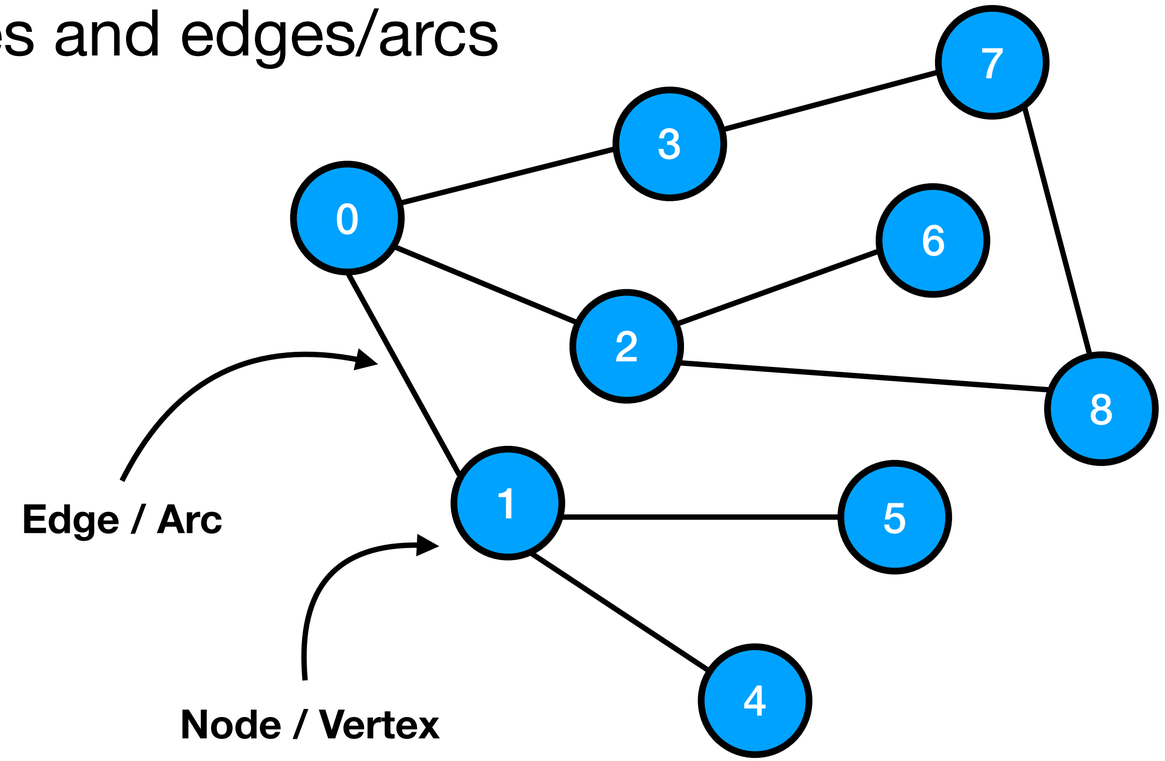
Code Prime Theory

Graph Traversals

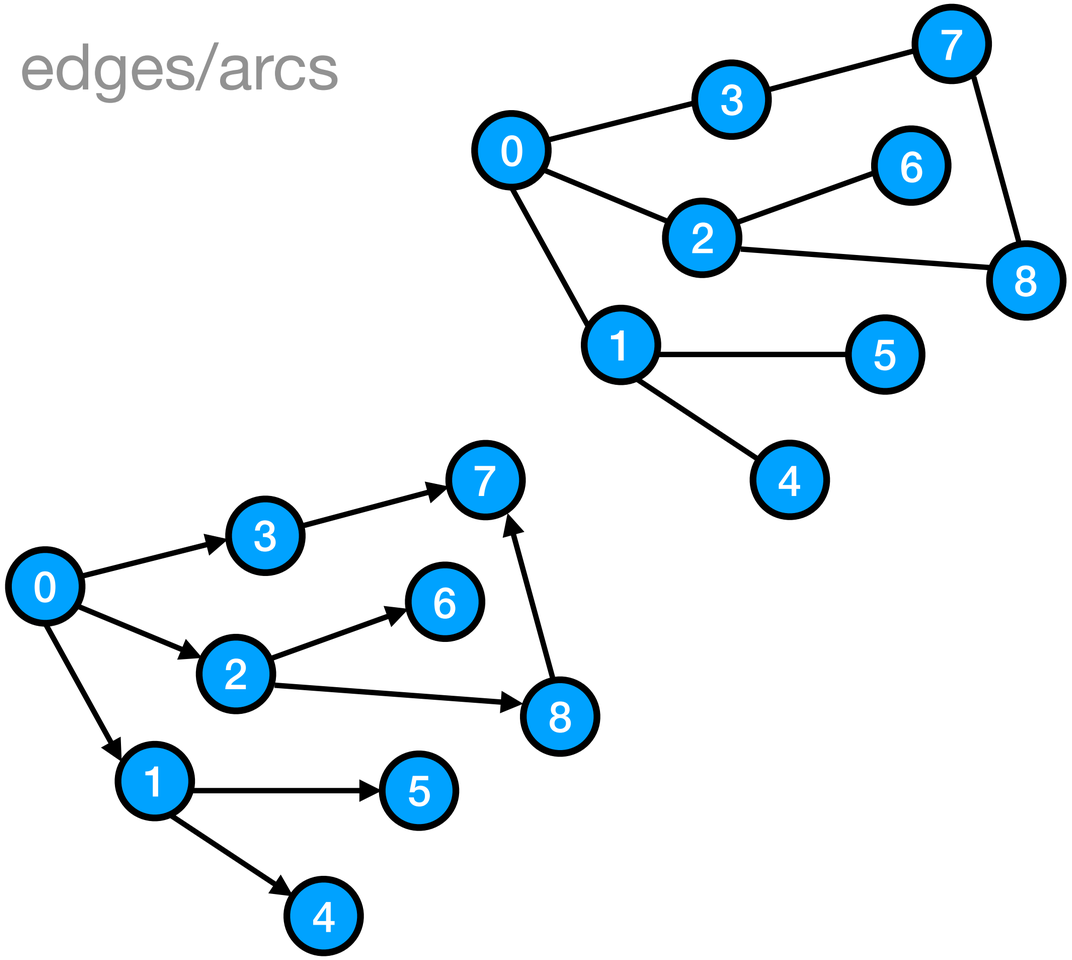
Recap

- Graph consists of vertices/nodes and edges/arcs
- Undirected/Directed graph
- Weighted graph
- Graph representation



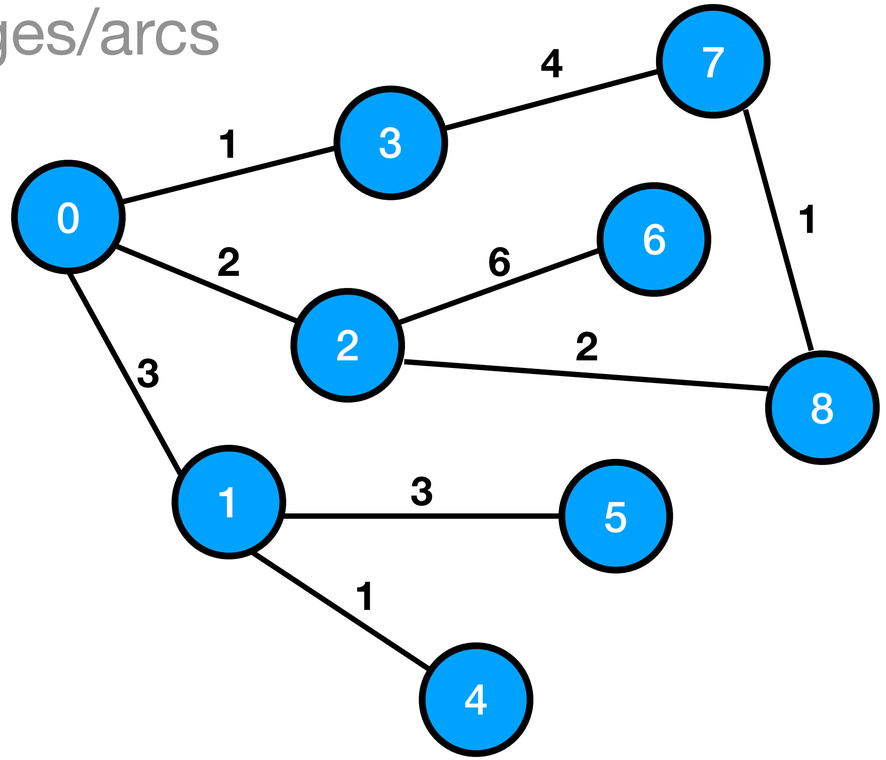
Recap

- Graph consists of vertices/nodes and edges/arcs
- Undirected/Directed graph
- Weighted graph
- Graph representation



Recap

- Graph consists of vertices/nodes and edges/arcs
- Undirected/Directed graph
- Weighted graph
- Graph representation

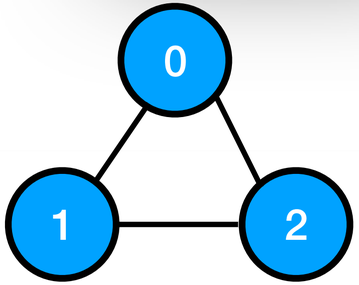


Recap

- Graph consists of vertices/nodes and edges/arcs
- Undirected/Directed graph
- Weighted graph
- Graph representation

```
Edge list
[
  [0, 1],
  [1, 2],
  [2, 0]
]
```

```
Adjacency metrix
[
  [0, 1, 1],
  [1, 0, 1],
  [1, 1, 0],
]
```



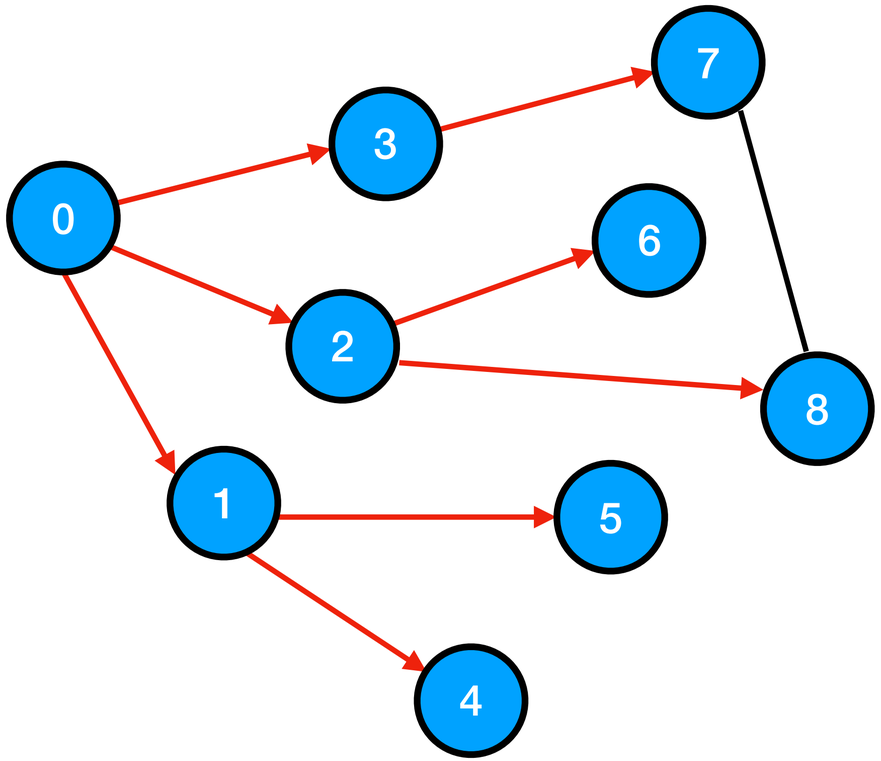

```
Adjacency list
[
  [1, 2],
  [0, 2],
  [0, 1],
]
```

Graph Traversals

Concept and Implementation

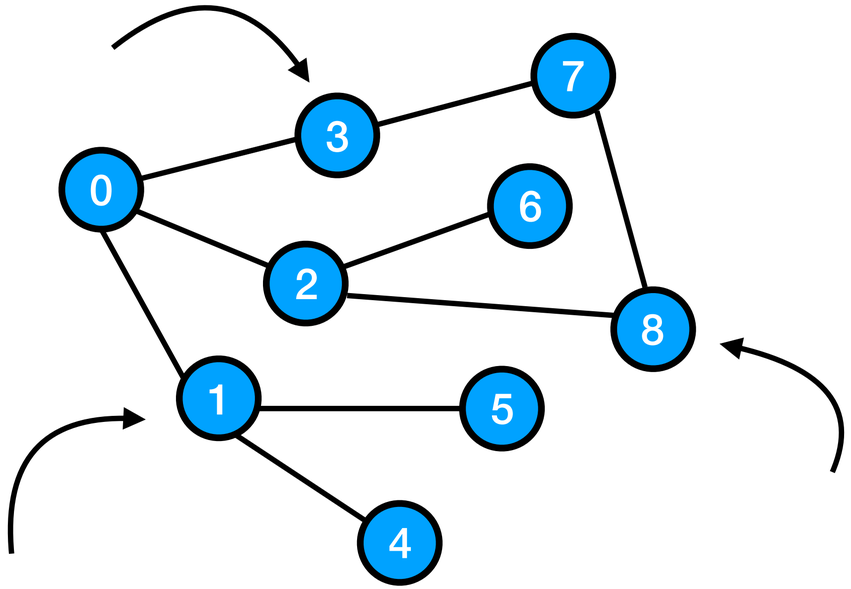
Traverse

Process of searching through a graph by visiting vertices or nodes.



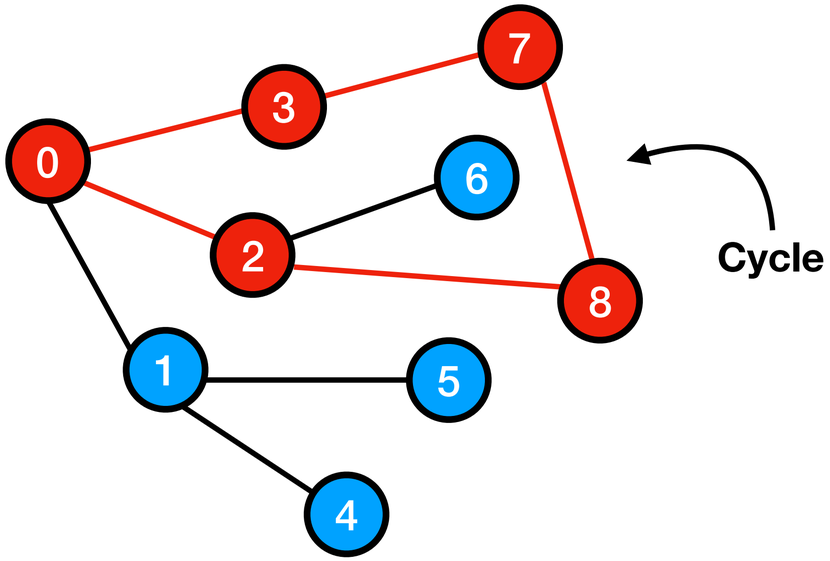
Graph Traversal

Graph traversing start with any vertex.



Graph Traversal

Need to remember which vertex has already visited.

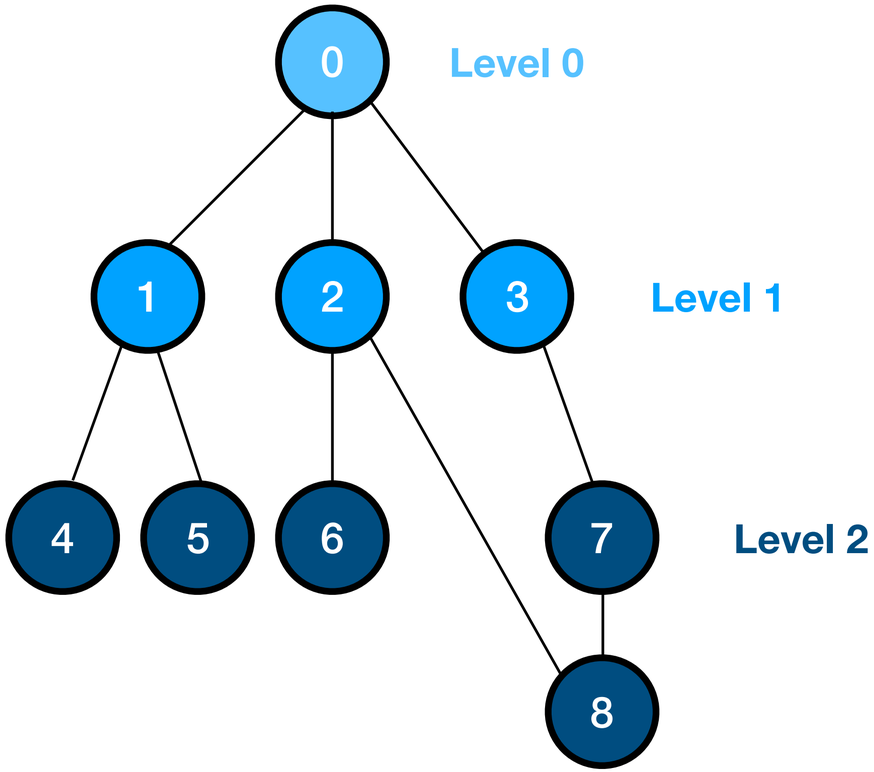


Graph Traversal Algorithms

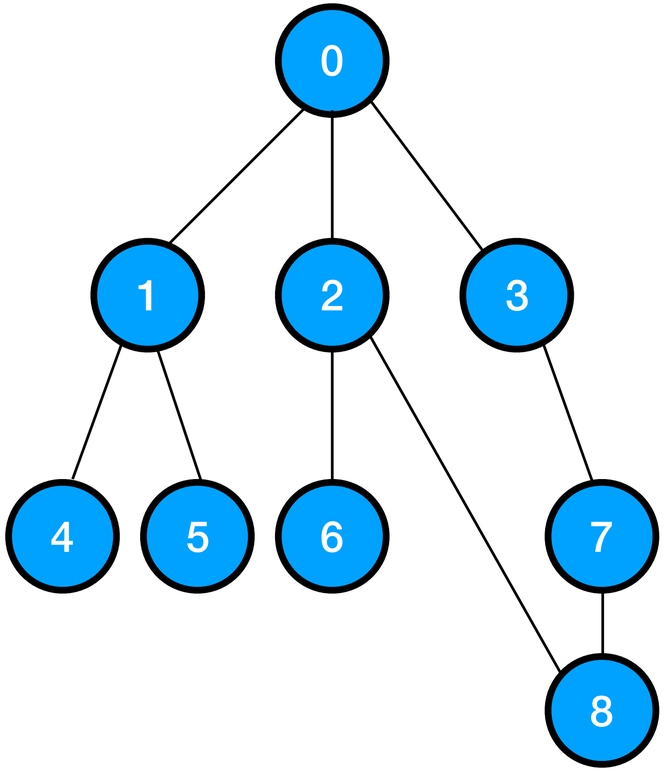
- Breath First Search (BFS)
- Depth First Search (DFS)

Breath First Search (BFS)

- Traverse into sibling/neighboring before children.
- Implement using queue.

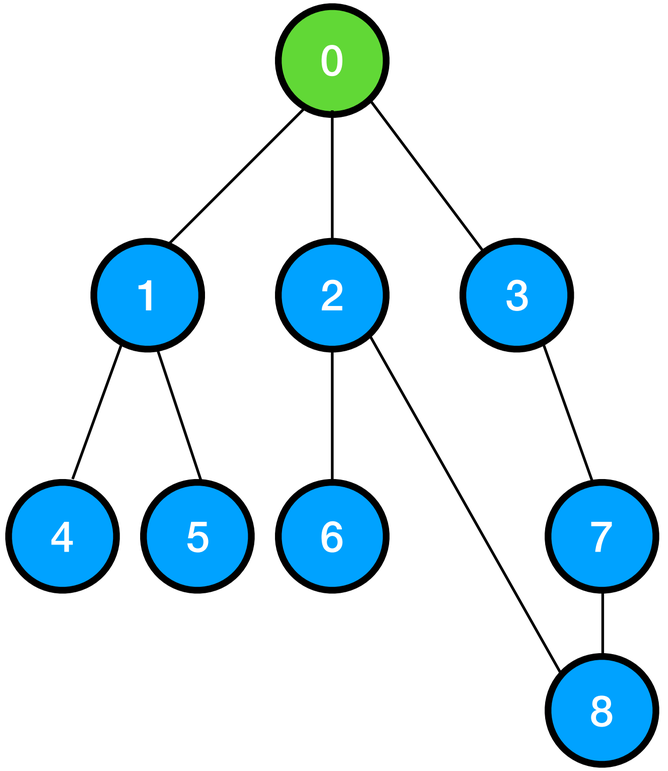


Breath First Search (BFS)



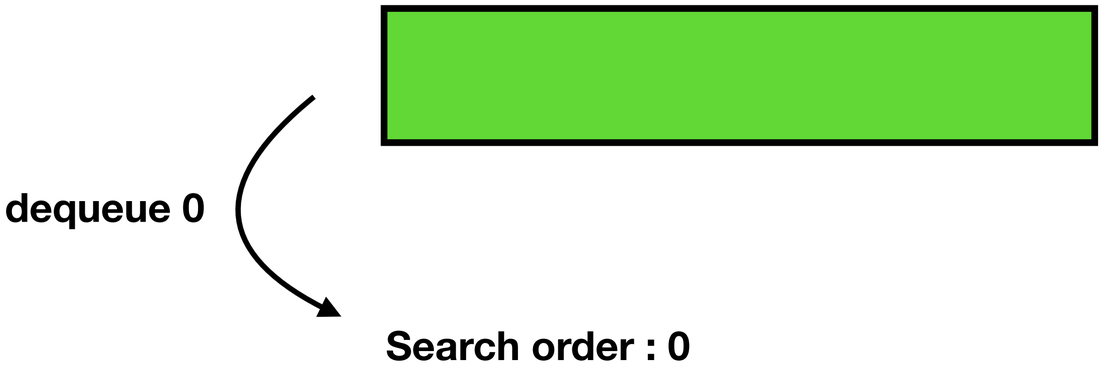
Search order :

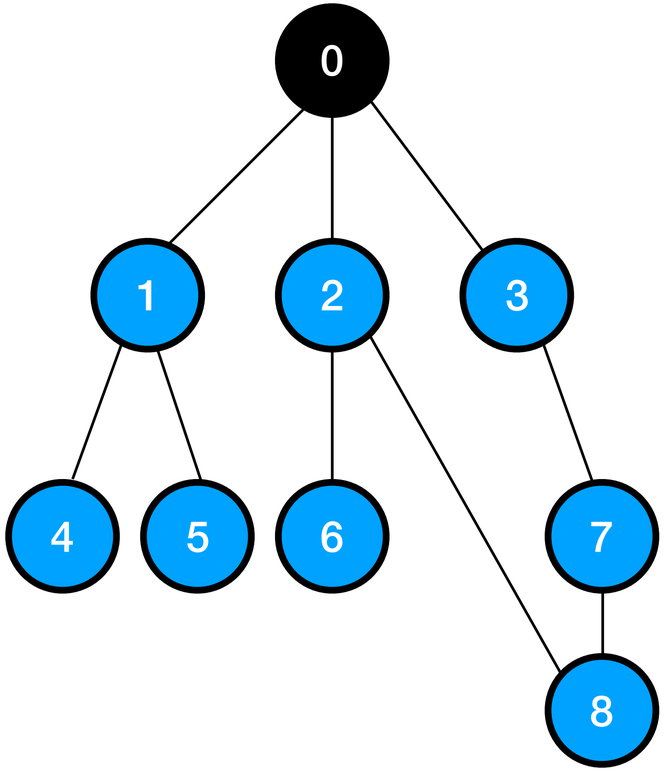
Breath First Search (BFS)



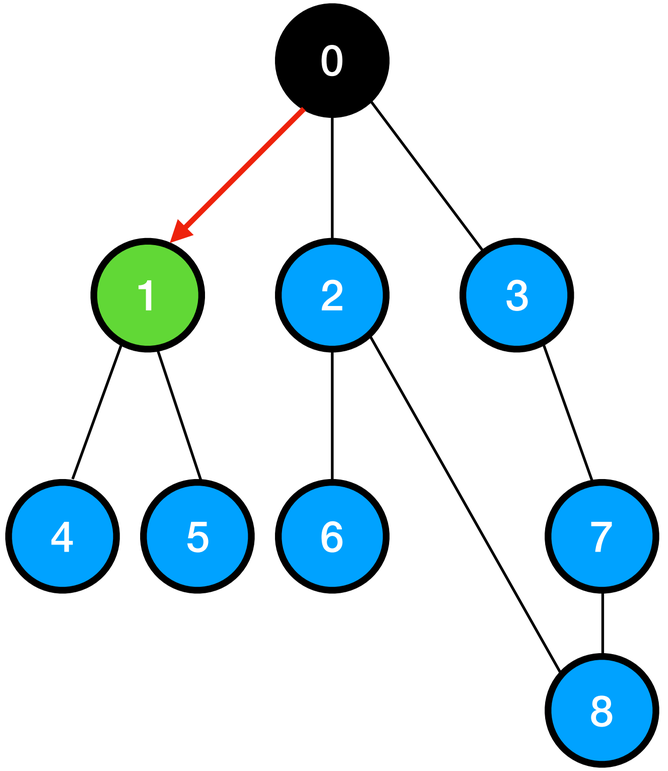
Search order :

Breath First Search (BFS)





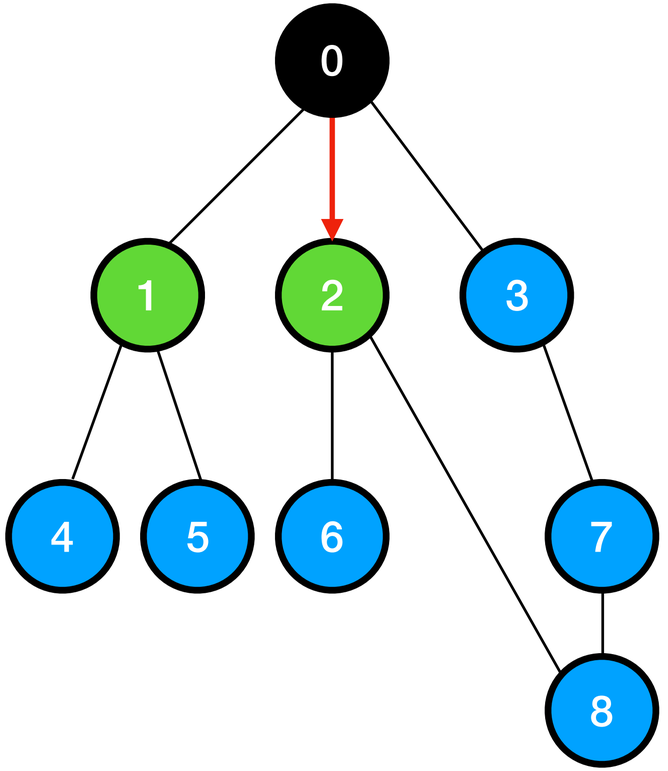
Breath First Search (BFS)



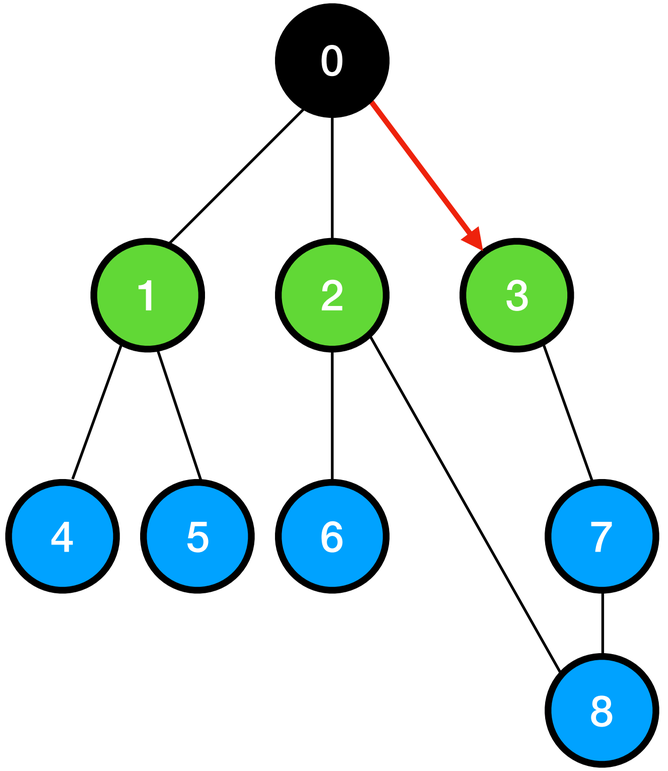
Search order : 0

Breath First Search (BFS)

Search order : 0

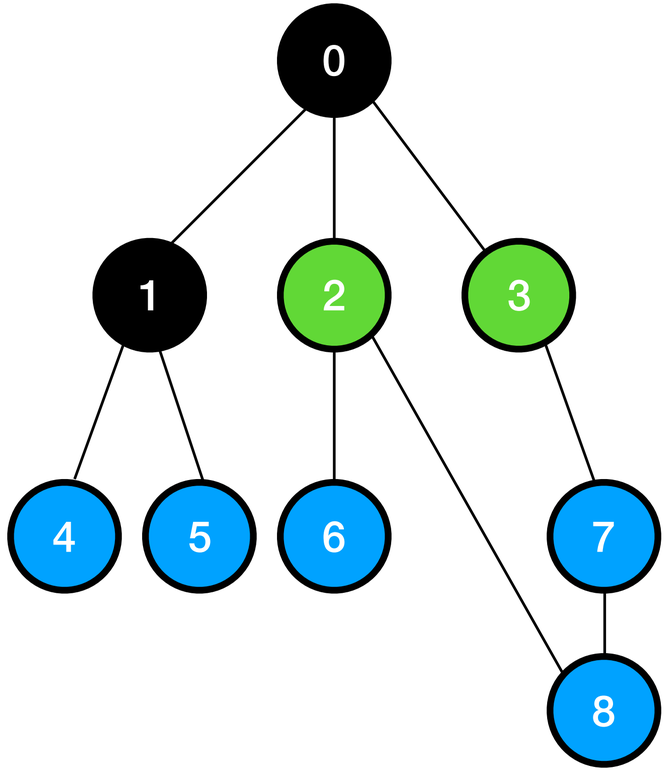


Breath First Search (BFS)

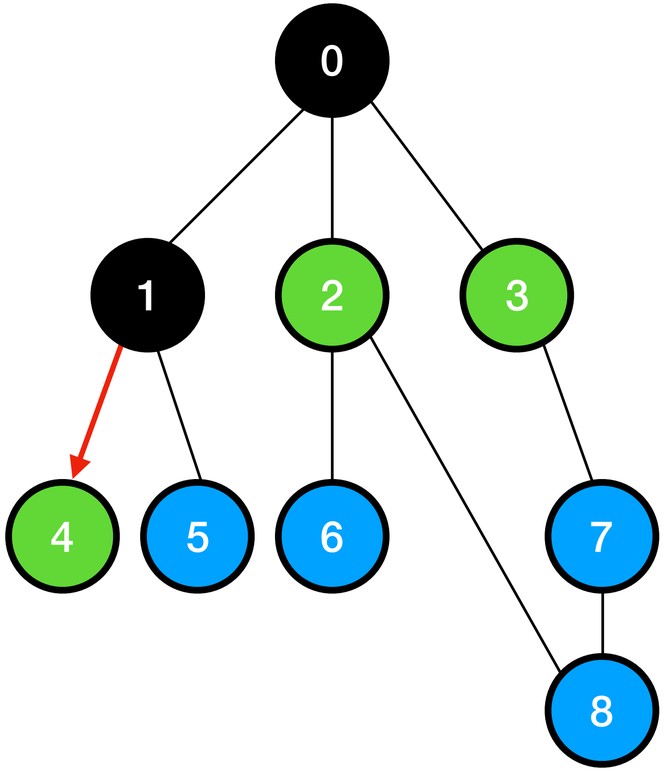


Search order : 0

Breath First Search (BFS)



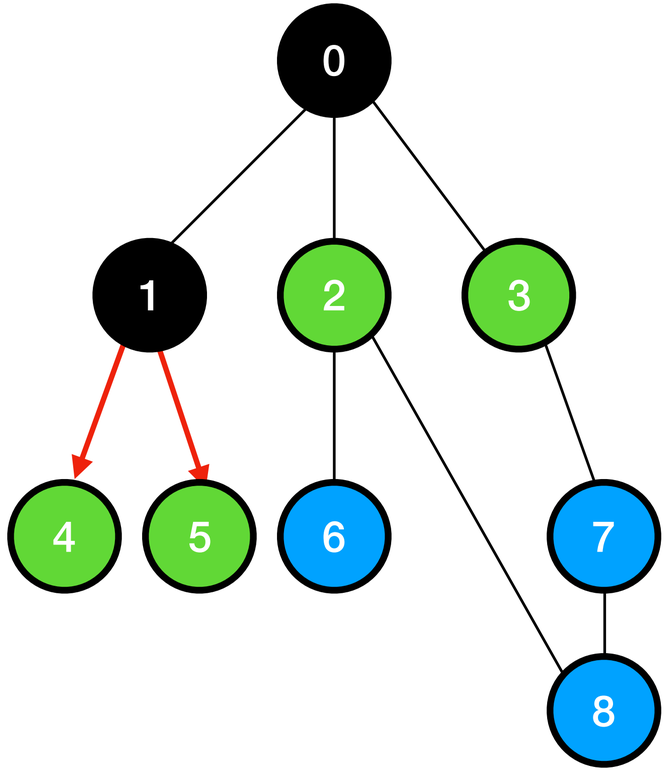
Breath First Search (BFS)



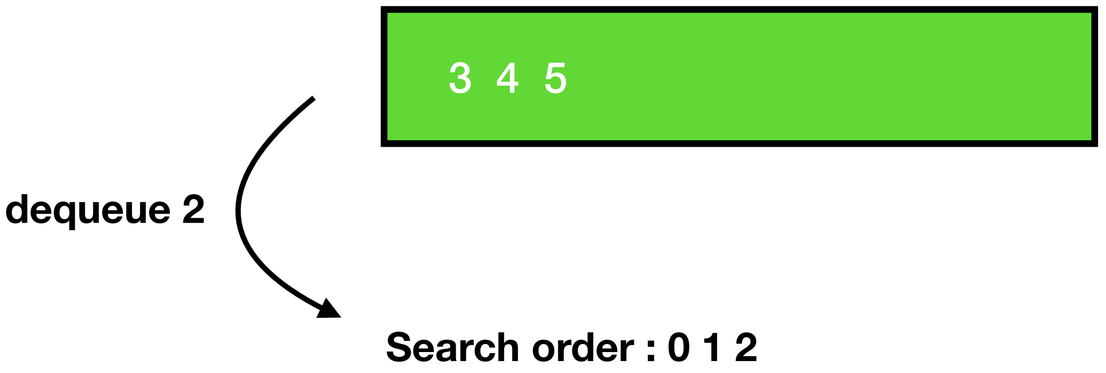
Search order : 0 1

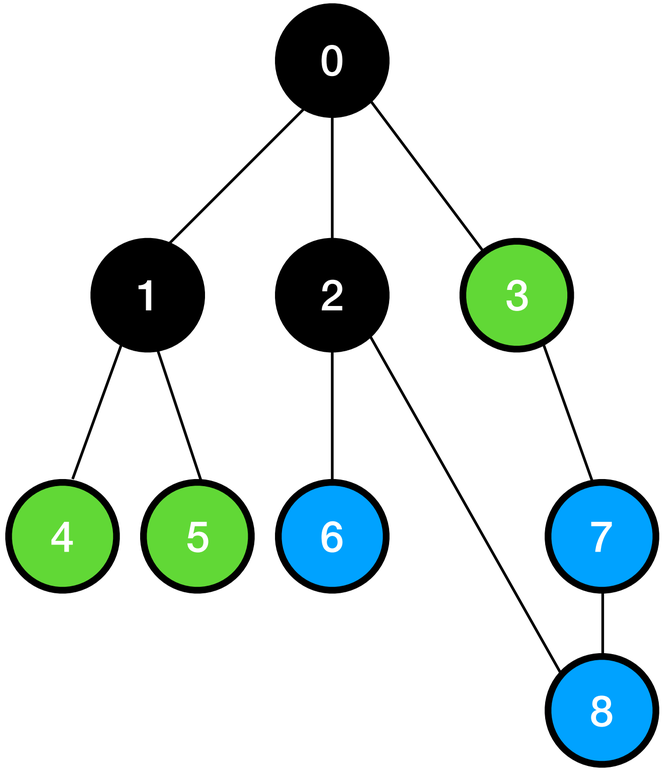
Breath First Search (BFS)

Search order : 0 1



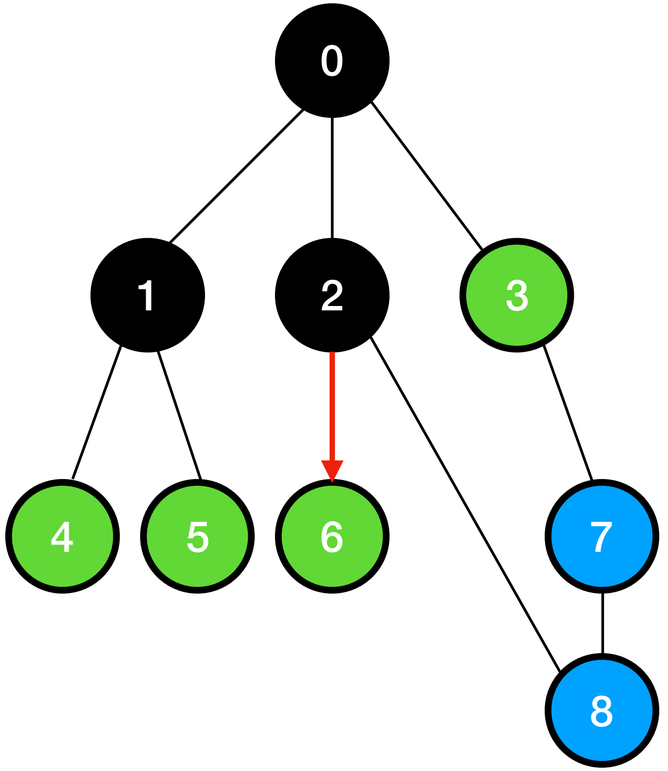
Breath First Search (BFS)





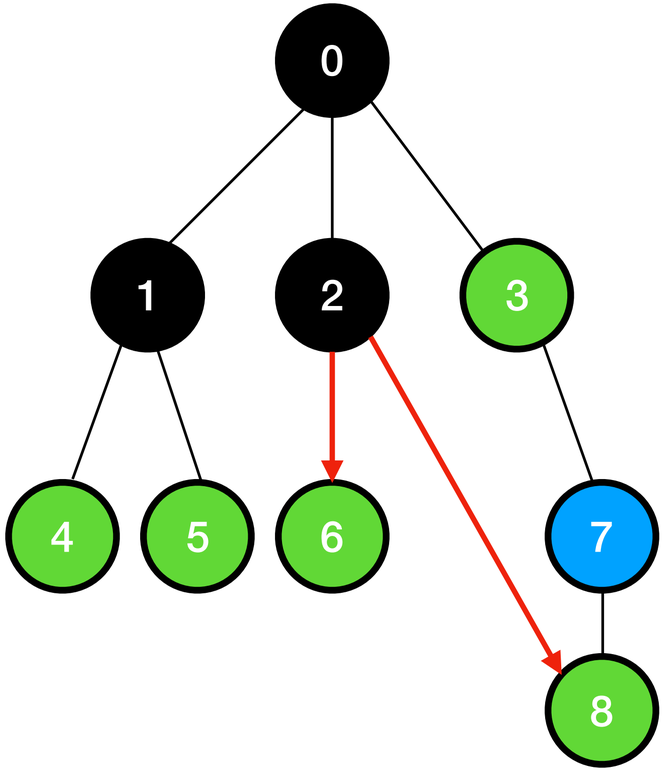
Breath First Search (BFS)

Search order : 0 1 2

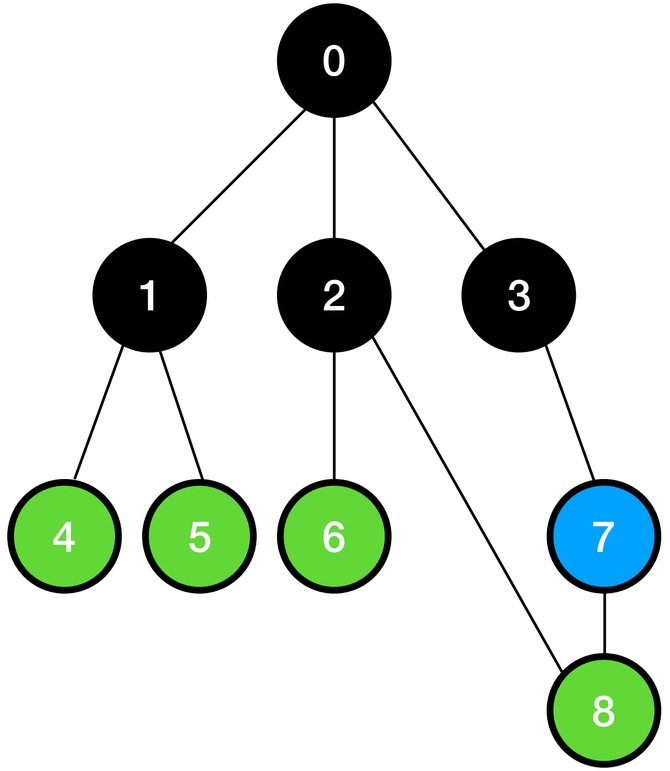


Breath First Search (BFS)

Search order : 0 1 2

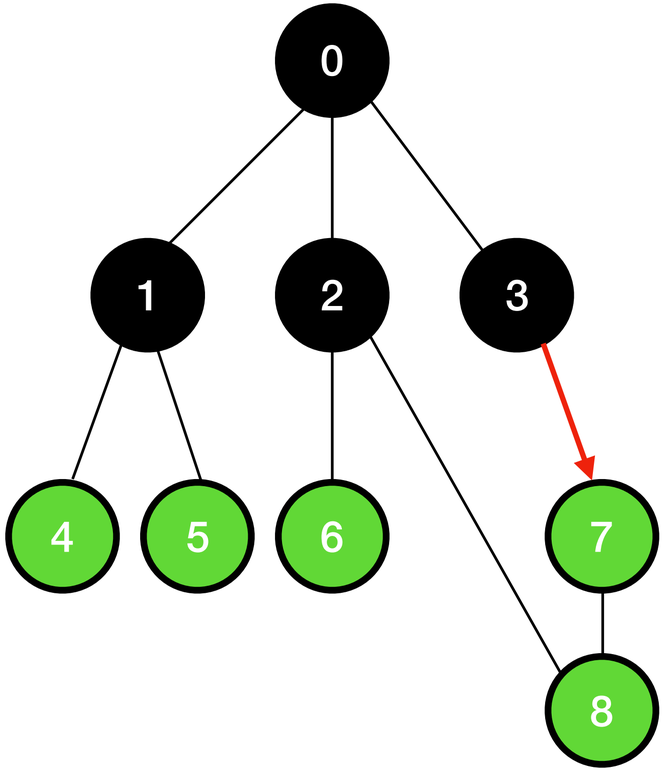


Breath First Search (BFS)

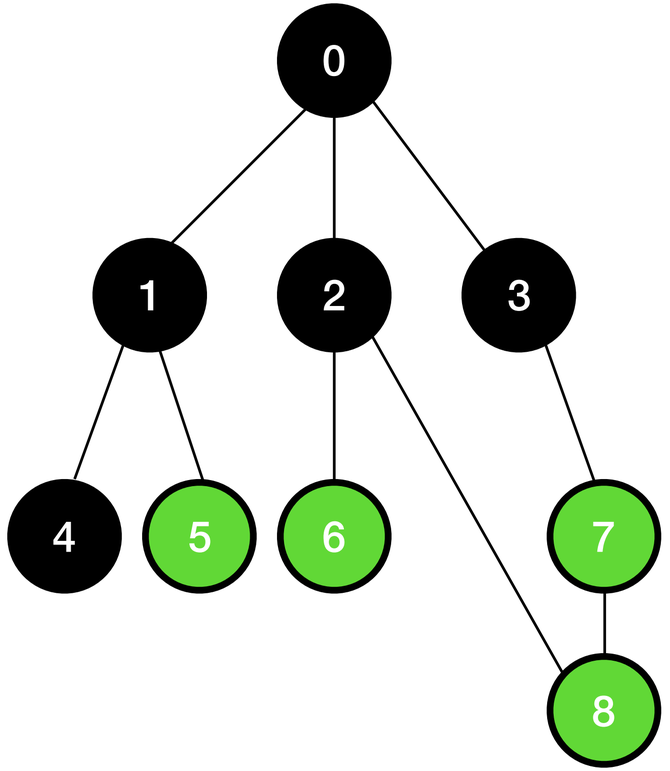


Breath First Search (BFS)

Search order : 0 1 2 3



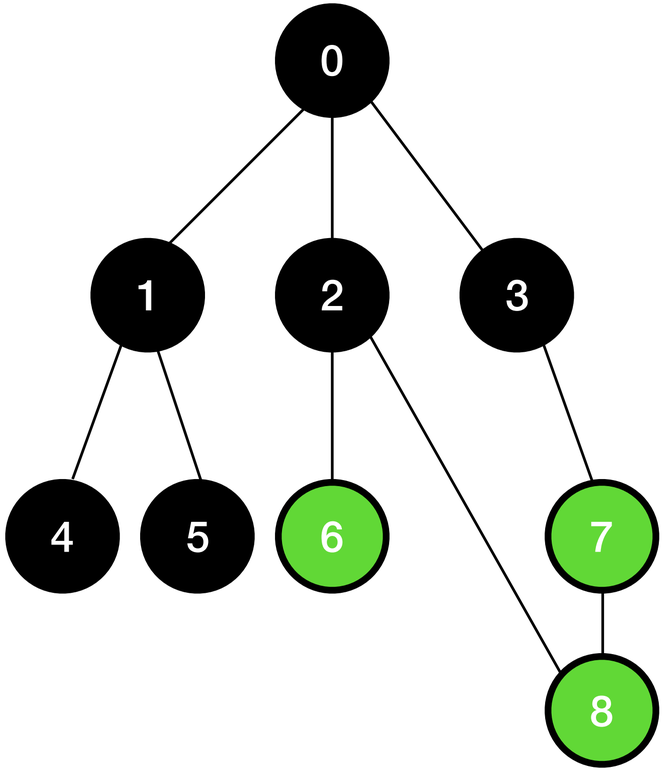
Breath First Search (BFS)



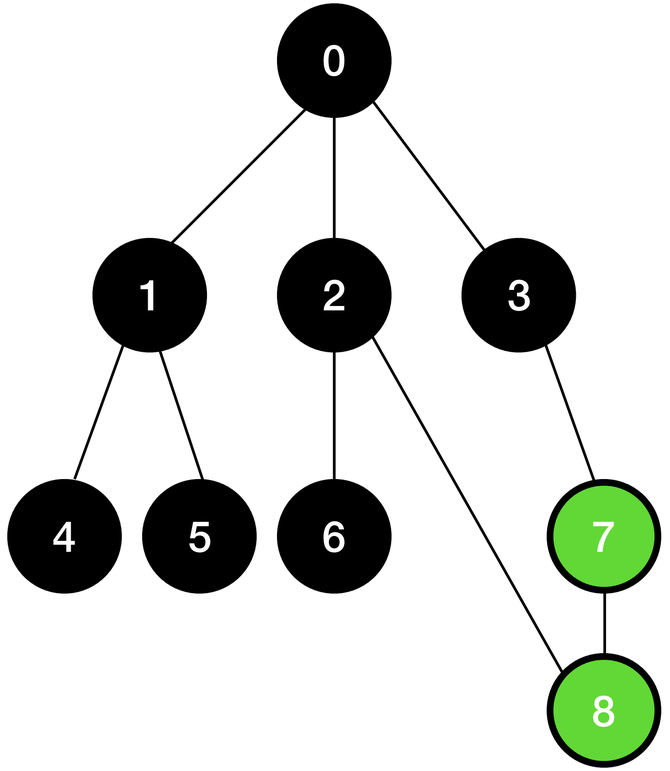
Breath First Search (BFS)

dequeue 5

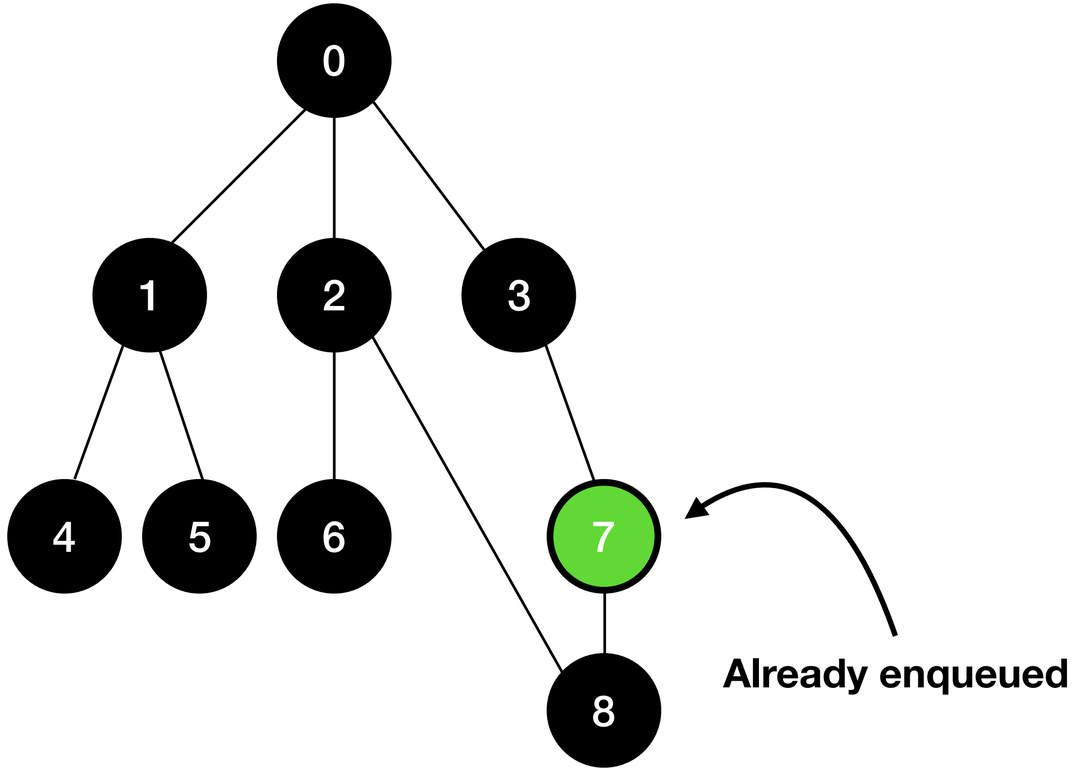
Search order : 0 1 2 3 4 5



Breath First Search (BFS)



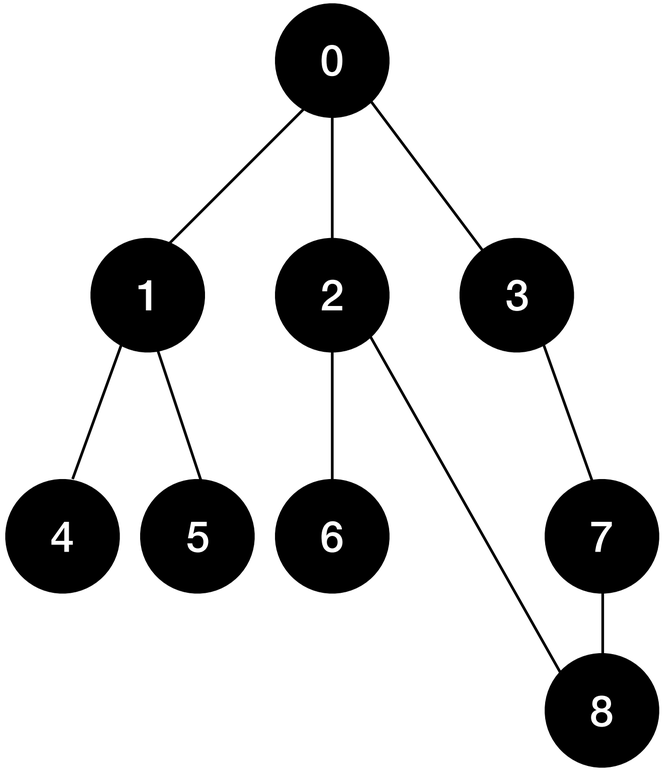
Breath First Search (BFS)



Breath First Search (BFS)

dequeue 7

Search order : 0 1 2 3 4 5 6 8 7



BFS Pseudocode

```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

BFS Pseudocode

```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

BFS Pseudocode

```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

BFS Pseudocode

```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

BFS Pseudocode

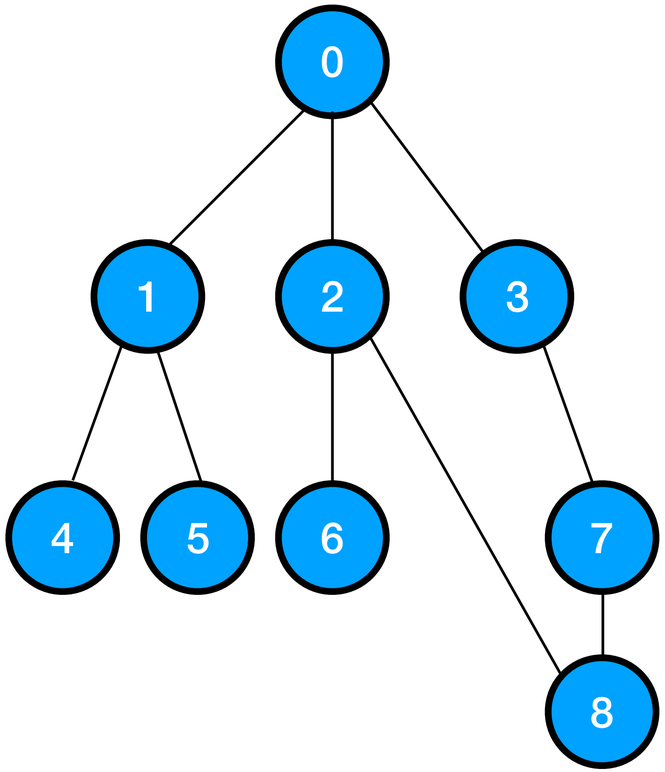
```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

BFS Pseudocode

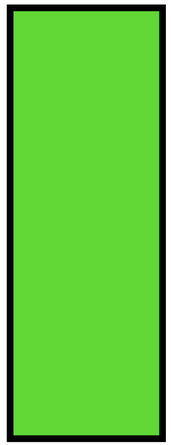
```
bfs(startVertex)
    visitedVertices = []
    queue = new Queue
    queue.enqueue(startVertex)
    visitedVertices[startVertex] = true

    while queue.isNotEmpty
        currentVertex = queue.dequeue()
        for children of currentVertex
            if visitedVertices[children] == false
                queue.enqueue(children)
                visitedVertices[children] = true
```

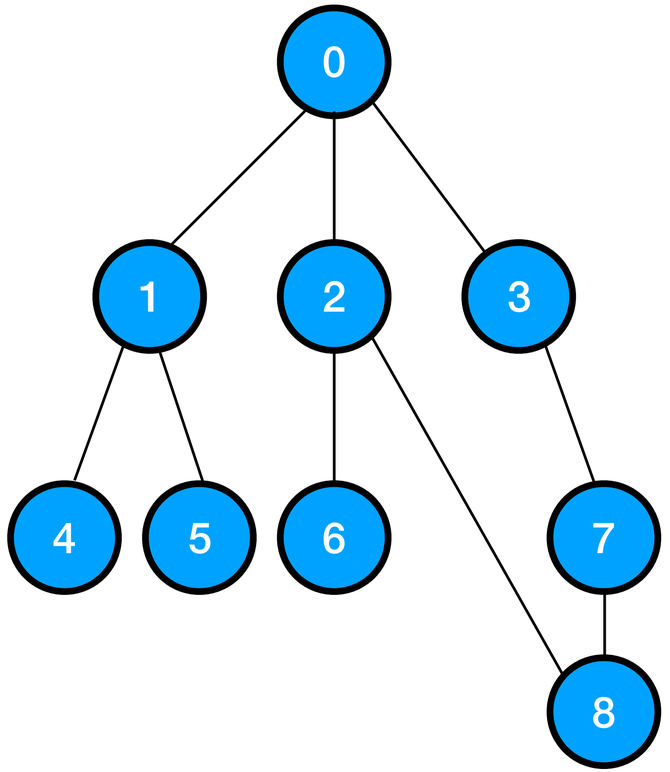
Depth First Search (DFS)

- Traverse into children before sibling/ neighboring.
- Implement using stack.



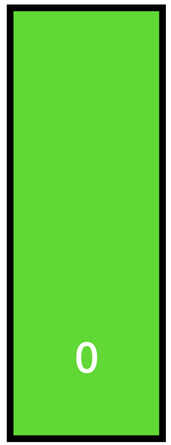
Depth First Search (DFS)

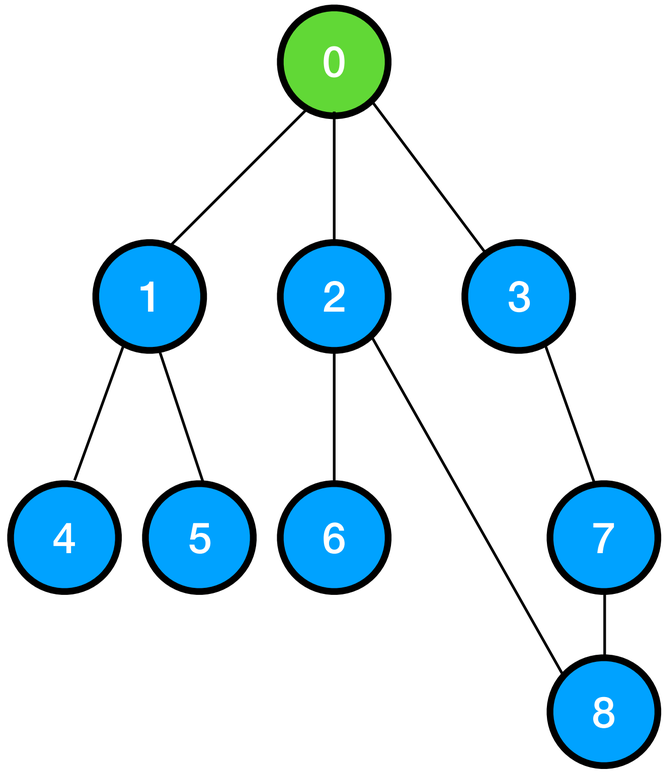




Search order :

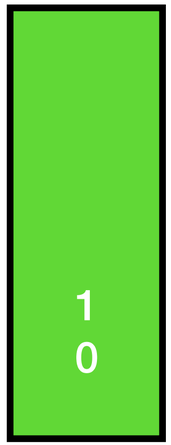
Depth First Search (DFS)

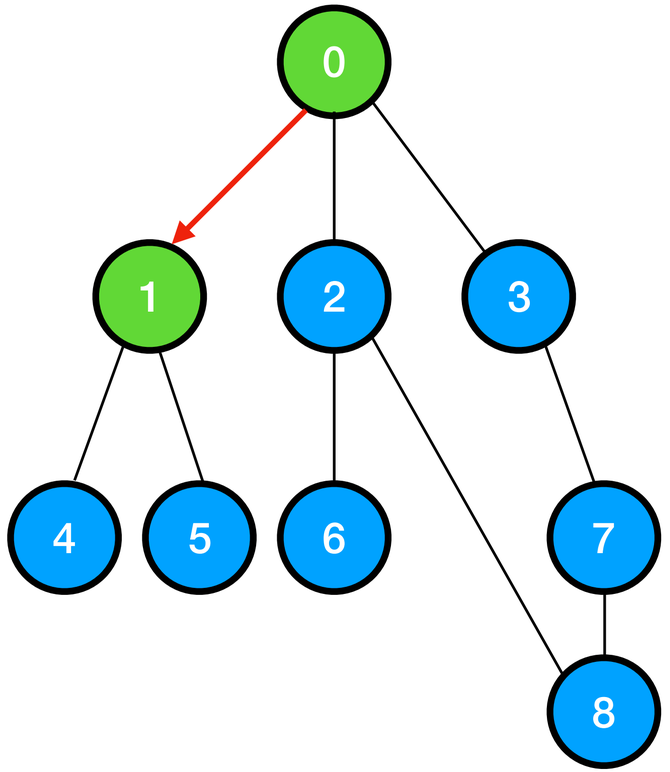




Search order : 0

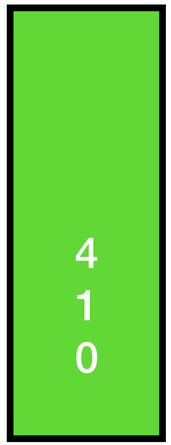
Depth First Search (DFS)

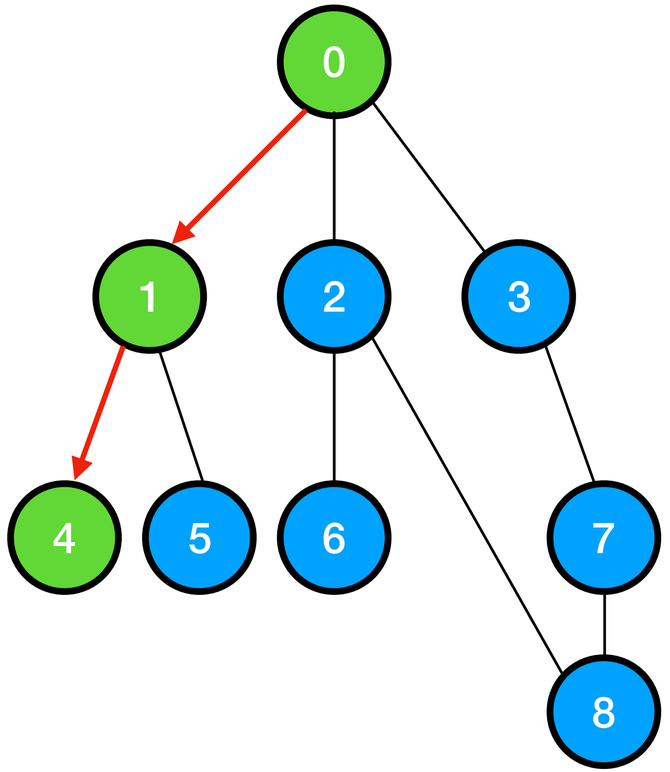




Search order : 0 1

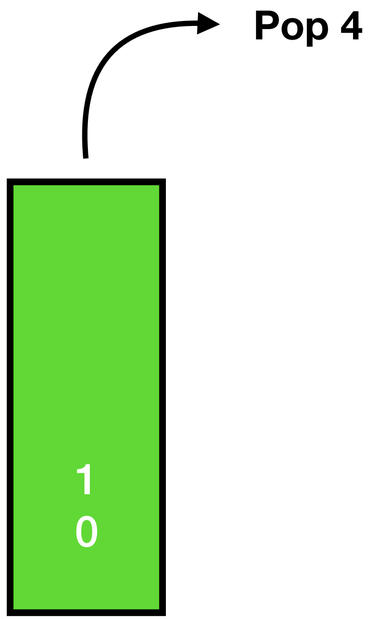
Depth First Search (DFS)

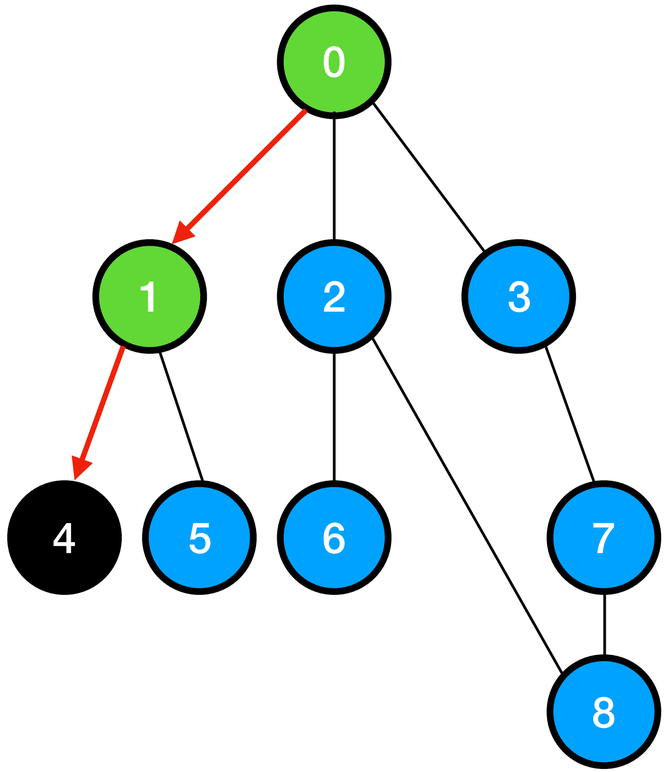




Search order : 0 1 4

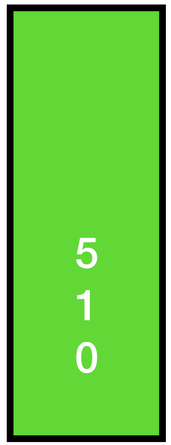
Depth First Search (DFS)

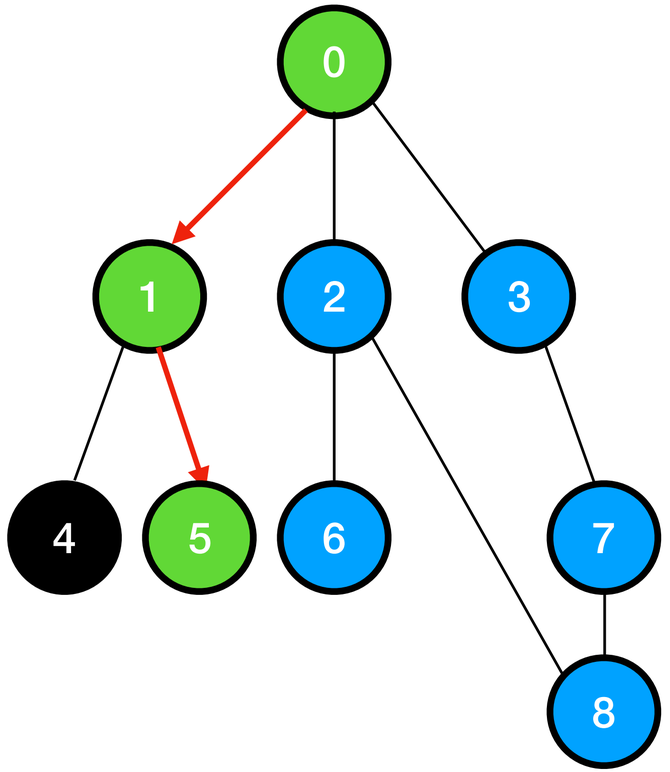




Search order : 0 1 4

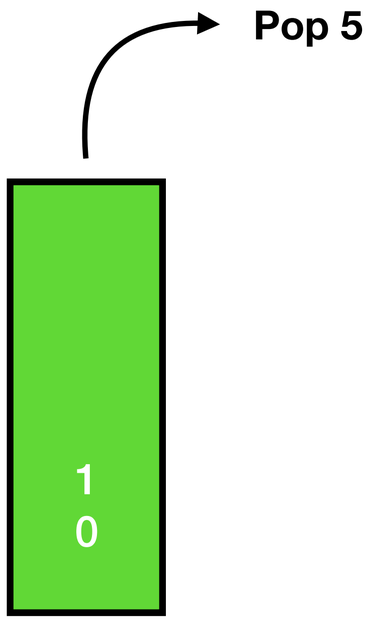
Depth First Search (DFS)

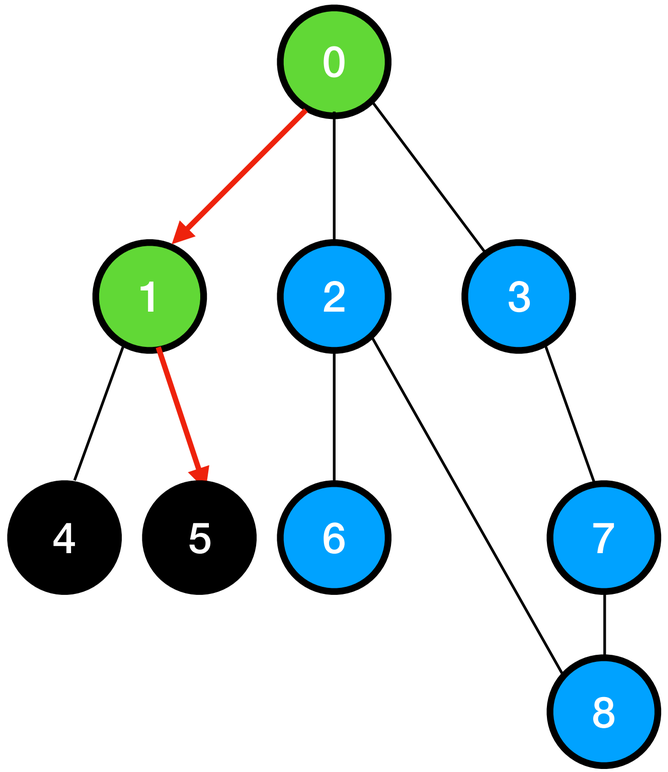




Search order : 0 1 4 5

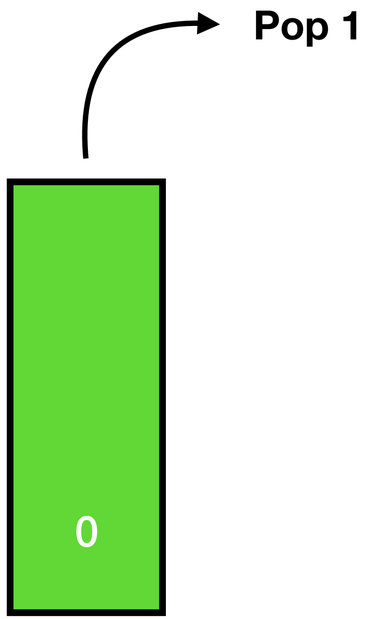
Depth First Search (DFS)

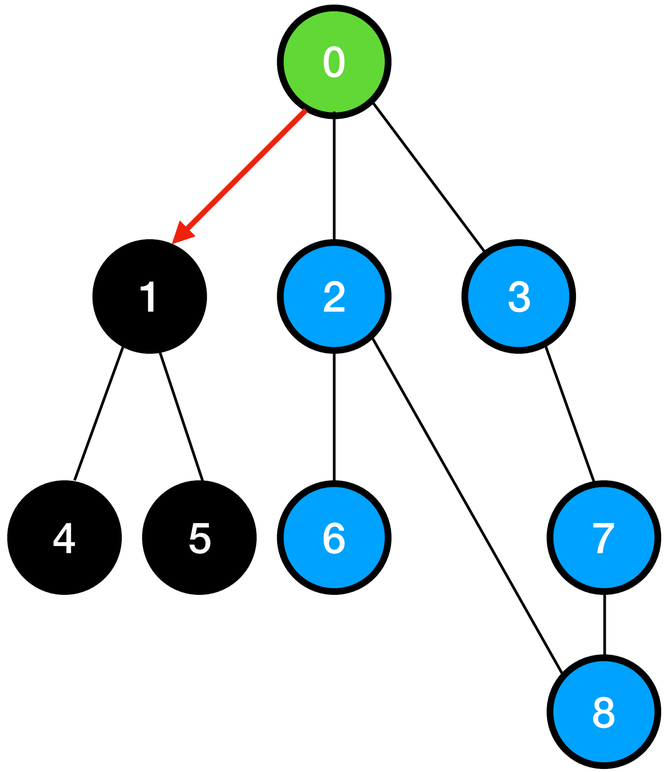




Search order : 0 1 4 5

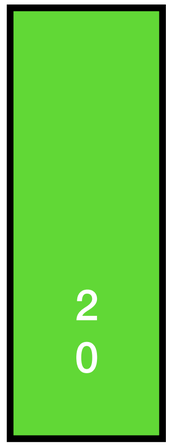
Depth First Search (DFS)

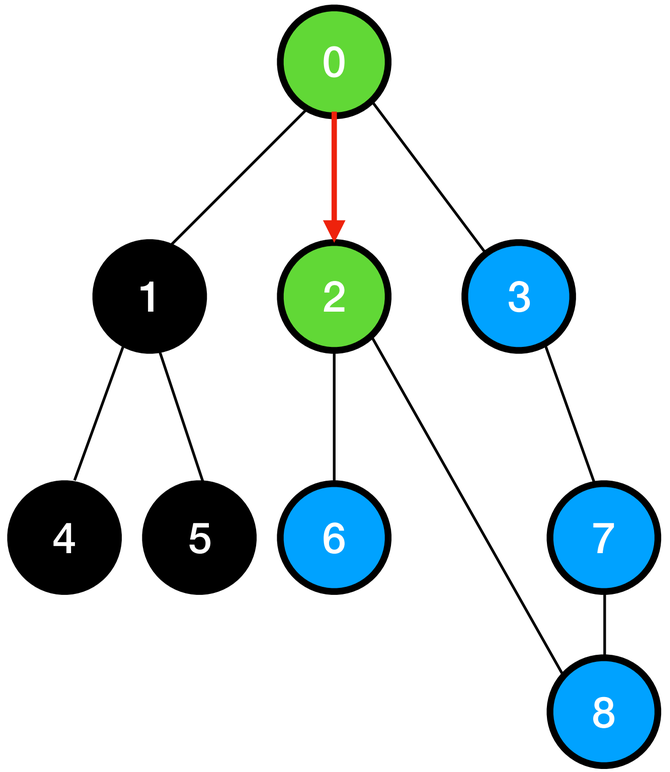




Search order : 0 1 4 5

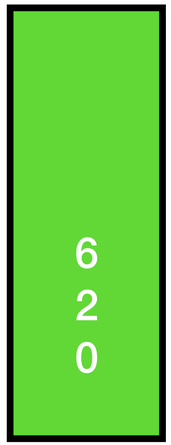
Depth First Search (DFS)

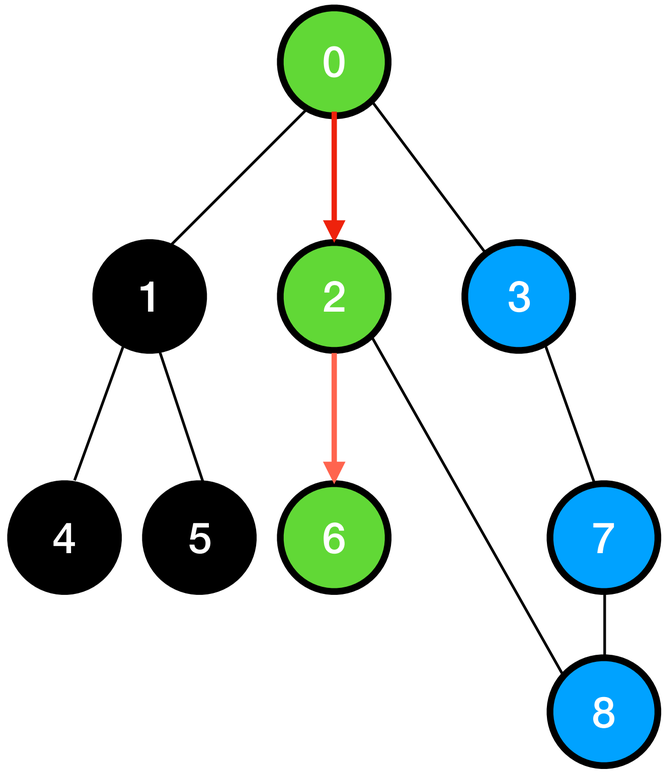




Search order : 0 1 4 5 2

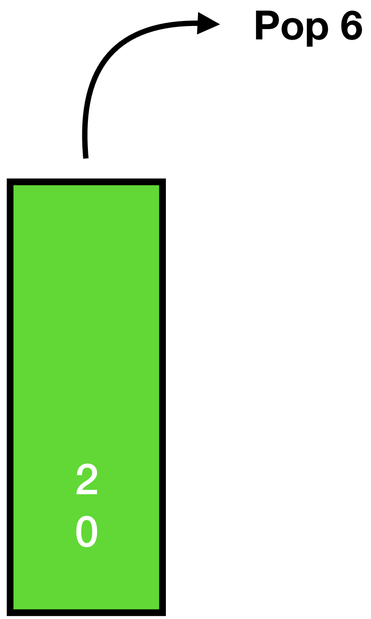
Depth First Search (DFS)

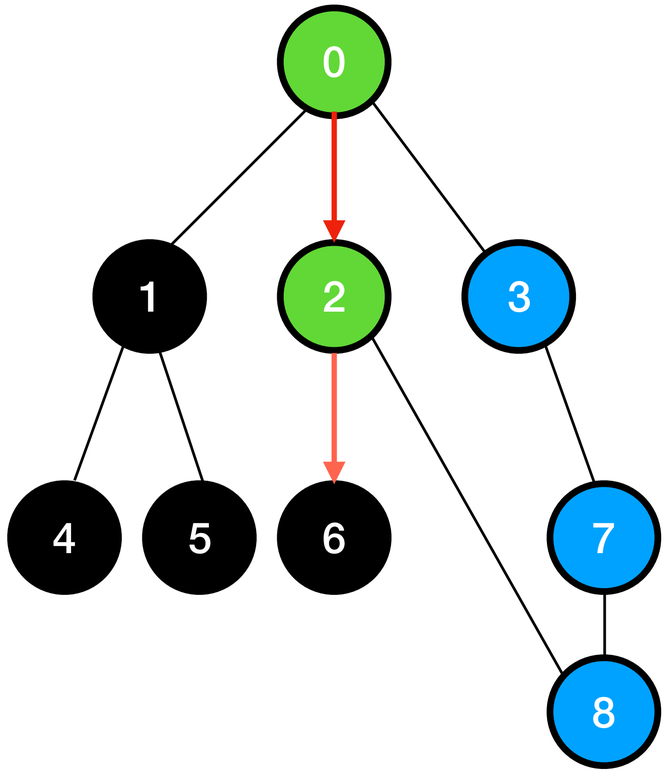




Search order : 0 1 4 5 2 6

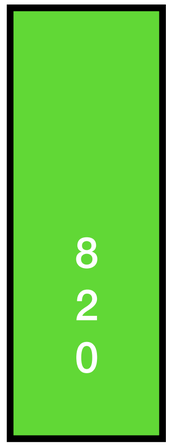
Depth First Search (DFS)



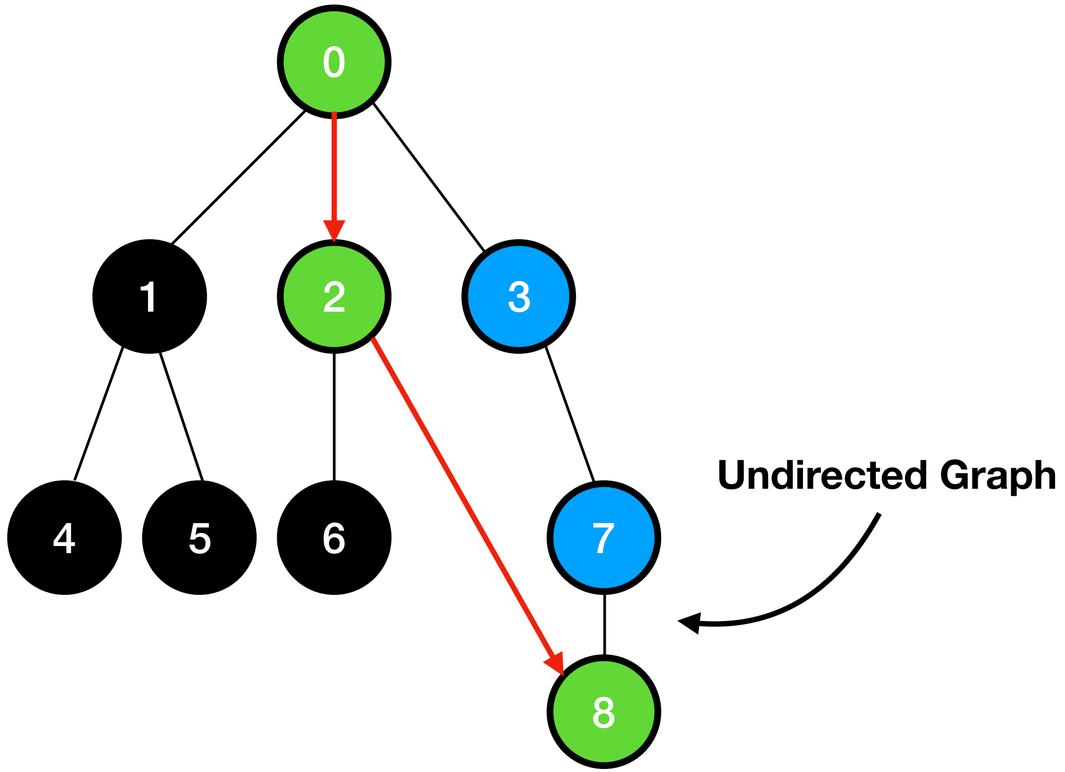


Search order : 0 1 4 5 2 6

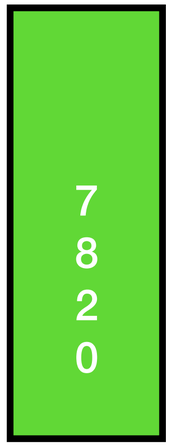
Depth First Search (DFS)

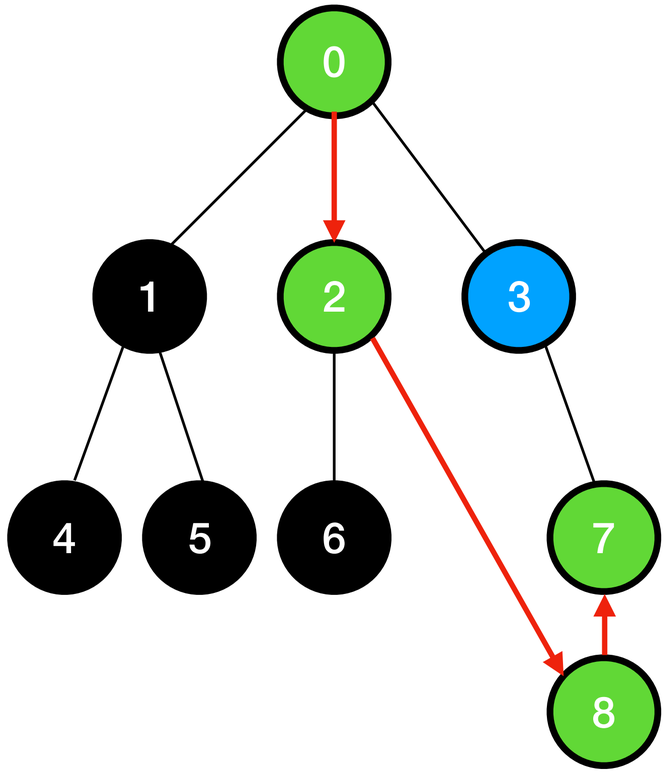


Search order : 0 1 4 5 2 6 8



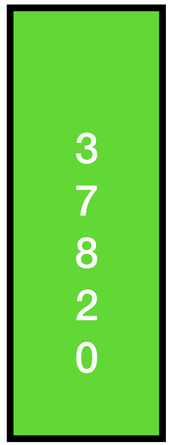
Depth First Search (DFS)

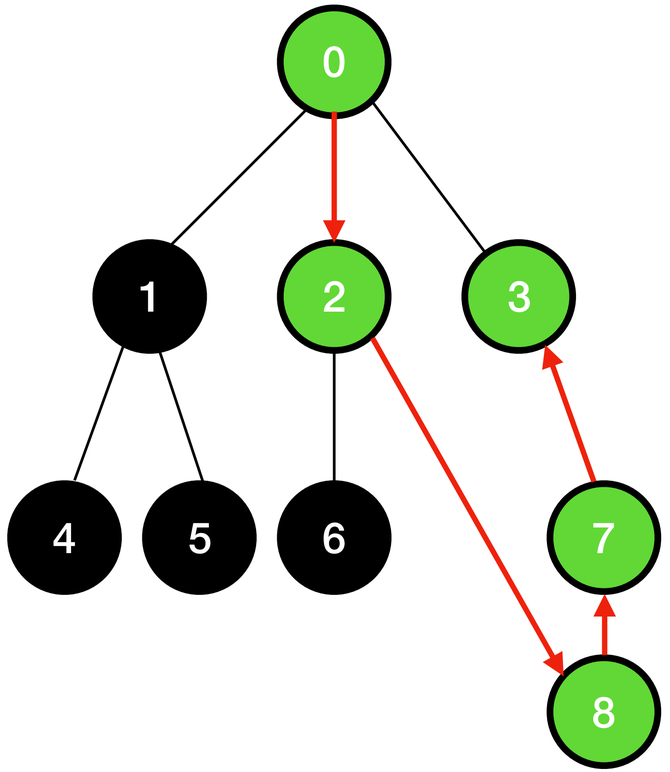




Search order : 0 1 4 5 2 6 8 7

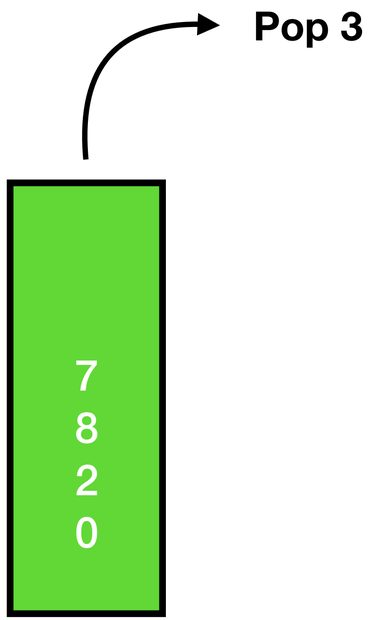
Depth First Search (DFS)

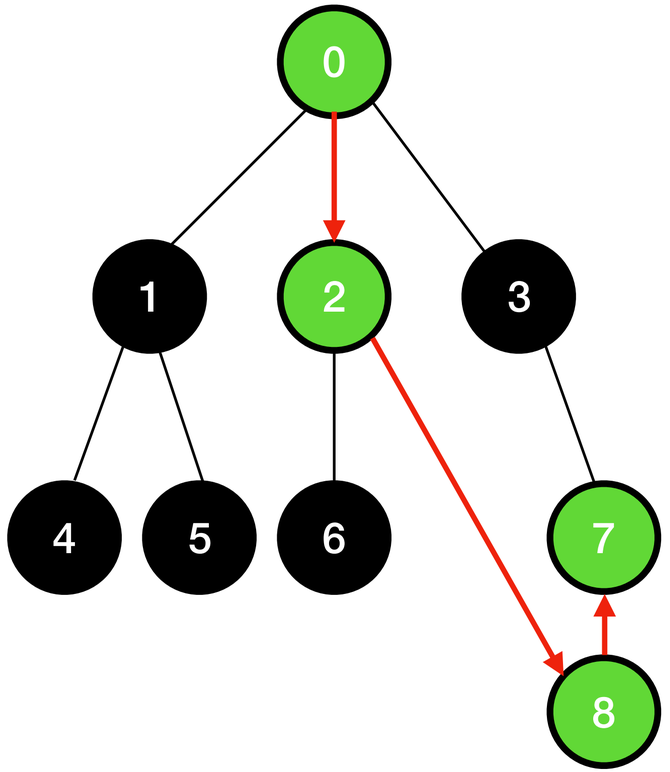




Search order : 0 1 4 5 2 6 8 7 3

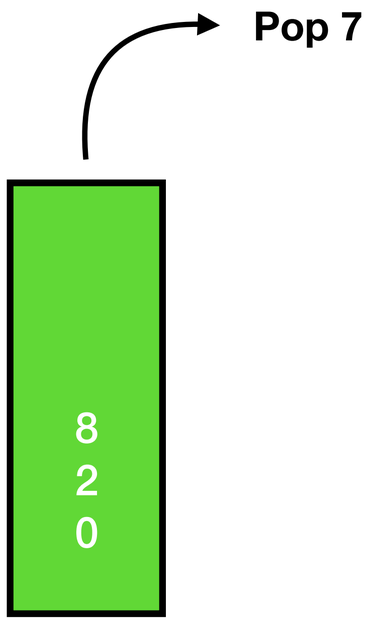
Depth First Search (DFS)

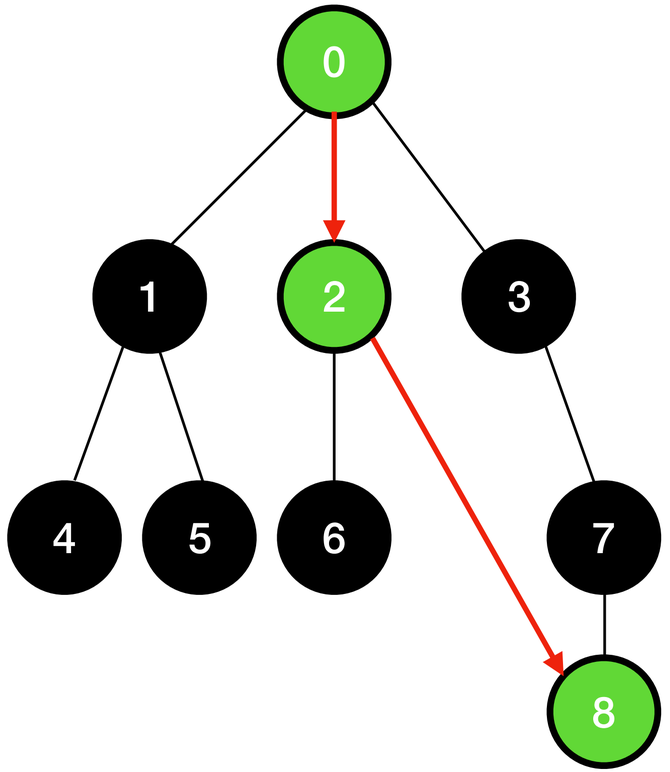




Search order : 0 1 4 5 2 6 8 7 3

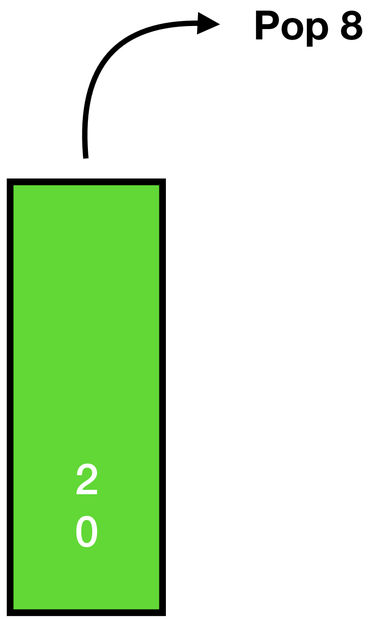
Depth First Search (DFS)

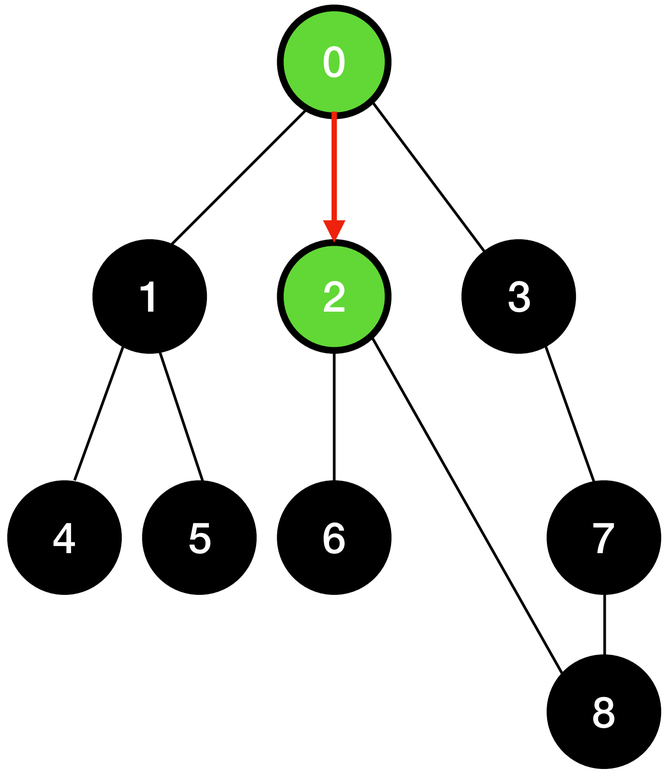




Search order : 0 1 4 5 2 6 8 7 3

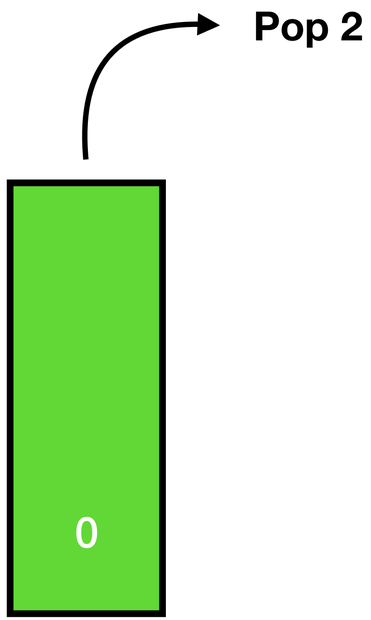
Depth First Search (DFS)

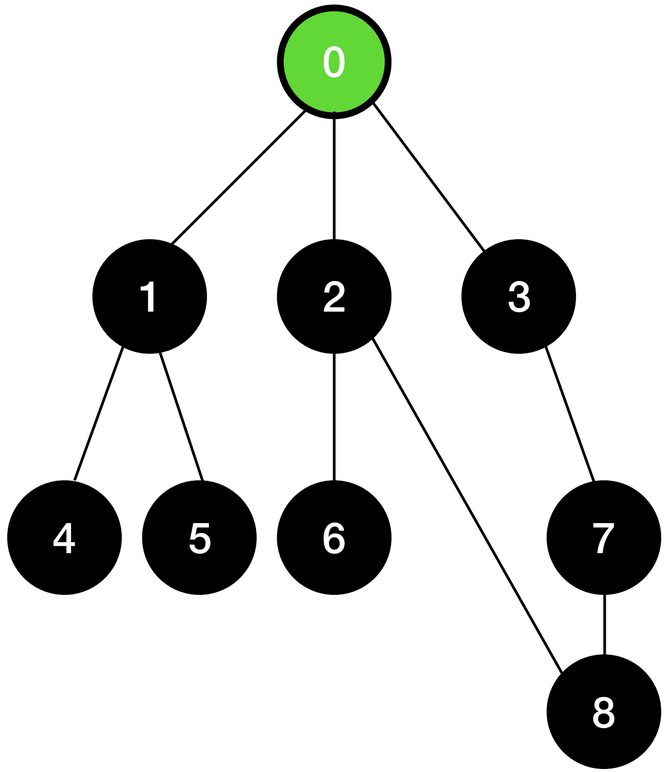




Search order : 0 1 4 5 2 6 8 7 3

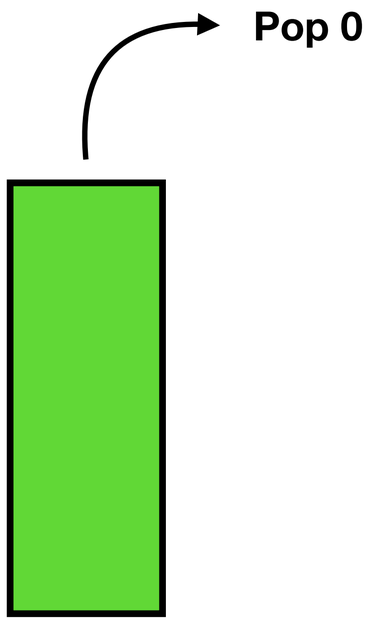
Depth First Search (DFS)

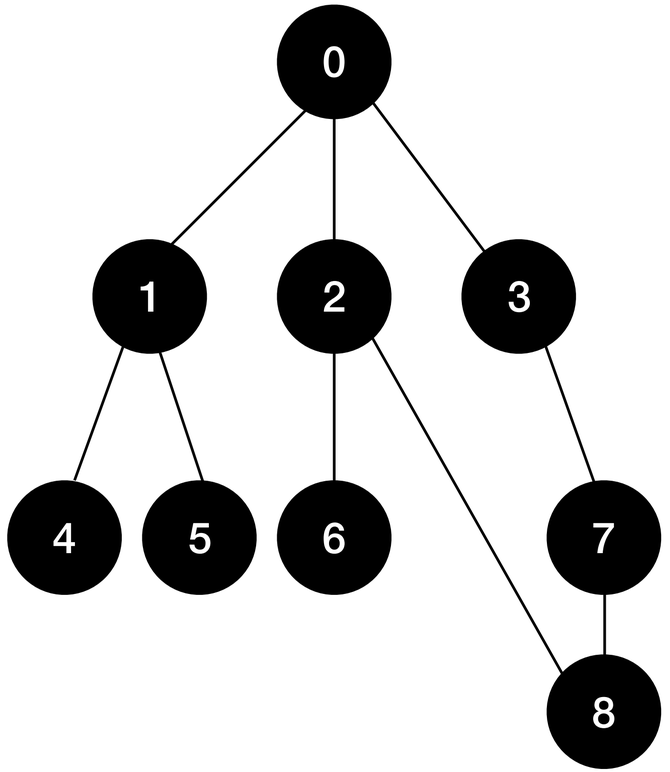




Search order : 0 1 4 5 2 6 8 7 3

Depth First Search (DFS)





Search order : 0 1 4 5 2 6 8 7 3

DFS Pseudocode

```
dfs(startVertex)
    visitedVertices = []
    dfsRecursive(startVertex, visitedVertices)


dfsRecursive(currentVertex, visitedVertices)
    visitedVertices[currentVertex] = true

    for children of currentVertex
        if visitedVertices[children] == false
            dfsRecursive(children, visitedVertices)
```

DFS Pseudocode

```
dfs(startVertex)
    visitedVertices = []
    dfsRecursive(startVertex, visitedVertices)


dfsRecursive(currentVertex, visitedVertices)
    visitedVertices[currentVertex] = true

    for children of currentVertex
        if visitedVertices[children] == false
            dfsRecursive(children, visitedVertices)
```

DFS Pseudocode

```
dfs(startVertex)
    visitedVertices = []
    dfsRecursive(startVertex, visitedVertices)


dfsRecursive(currentVertex, visitedVertices)
    visitedVertices[currentVertex] = true

    for children of currentVertex
        if visitedVertices[children] == false
            dfsRecursive(children, visitedVertices)
```

DFS Pseudocode

```
dfs(startVertex)
    visitedVertices = []
    dfsRecursive(startVertex, visitedVertices)


dfsRecursive(currentVertex, visitedVertices)
    visitedVertices[currentVertex] = true

    for children of currentVertex
        if visitedVertices[children] == false
            dfsRecursive(children, visitedVertices)
```

DFS Pseudocode

```
dfs(startVertex)
    visitedVertices = []
    dfsRecursive(startVertex, visitedVertices)


dfsRecursive(currentVertex, visitedVertices)
    currentVertex.visited = true

    for children of currentVertex
        if visitedVertices[children] == false
            dfsRecursive(children, visitedVertices)
```