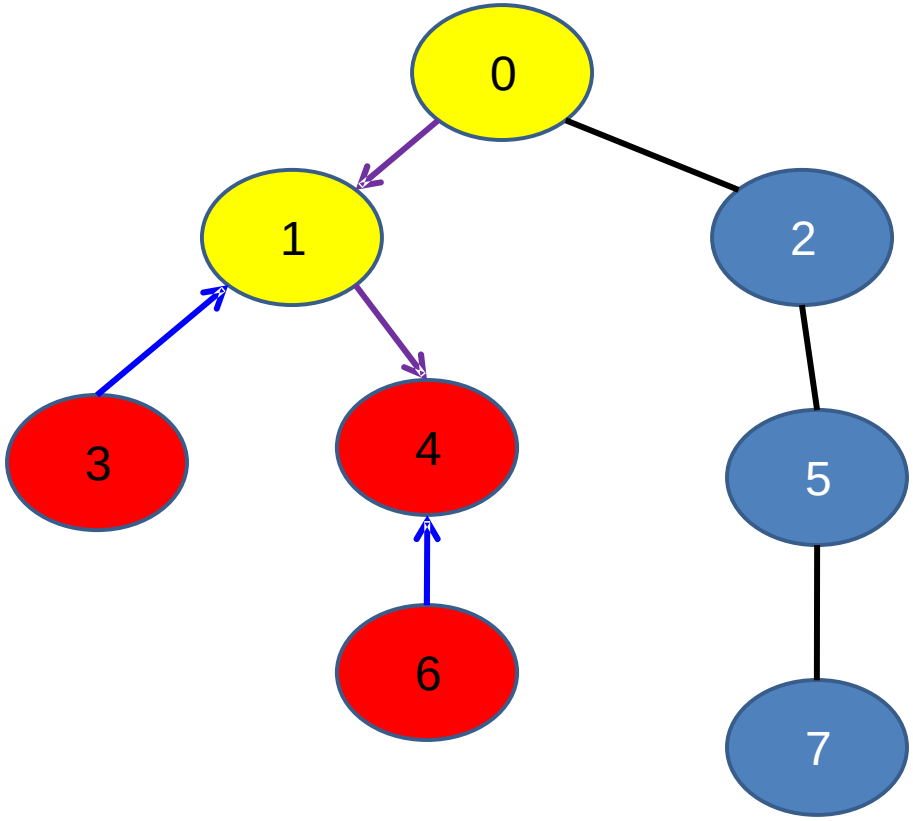
DFS



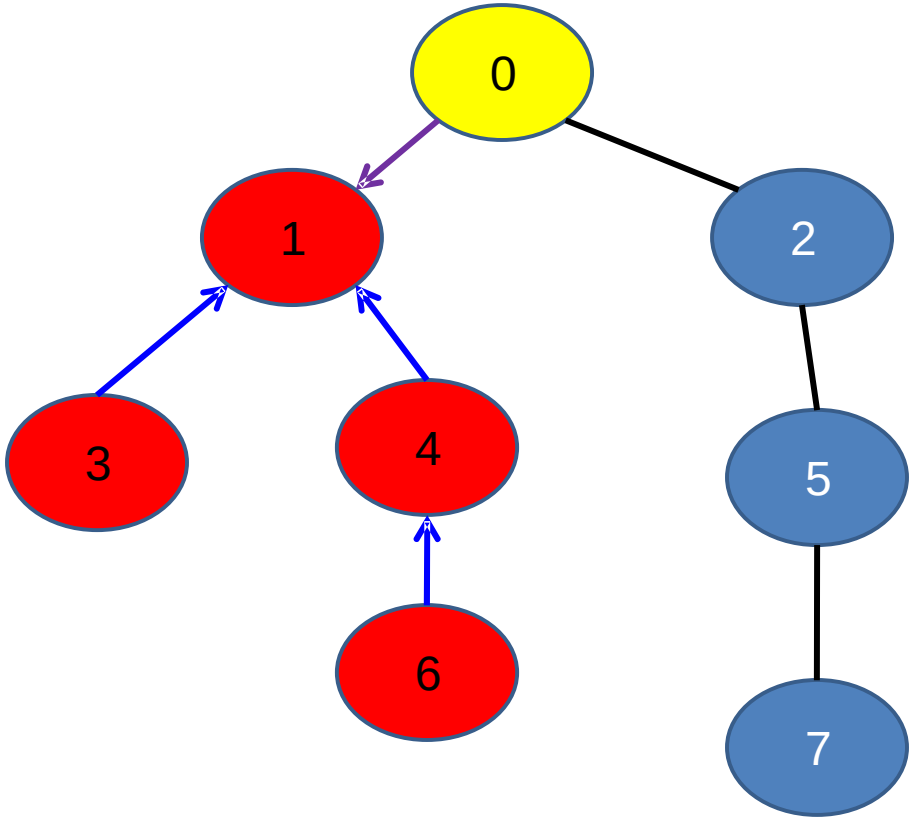
DFS



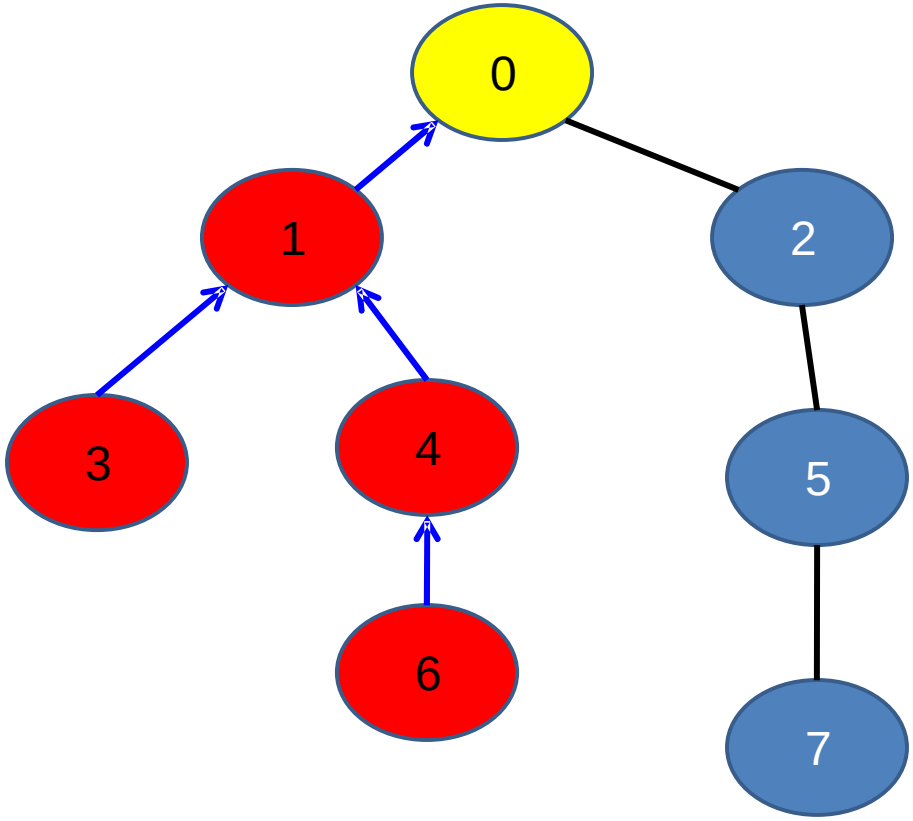
DFS



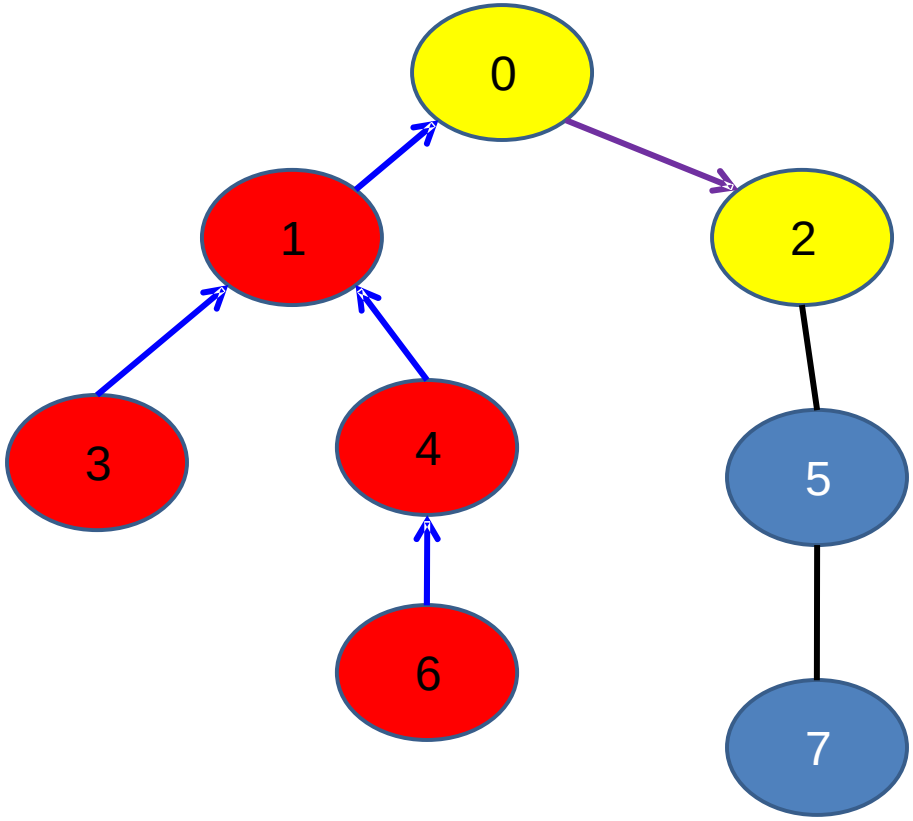
DFS



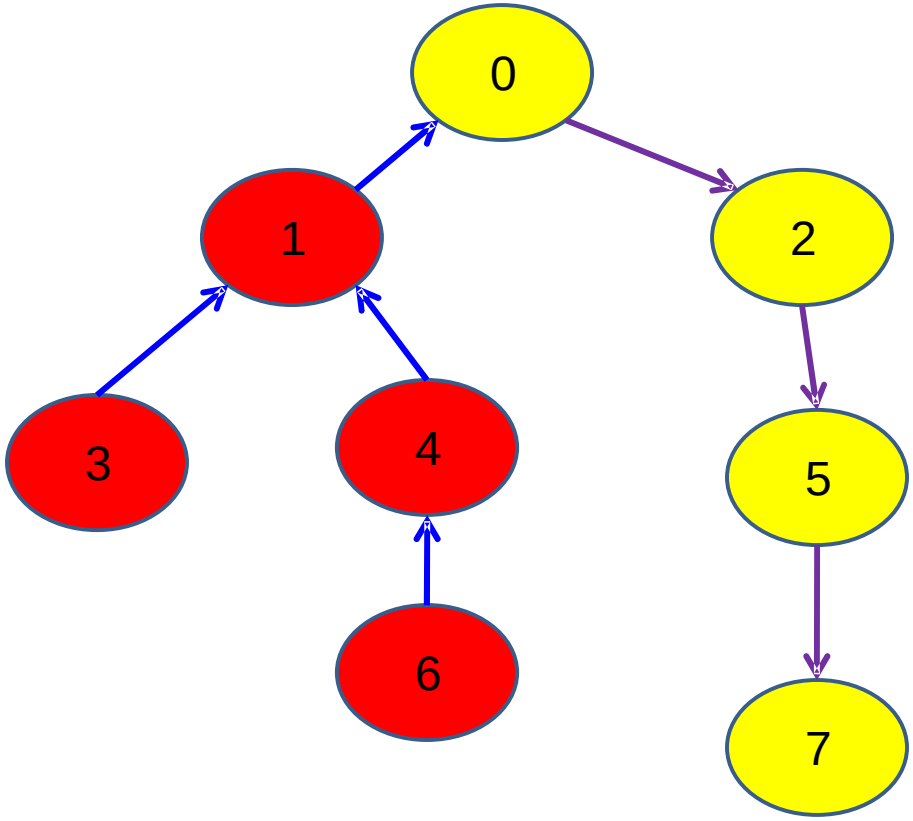
DFS



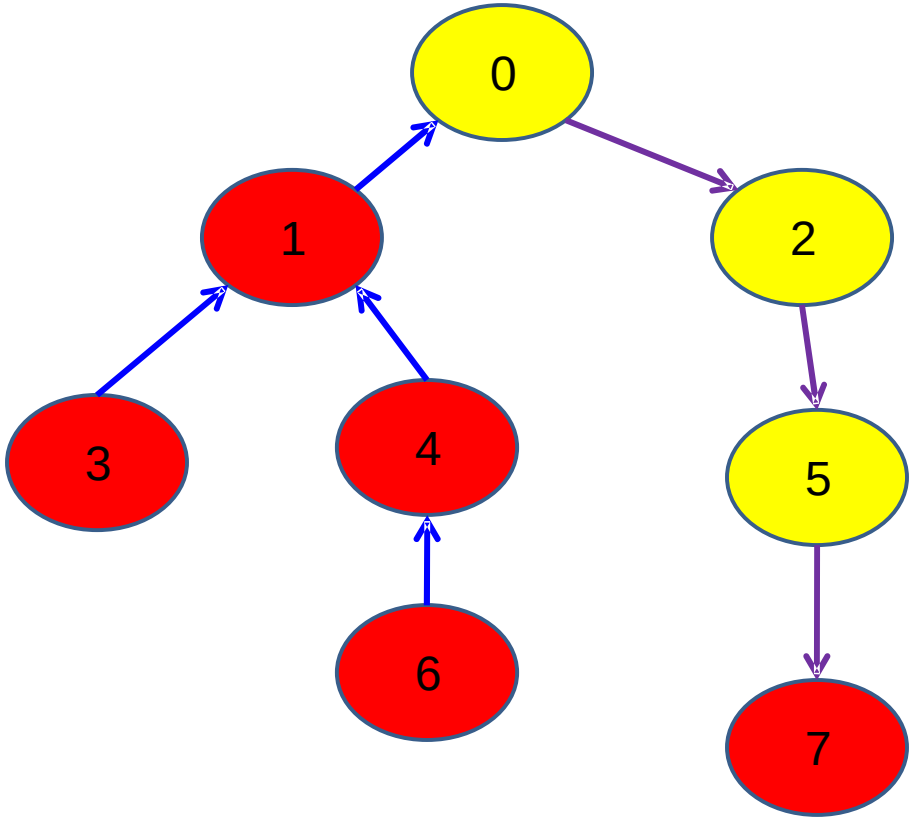
DFS



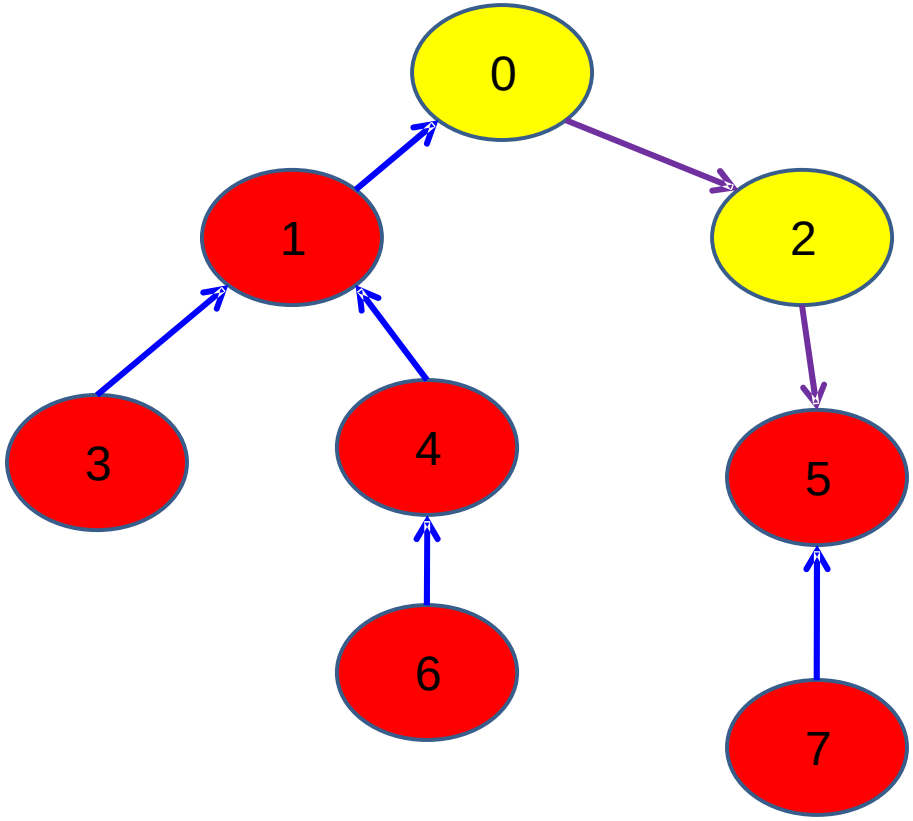
DFS



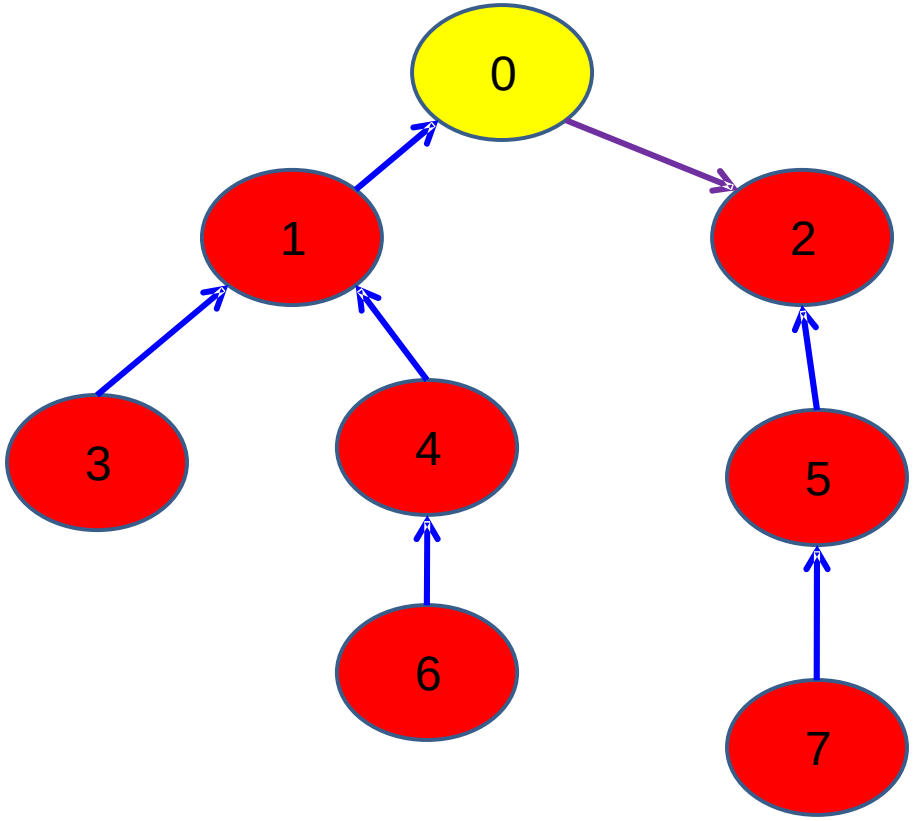
DFS



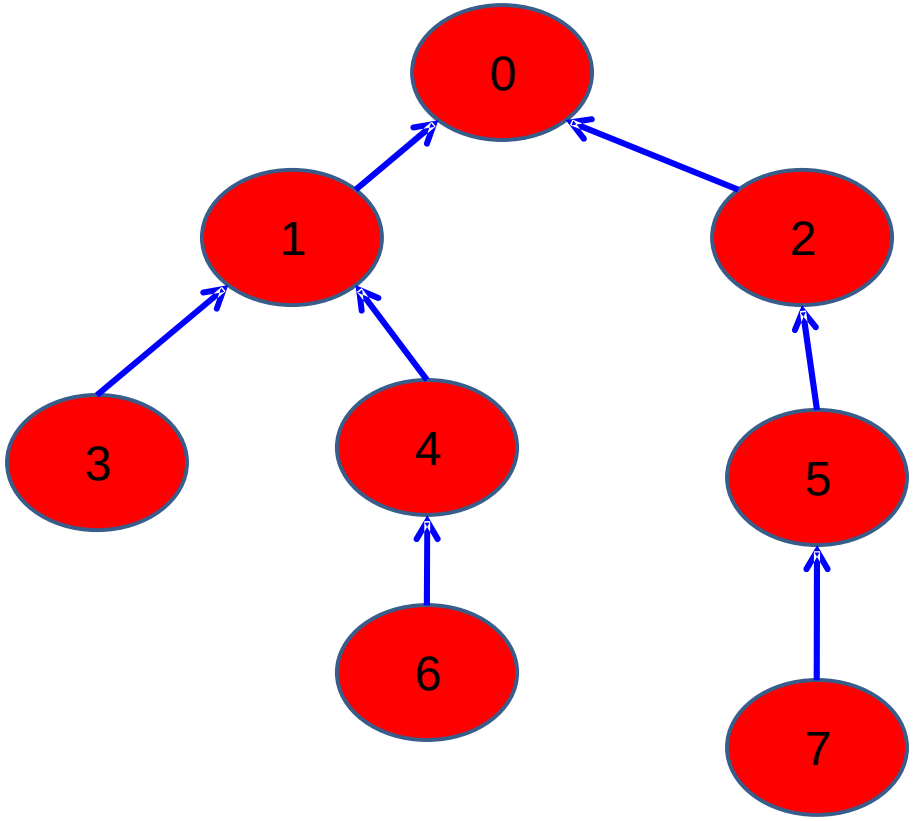
DFS



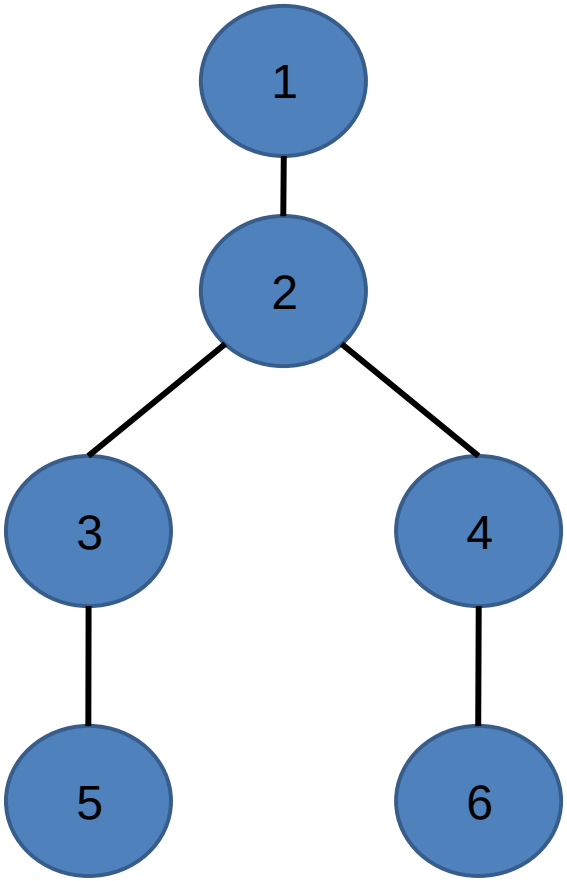
DFS



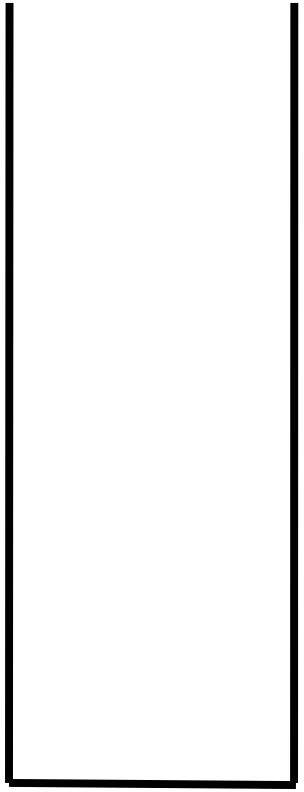
DFS



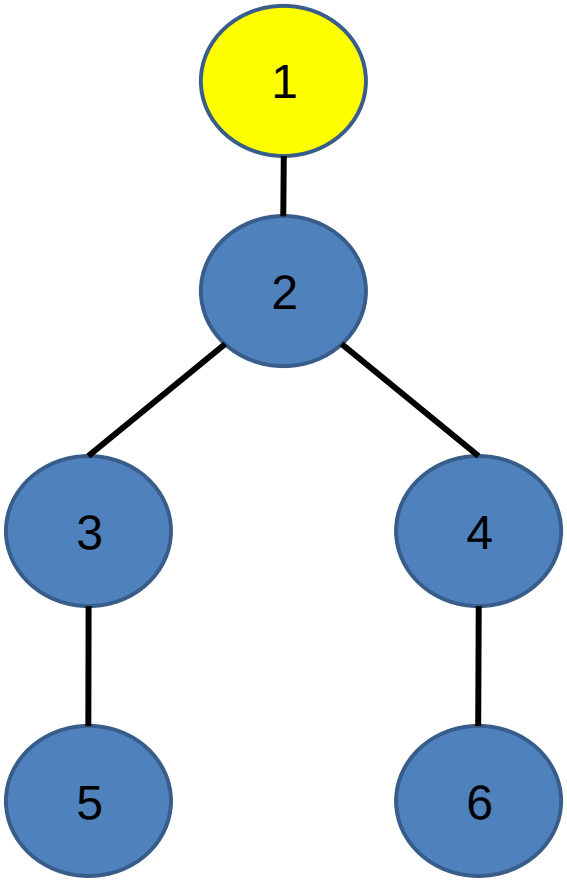
DFS

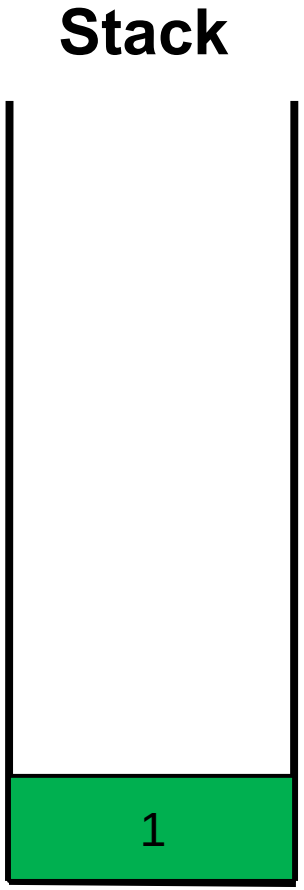


Stack

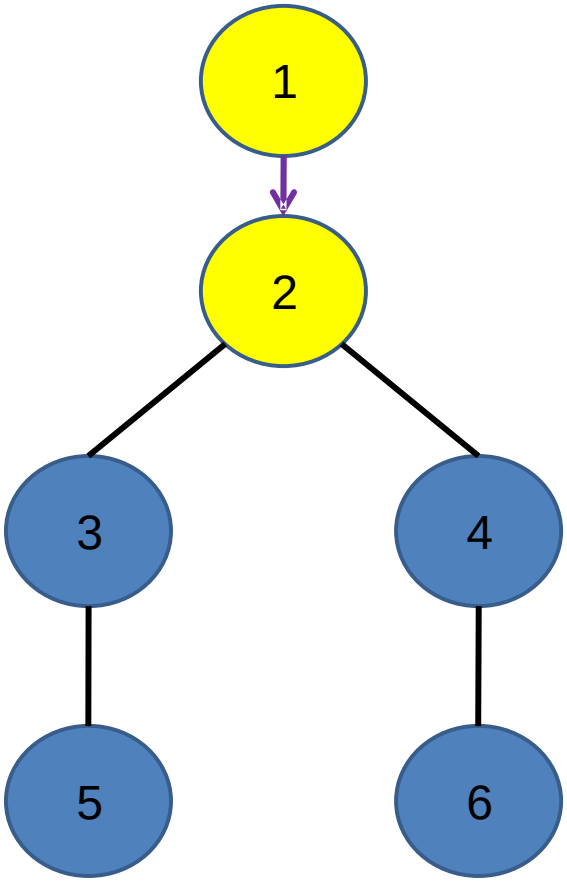


DFS

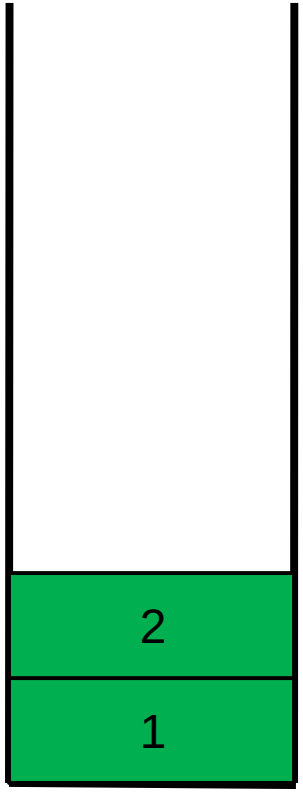




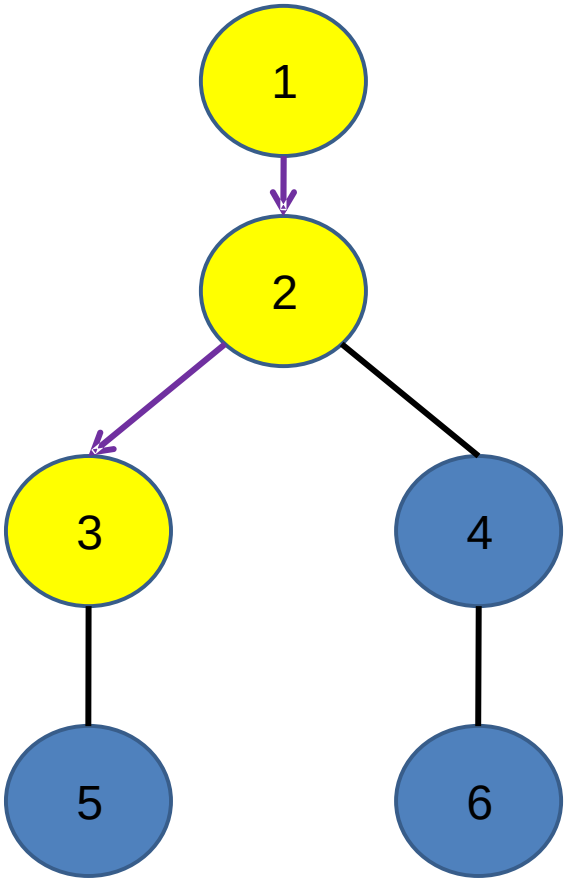
DFS



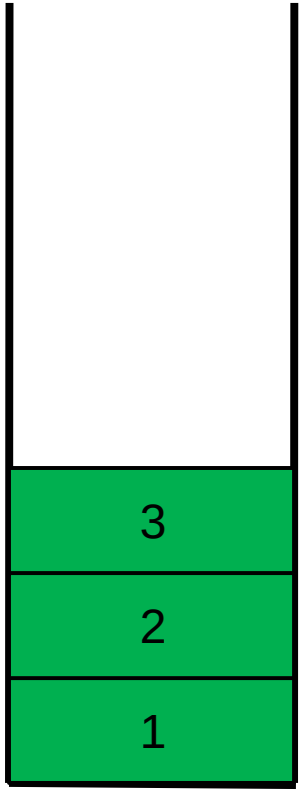
Stack



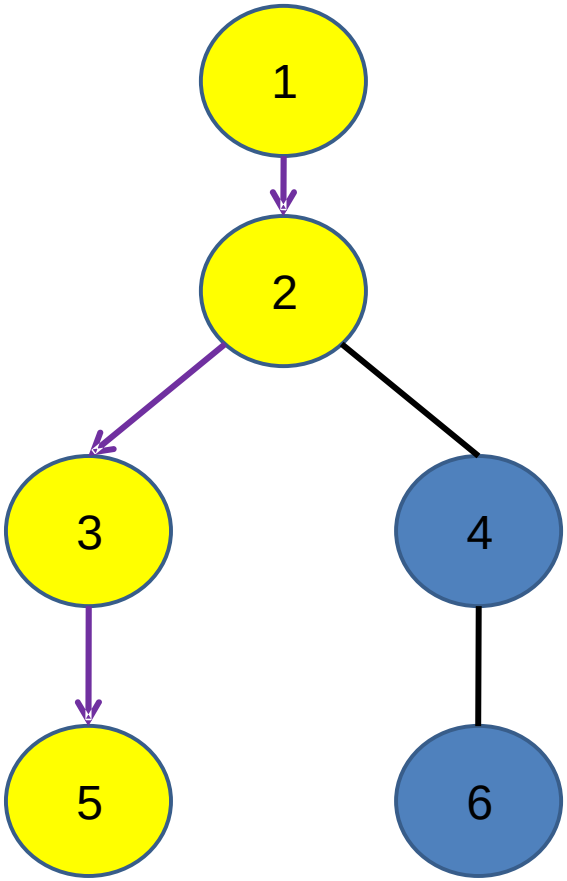
DFS



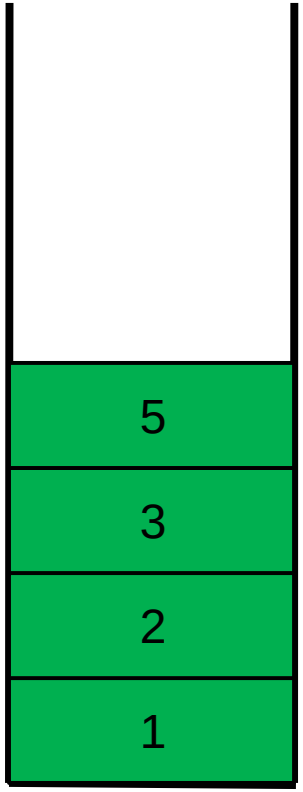
Stack



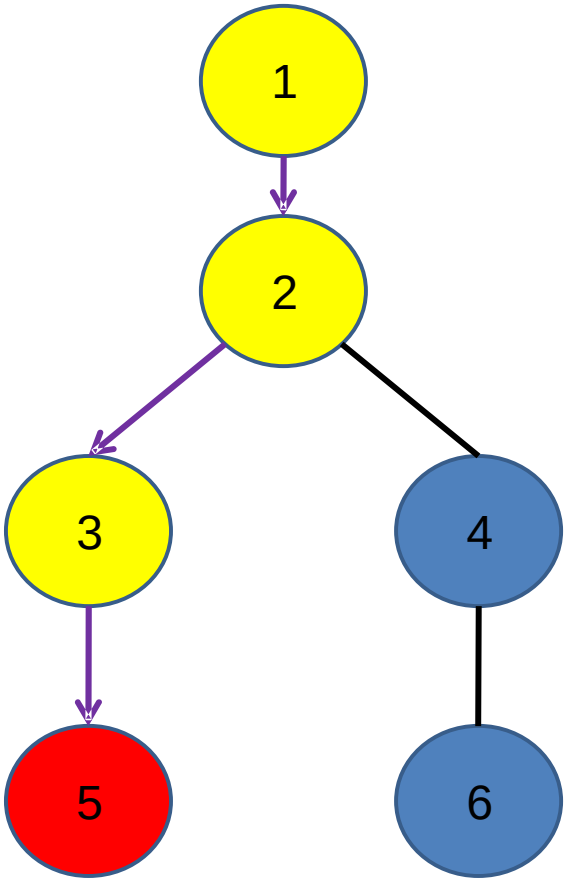
DFS



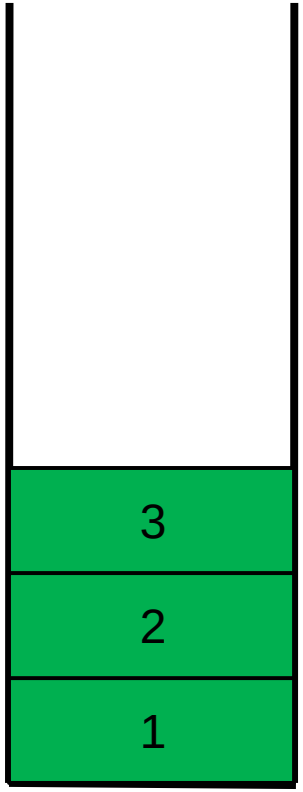
Stack



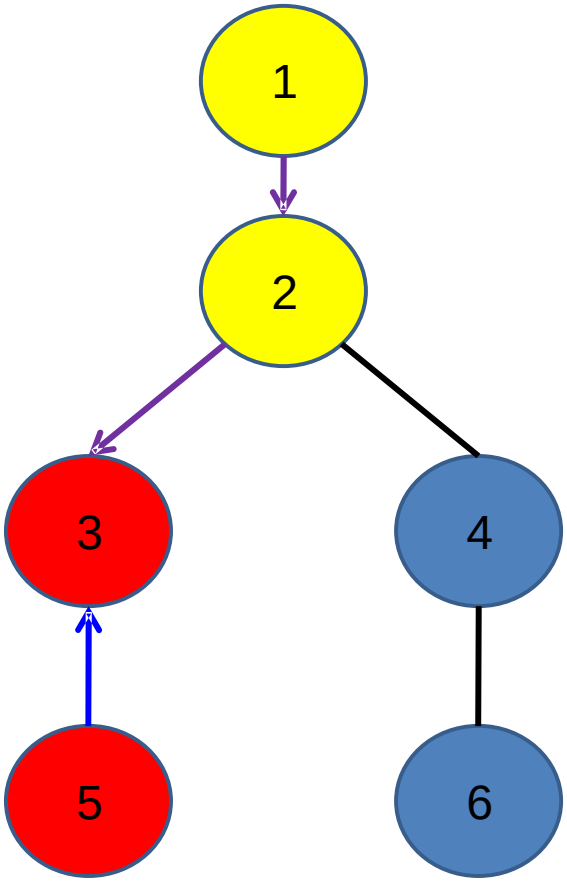
DFS

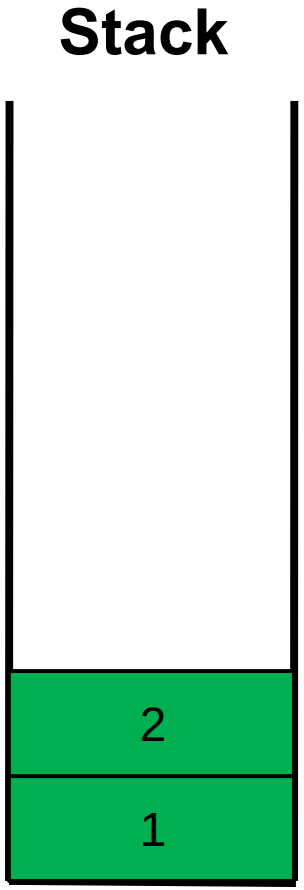


Stack

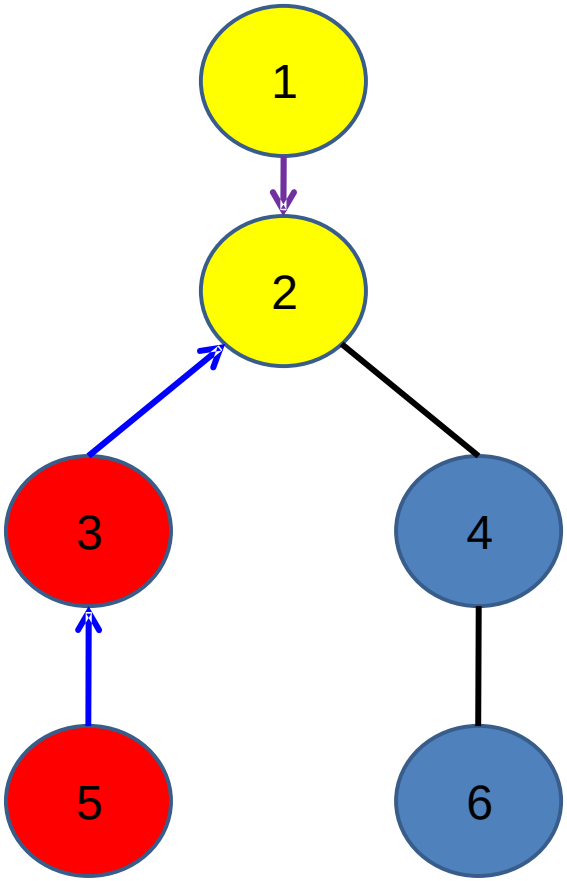


DFS

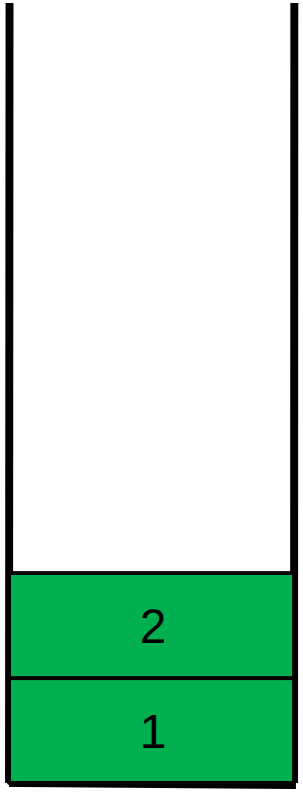




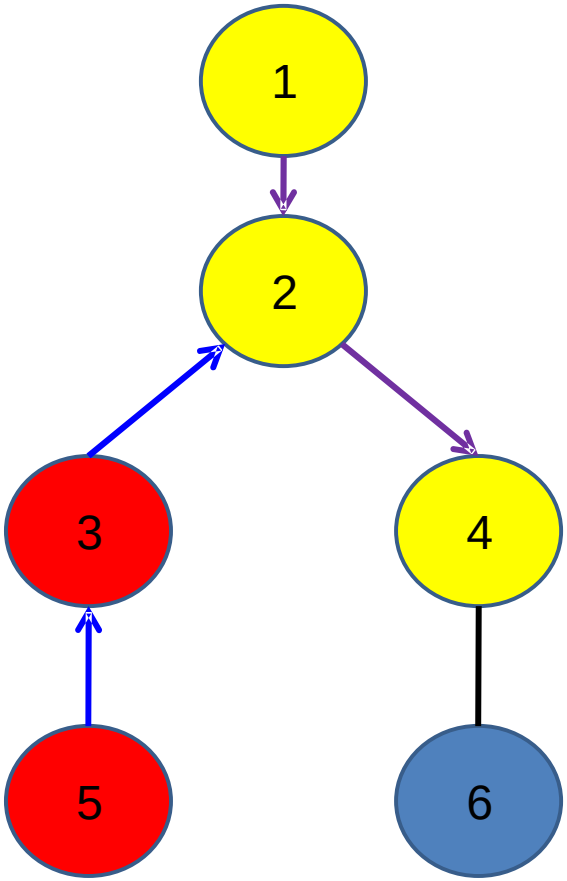
DFS



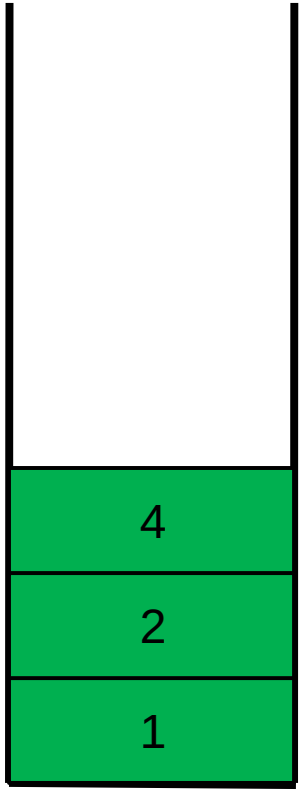
Stack



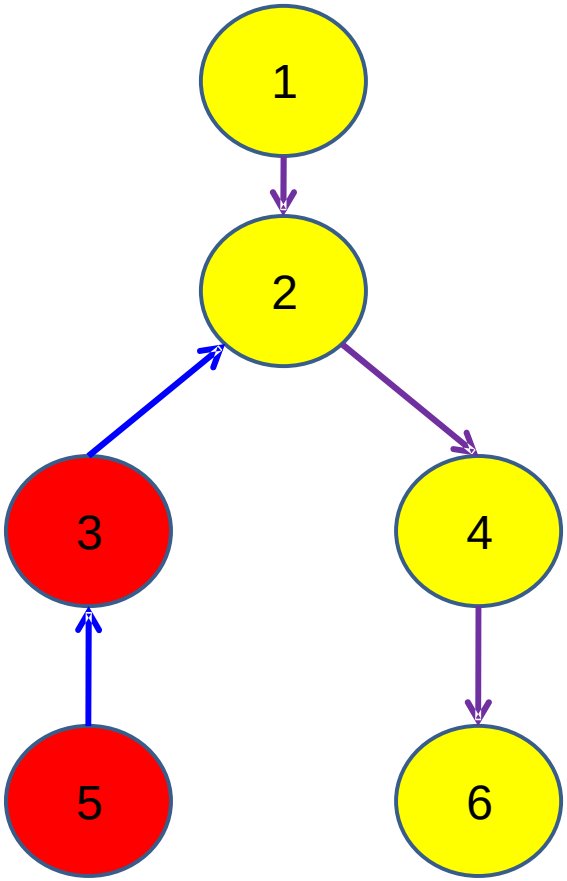
DFS



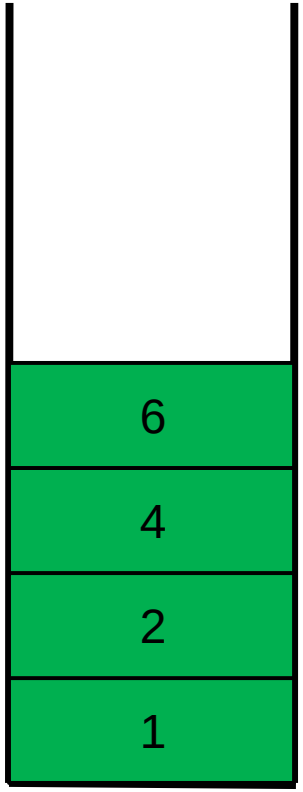
Stack



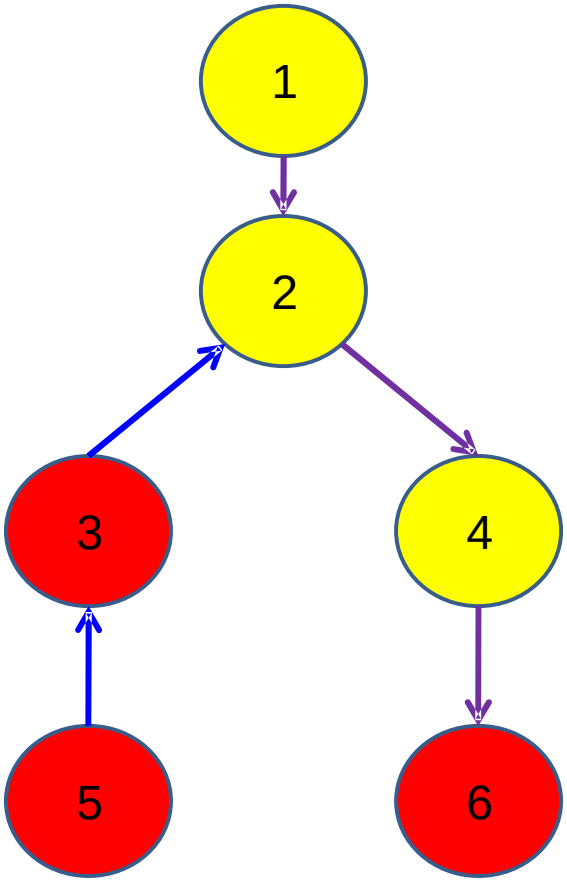
DFS



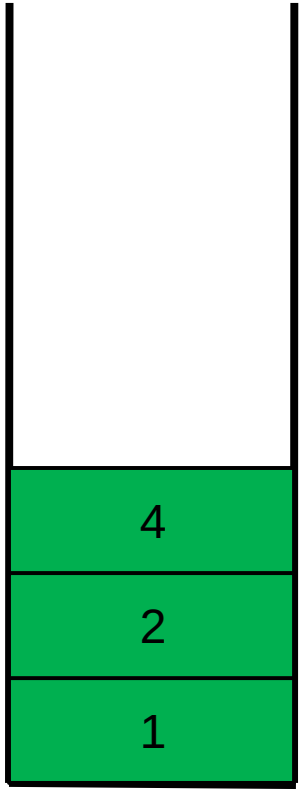
Stack



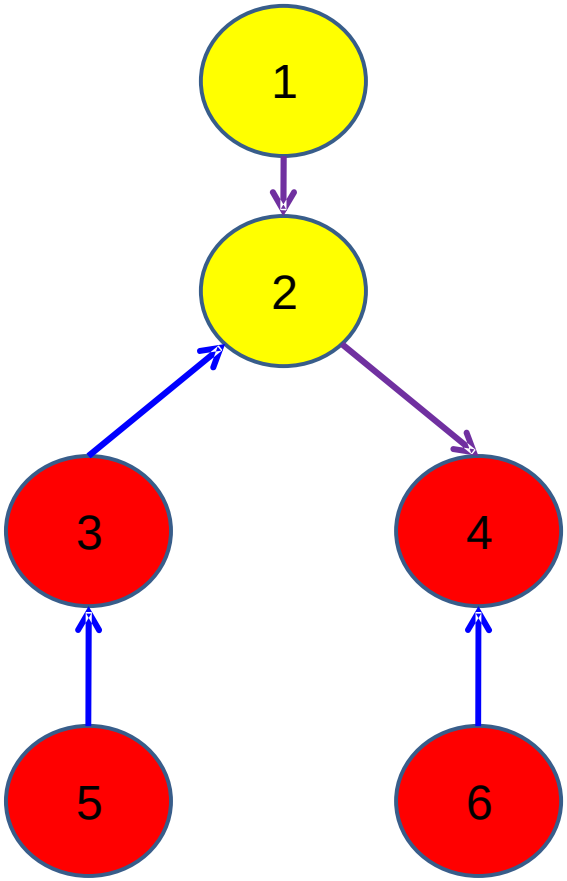
DFS

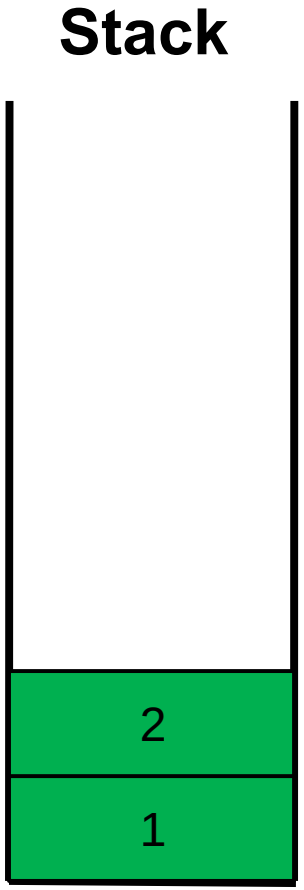


Stack

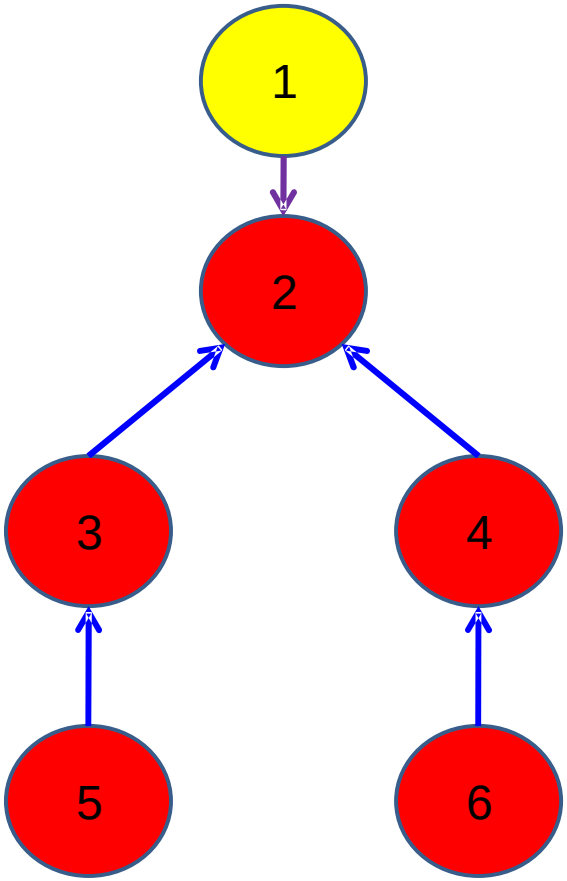


DFS

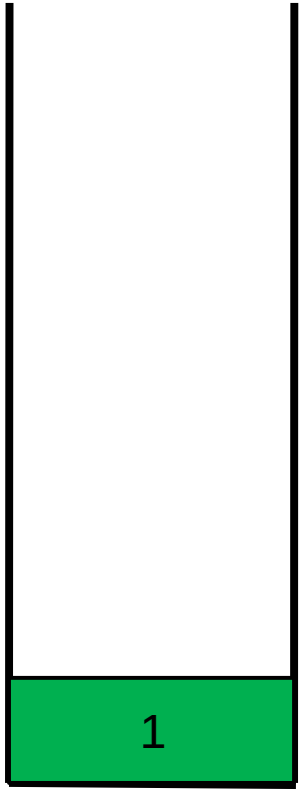




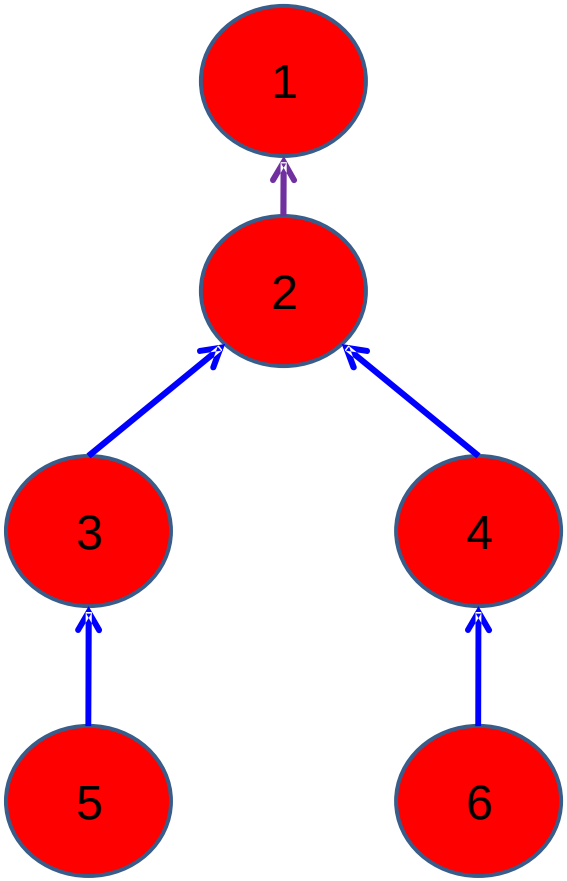
DFS



Stack



DFS



Stack

DFS

- Sourse Code(adjacency list)

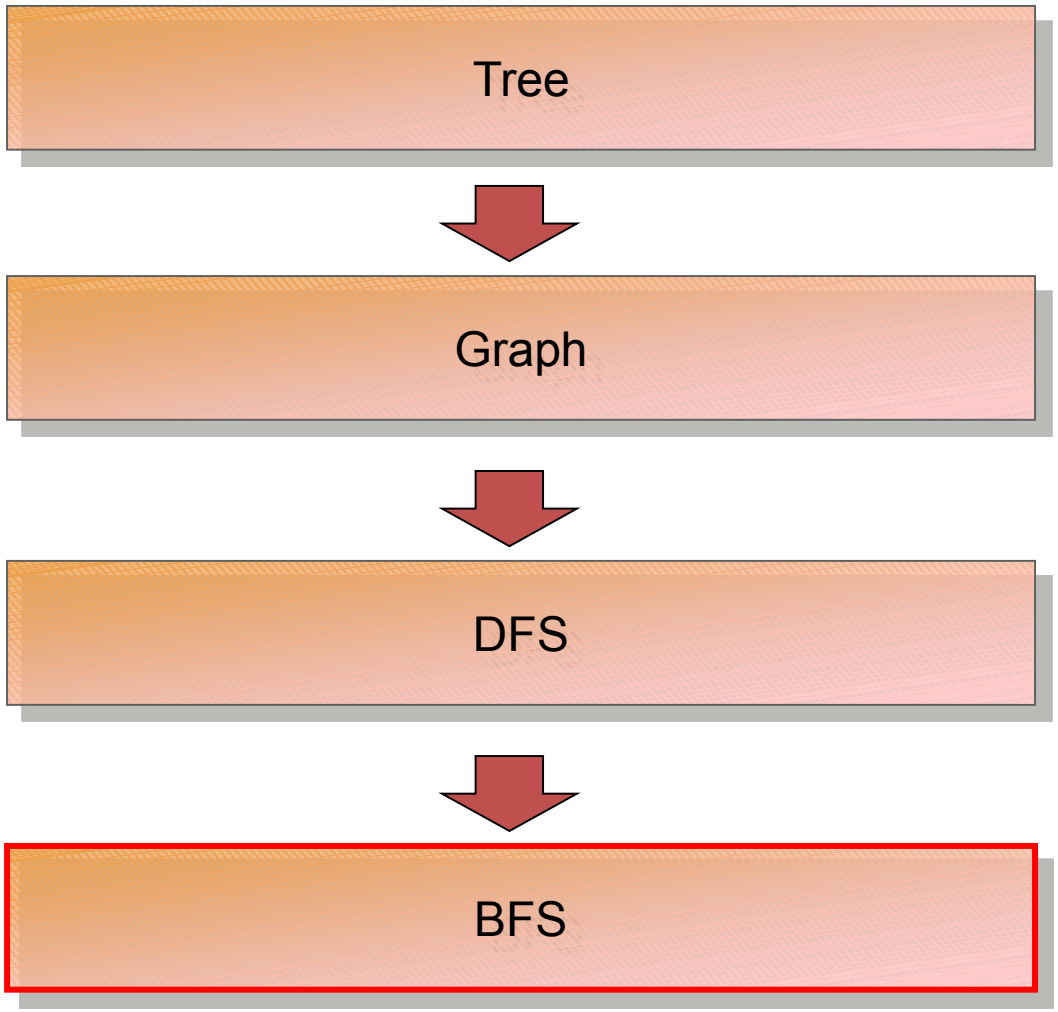
```
void DFS(int cur)
{
    vis[cur]=true;

    for(int i=0;i<adj[cur].size();i++)
    {
        int next=adj[cur][i];
        if(!vis[next])
            DFS(next);
    }
    return;
}
```

DFS

- Practice

[UVA-352] The Seasonal War

Outline



BFS

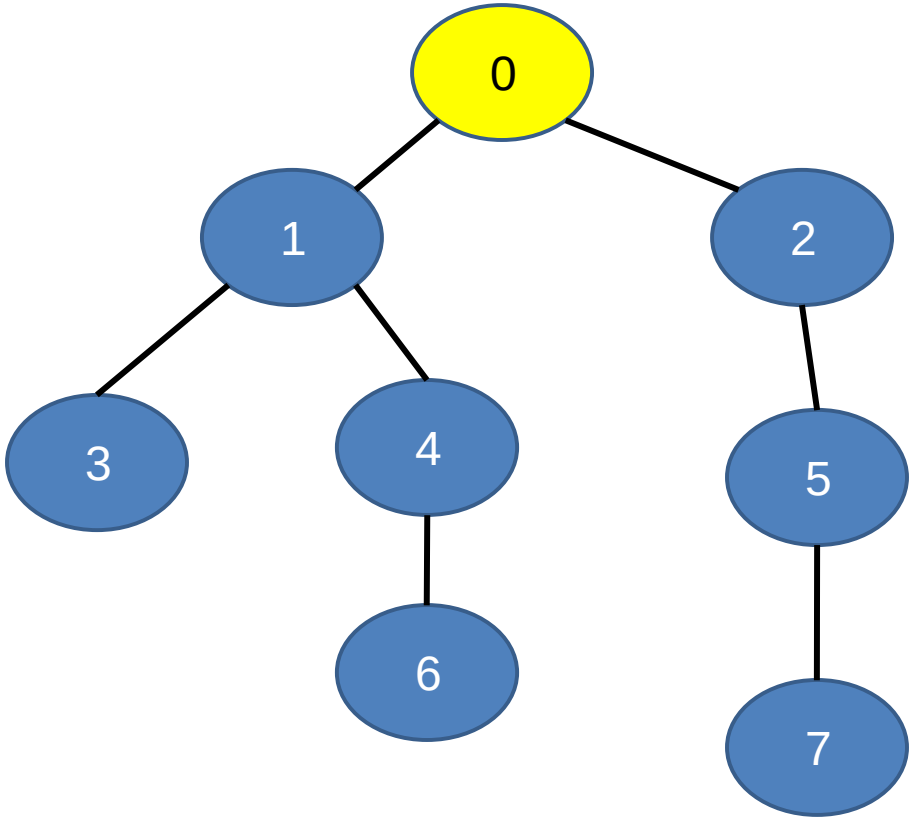
(B)readth-(F)irst-(S)earch

BFS

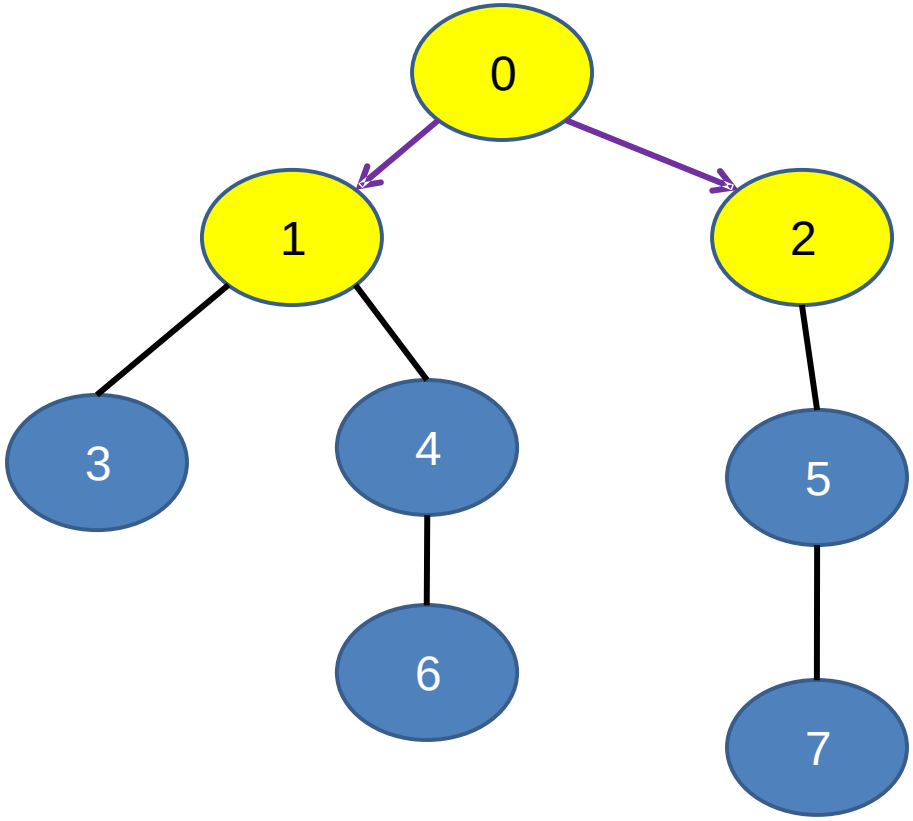
(B)readth-(F)irst-(S)earch

Queue

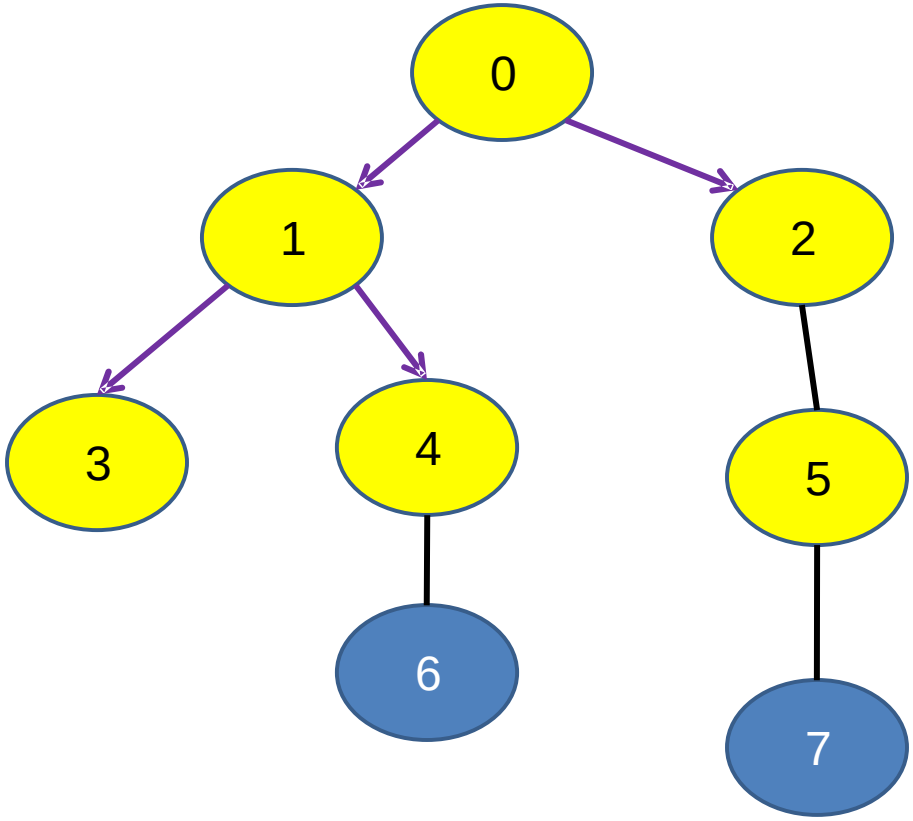
BFS



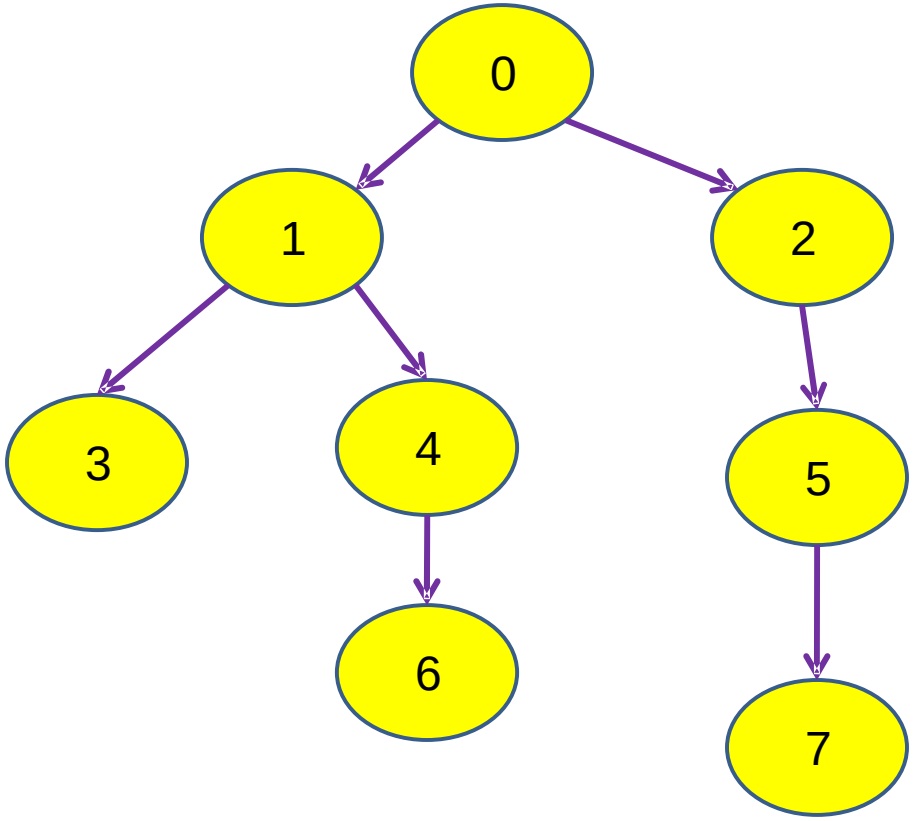
BFS



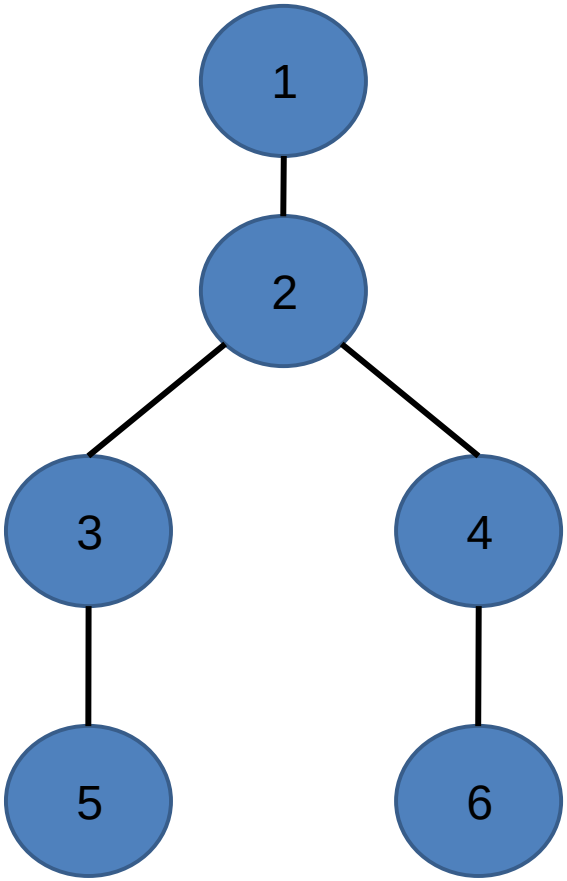
BFS



BFS

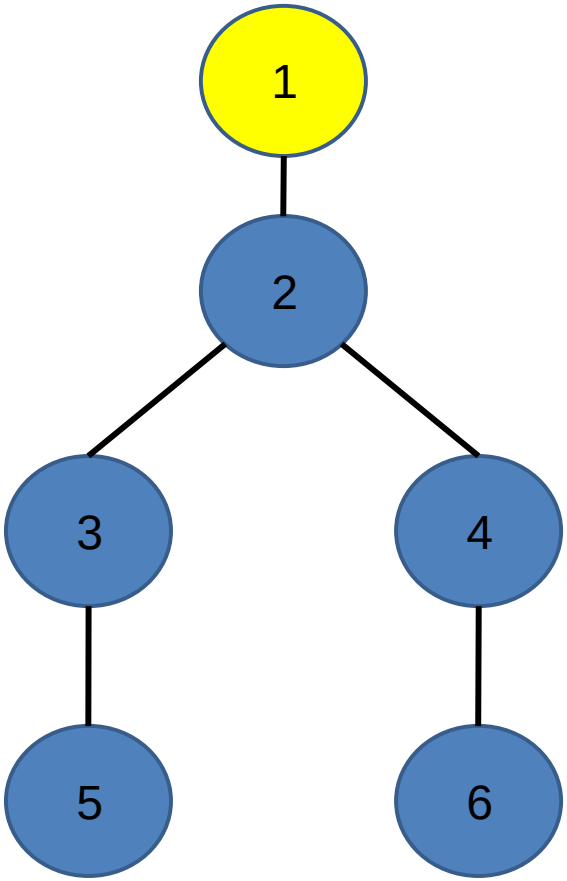


BFS



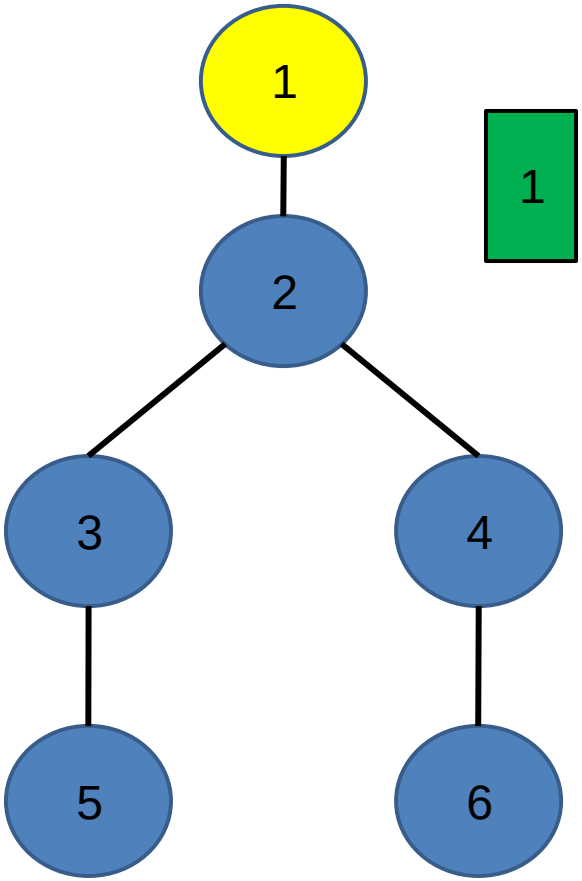
Queue

BFS



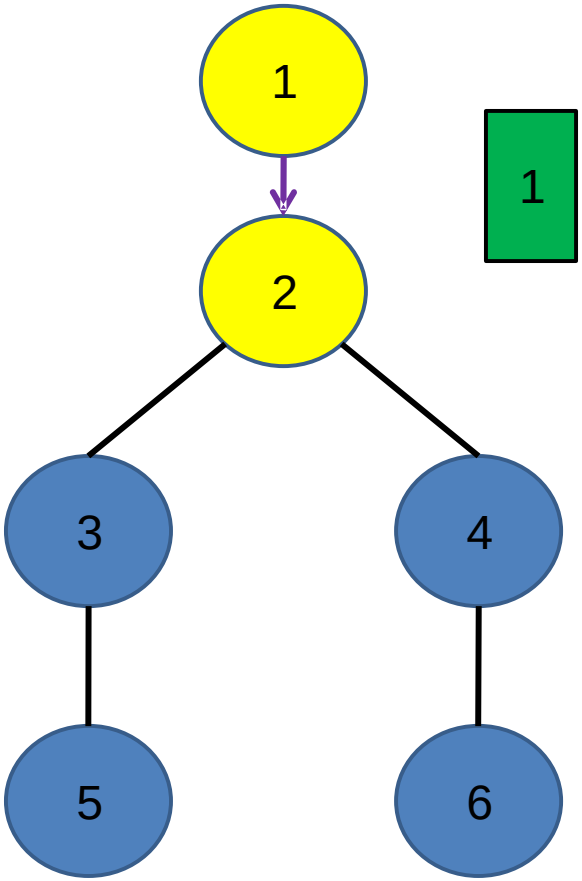
Queue

BFS



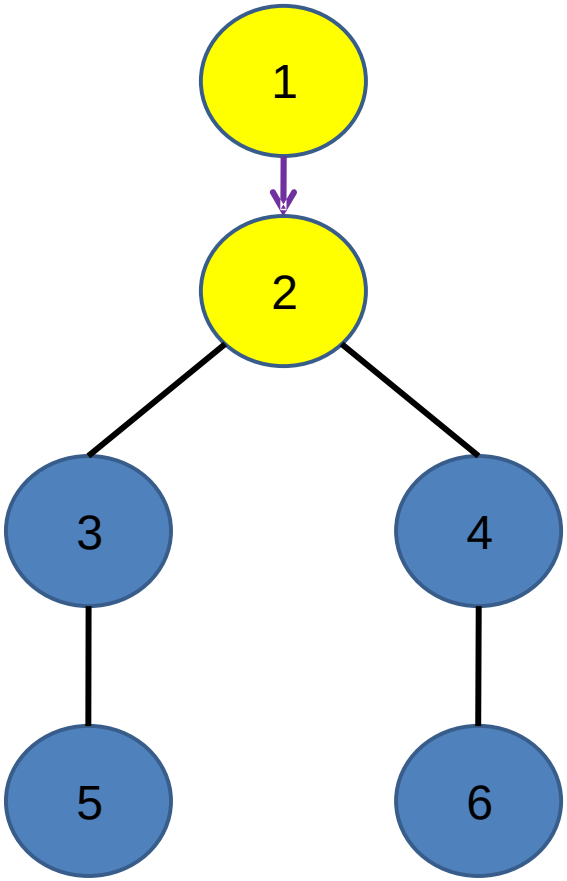
Queue

BFS



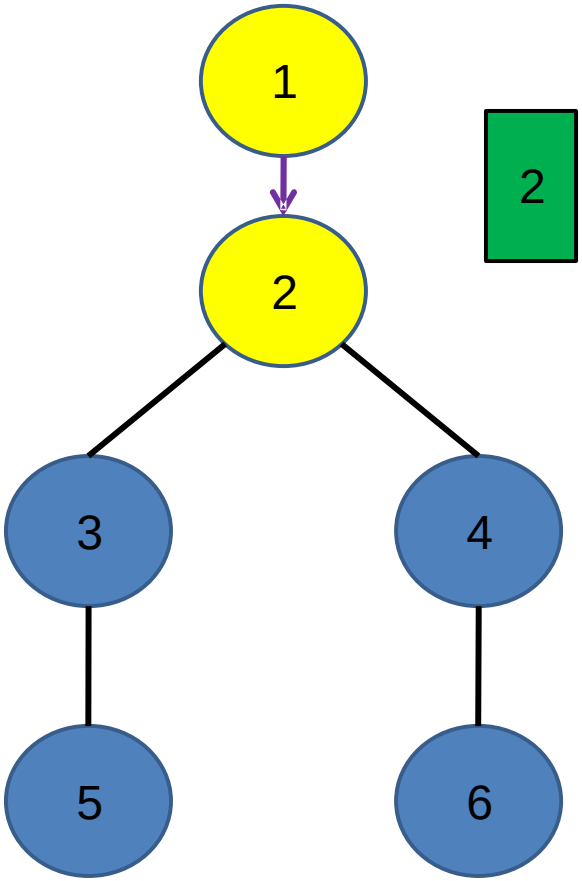
Queue

BFS



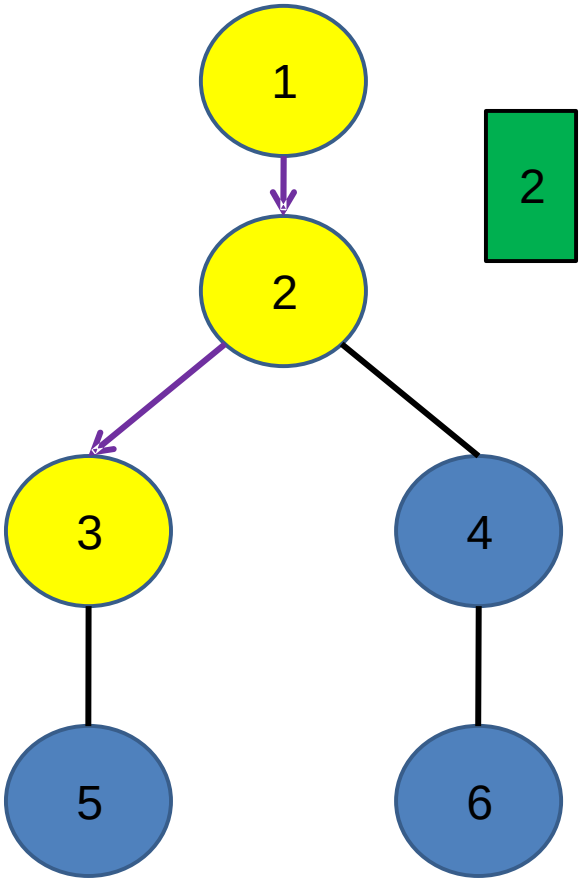
Queue

BFS



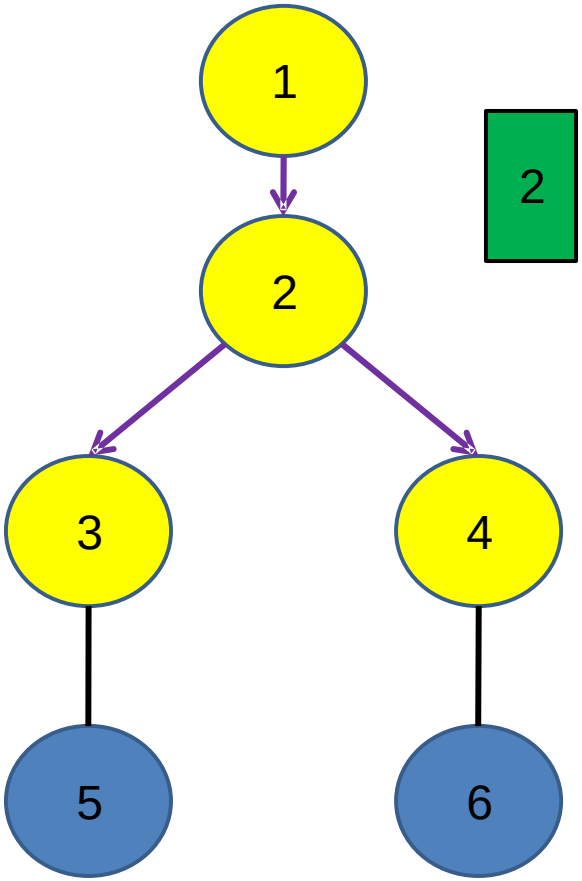
Queue

BFS



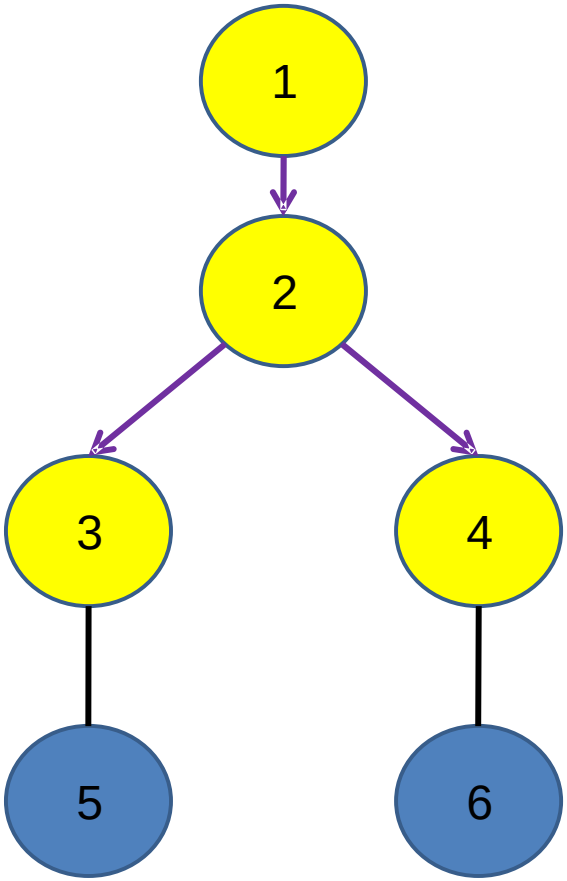
Queue

BFS



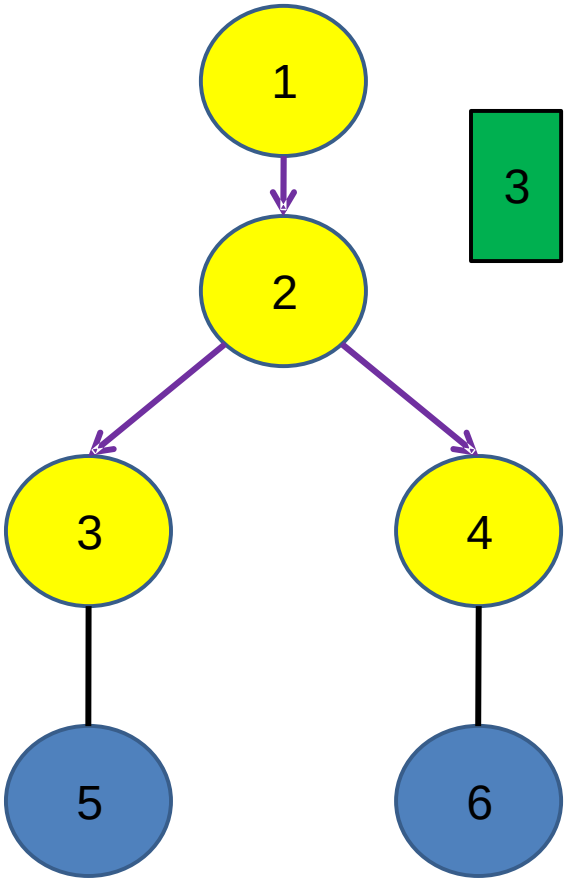
Queue

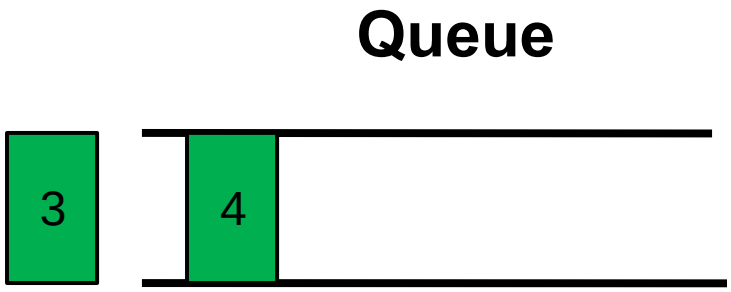
BFS



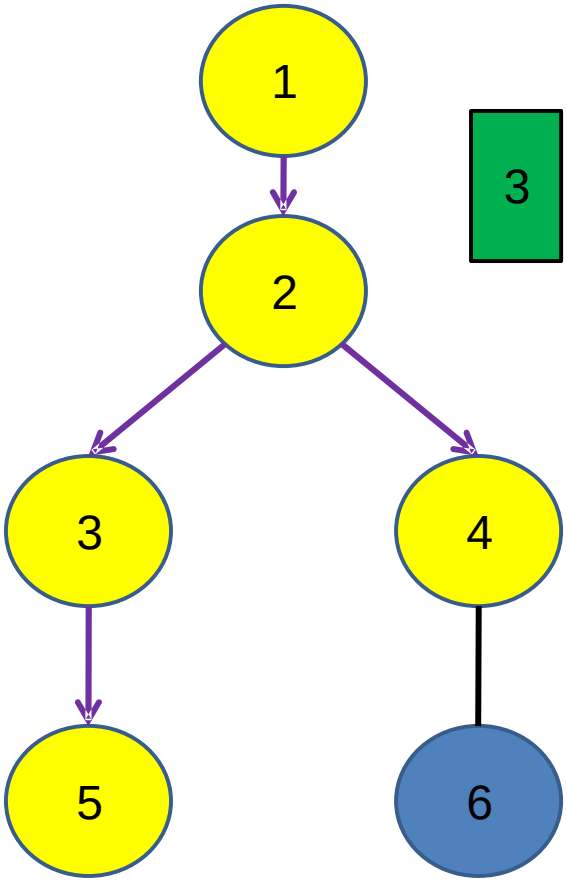
Queue

BFS

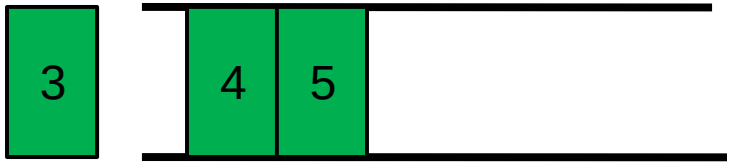




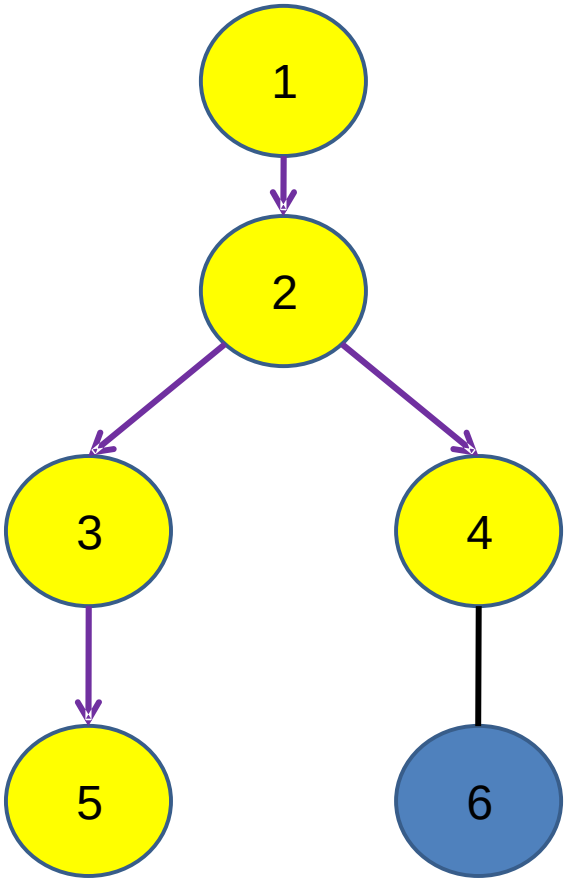
BFS



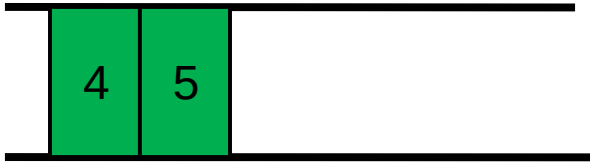
Queue



BFS

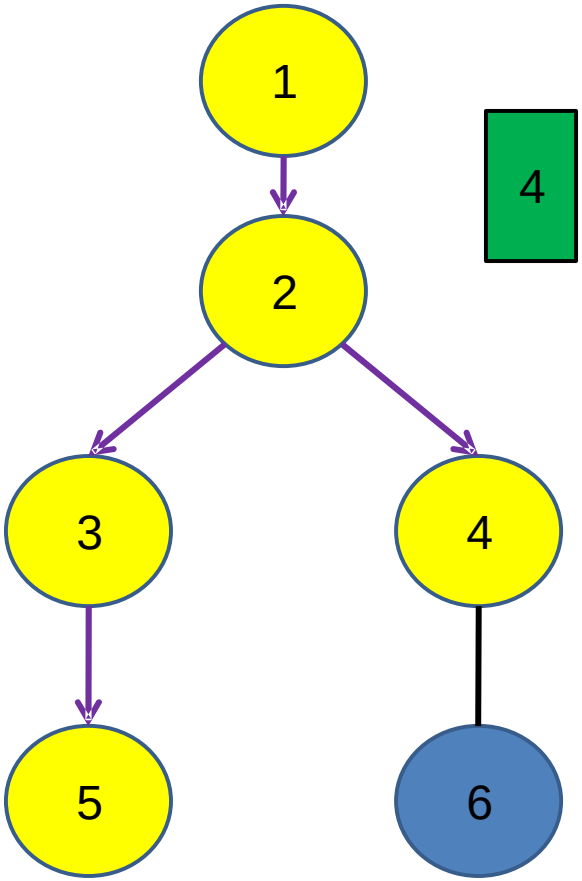


Queue



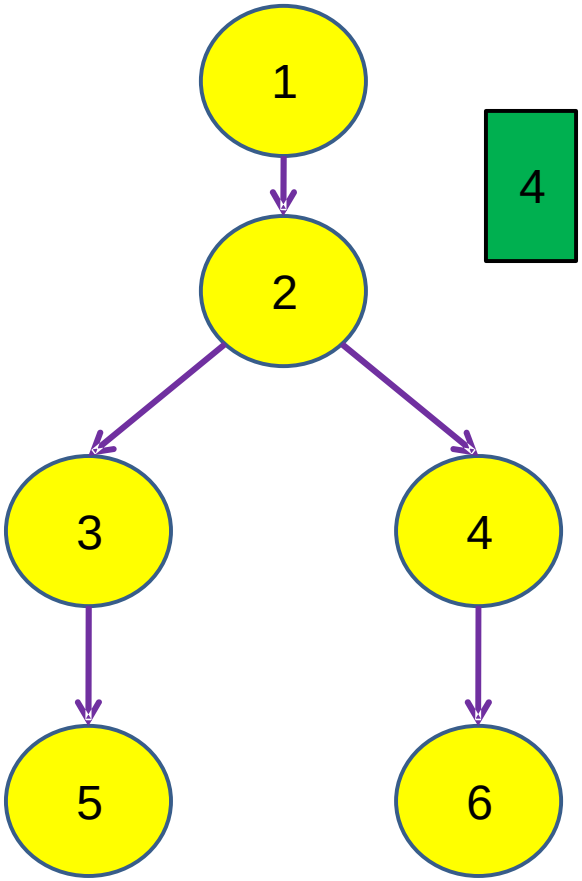
Made By louis7340

BFS

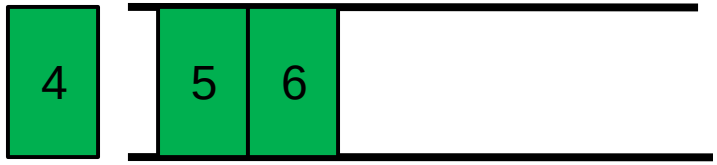


Queue

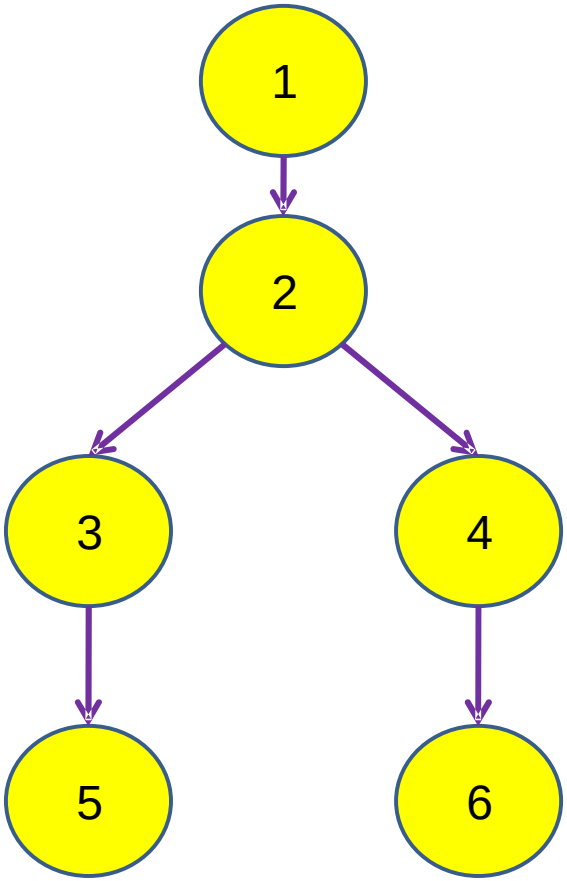
BFS



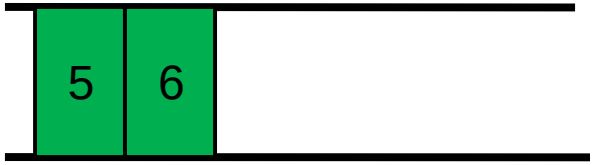
Queue



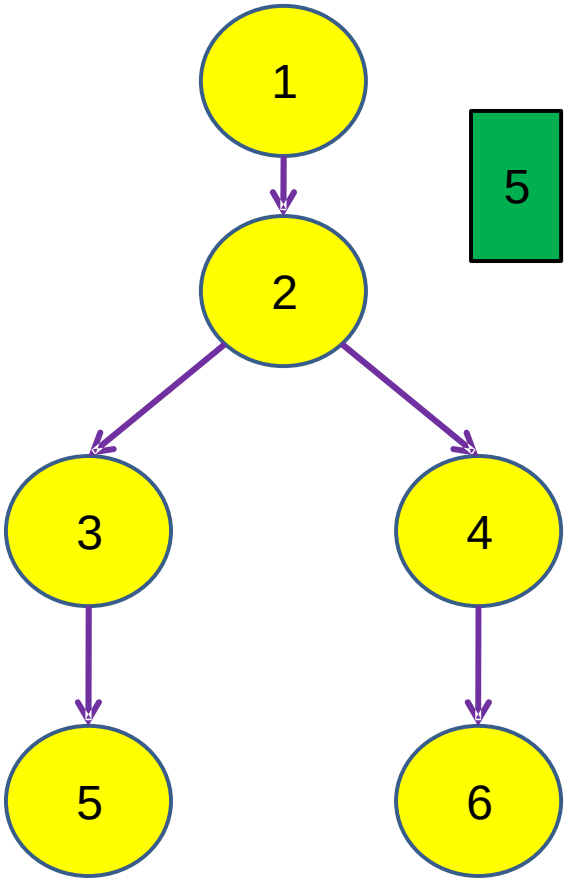
BFS



Queue

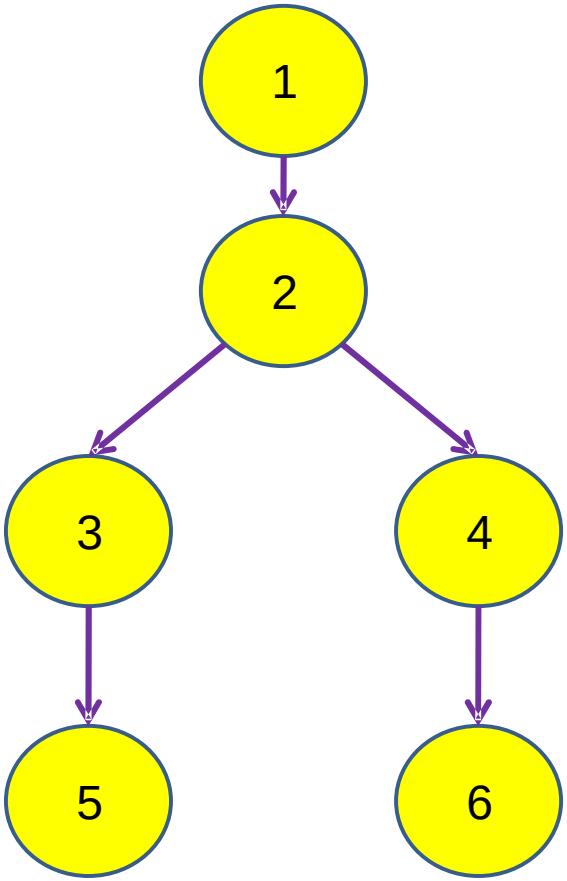


BFS



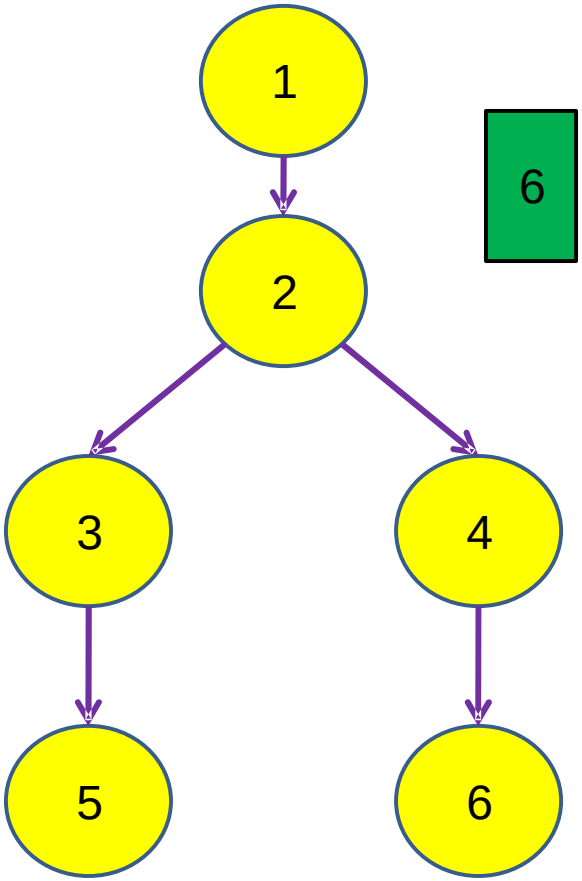
Queue

BFS



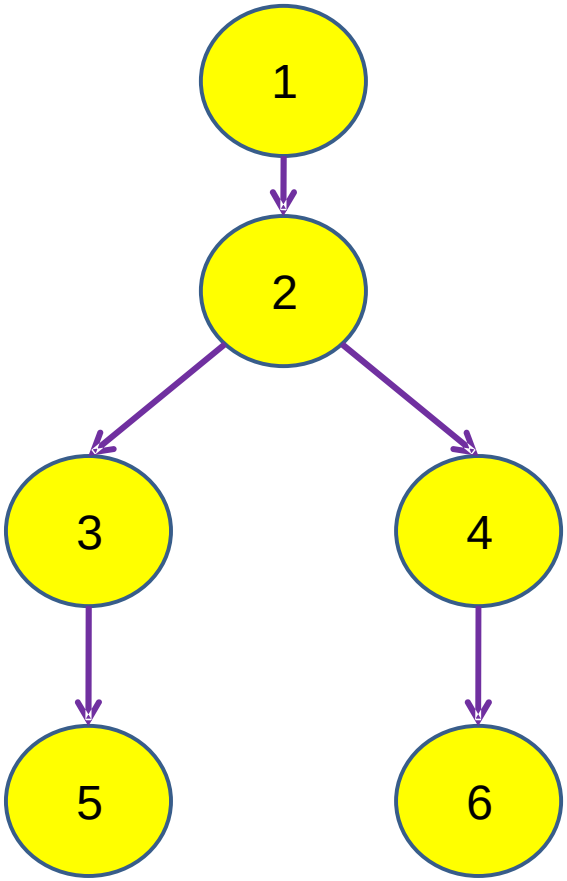
Queue

BFS



Queue

BFS



Queue

BFS

- **Source Code** (adjacency list)

```
void BFS(int root)
{
    queue<int> que;
    que.push(root);
    visited[root]=true;

    while(!que.empty())
    {
        int cur=que.front();
        que.pop();

        for(int i=0;i<adj[cur].size();i++)
        {
            int next=adj[cur][i];
            if(!visited[next])
            {
                visited[next]=true;
                que.push(next);
            }
        }
    }
    return;
}
```

BFS

- **Practice**

[UVA-532]　Dungeon Master

- **Skill:**

  int dir[4][2]={ {1,0},{-1,0},{0,1},{0,-1} }

  Ⓟ search four directions

- **Skill:**

```
int dir[4][2]={ {1,0},{-1,0},{0,1},{0,-1} };

void DFS(int cur_x,int cur_y)
{
    vis[cur_x][cur_y]=1;

    for(int i=0;i<4;i++)
    {
        int nx=cur_x+dir[i][0];
        int ny=cur_y+dir[i][1];

        //watch for boundary
        if( !vis[nx][ny])
            DFS(nx,ny);
    }
}
```

Time Complexity

DFS O(V + E)

BFS O(V + E)

V: the number of nodes

E: the number of edges

(adjacency list)

HW5

Toatlly 28 problems

UVa:

336,352,383,532,567,571,601,705,

762,10004,10009,10474, 10505,10592,10603,

10946, 11624

POJ:

1129,1154,1416,1606,1753,1915,1979,2243

Thank for Your Attention

Made By louis7340