

COL106: Data Structures and Algorithms

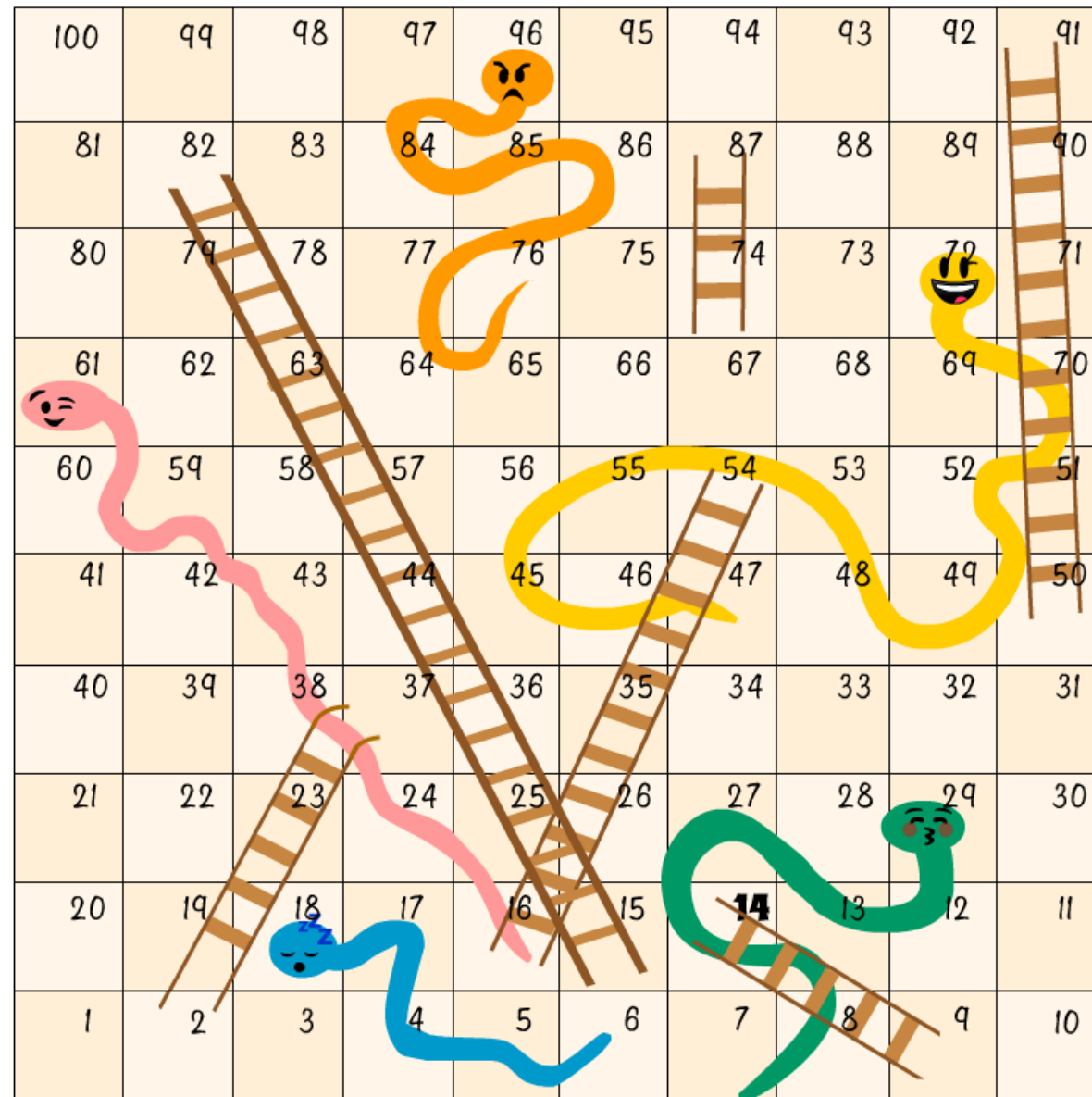
Lecture 31

Keerti Choudhary
Department of Computer Science, IIT Delhi

Today's Lecture

1. Hint for Assignment 5
2. DFS Traversal
3. DFS Tree

Hint for Assignment 5



Hint 1:

You need to model the board as a **directed graph**, and use BFS (or its variants).

Hint 2:

Claim: Let s, t be two vertices in a graph, then addition of a **new** edge (x, y) to graph decreases s to t distance if

$$\text{dist}(s, x) + 1 + \text{dist}(y, t) < \text{dist}(s, t).$$

Graph Traversal

Process of visiting vertices of a graph G.

Standard Types:

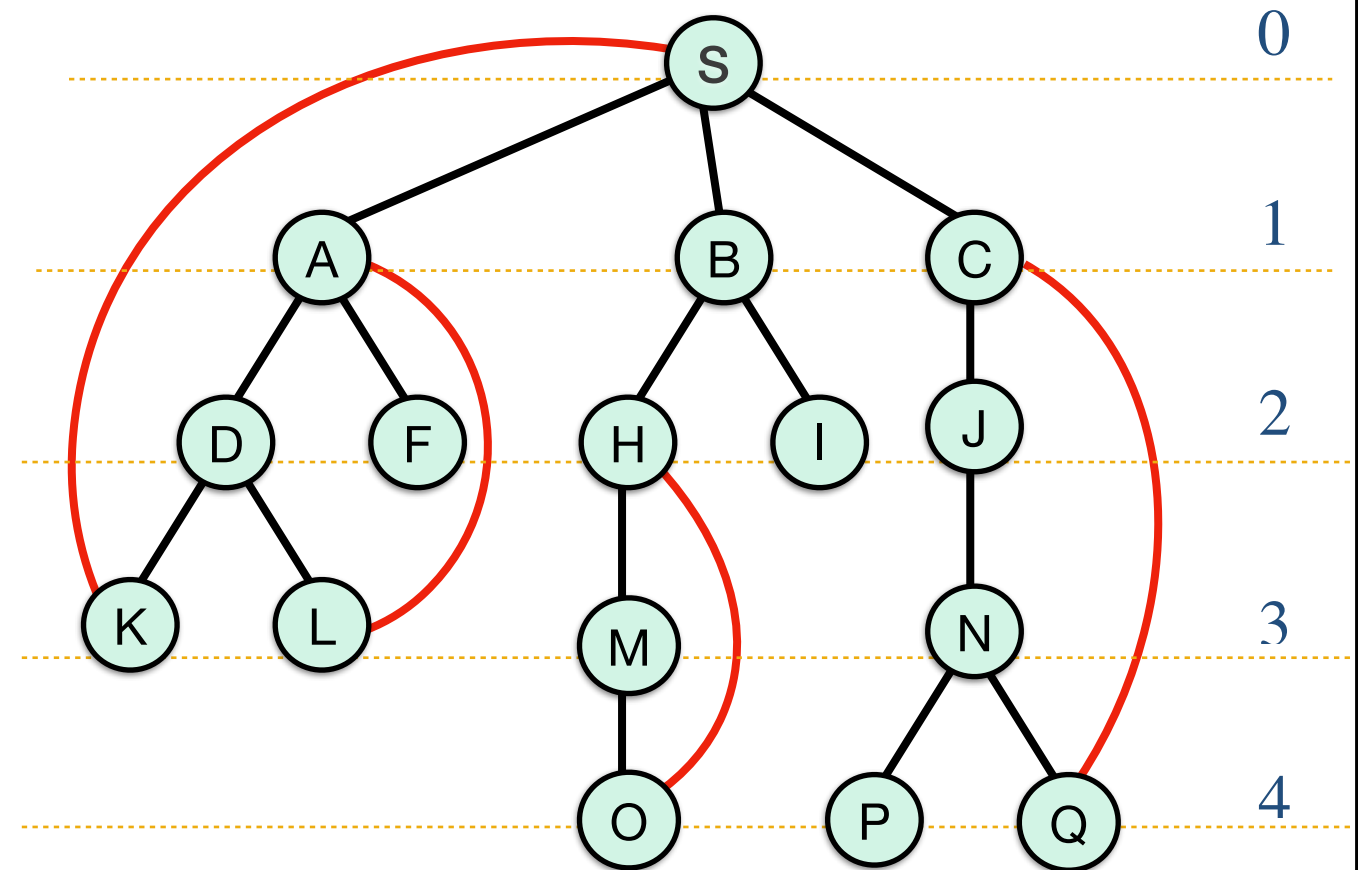
1. Breadth First Search(BFS) : visits vertices layer by layer
2. Depth First Search(DFS) : _____?

Lecture 30

Characterisation of DFS tree:

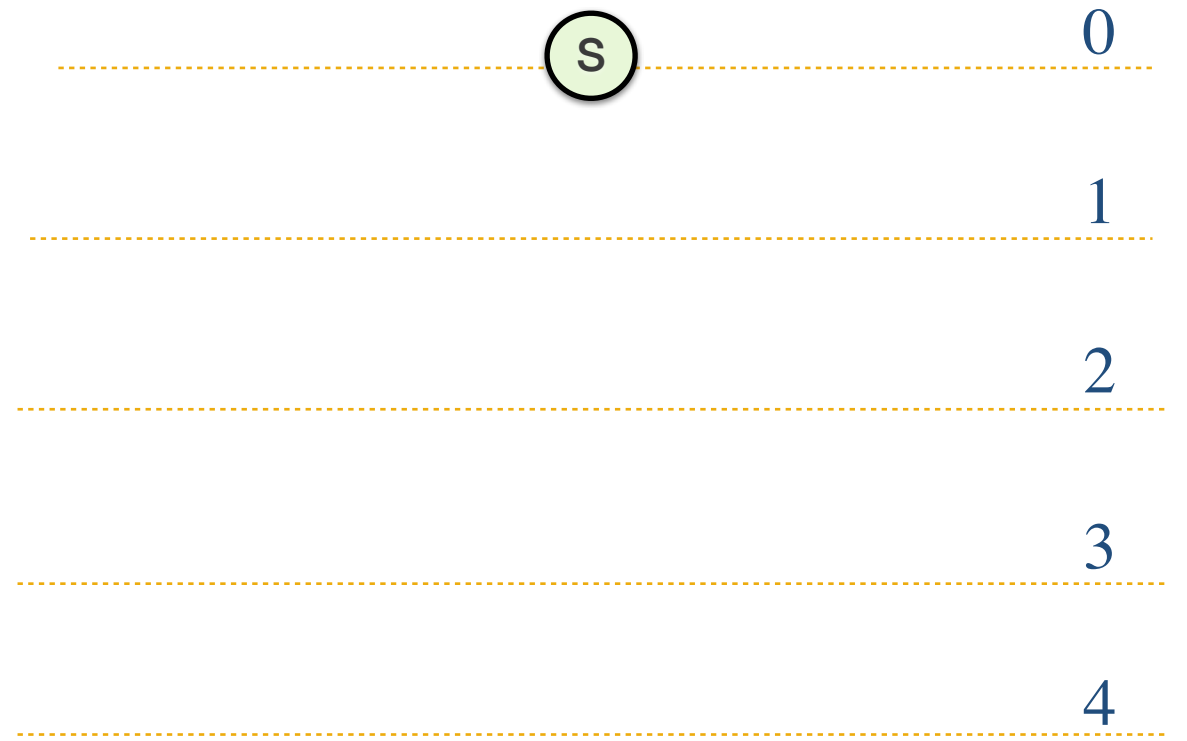
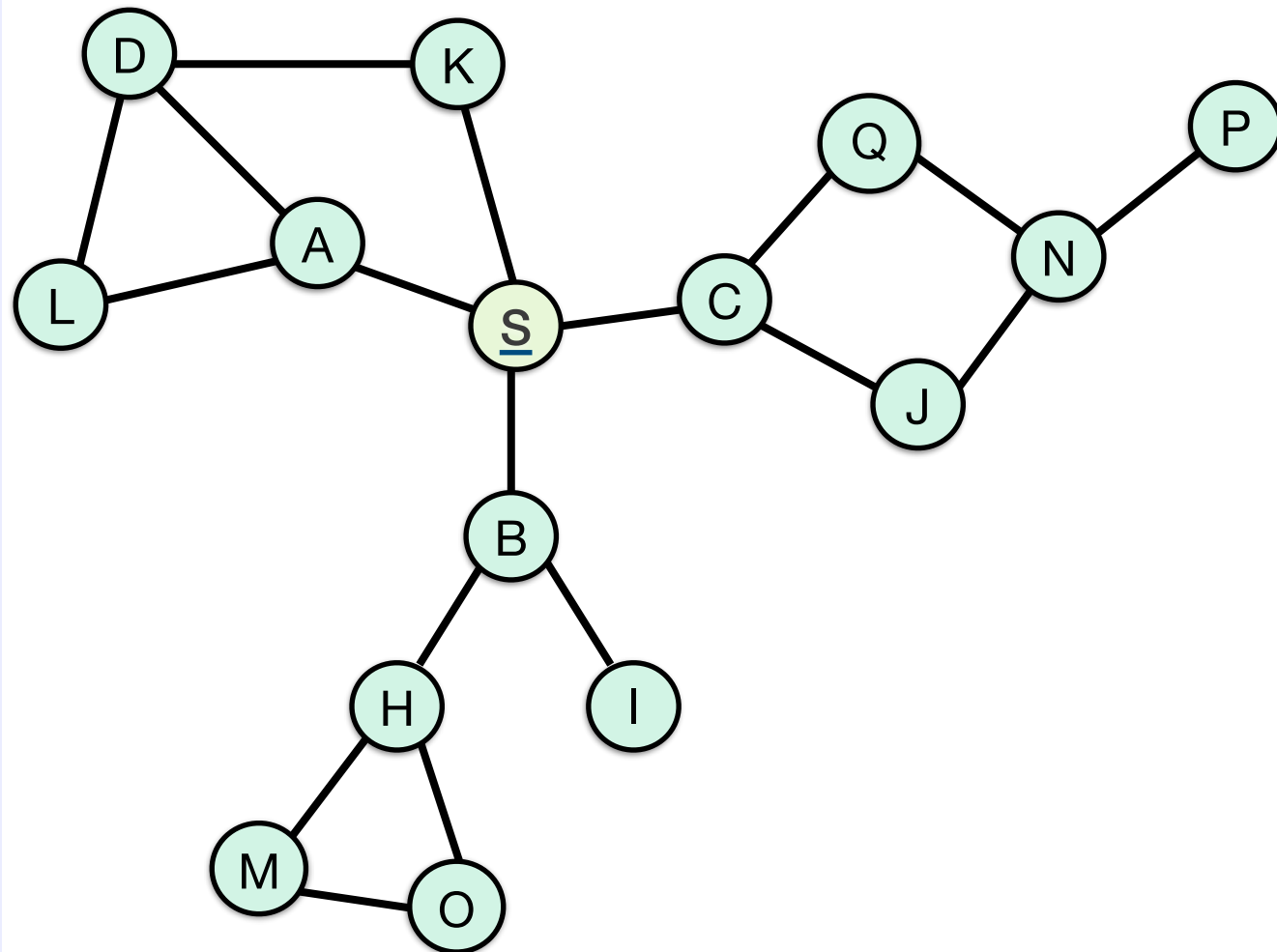
For an **undirected** connected graph, a rooted spanning tree is said to be DFS tree if

For each non-tree edge (a, b) ,
either
 a is ancestor of b , or vice-versa.



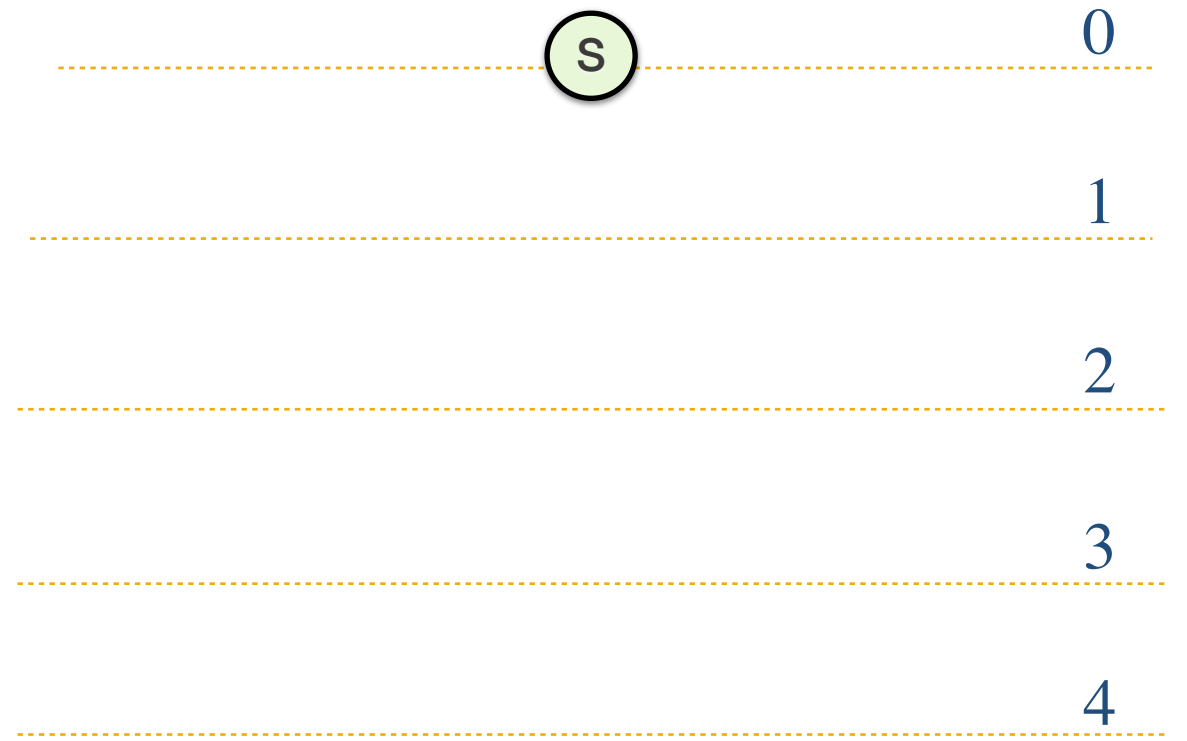
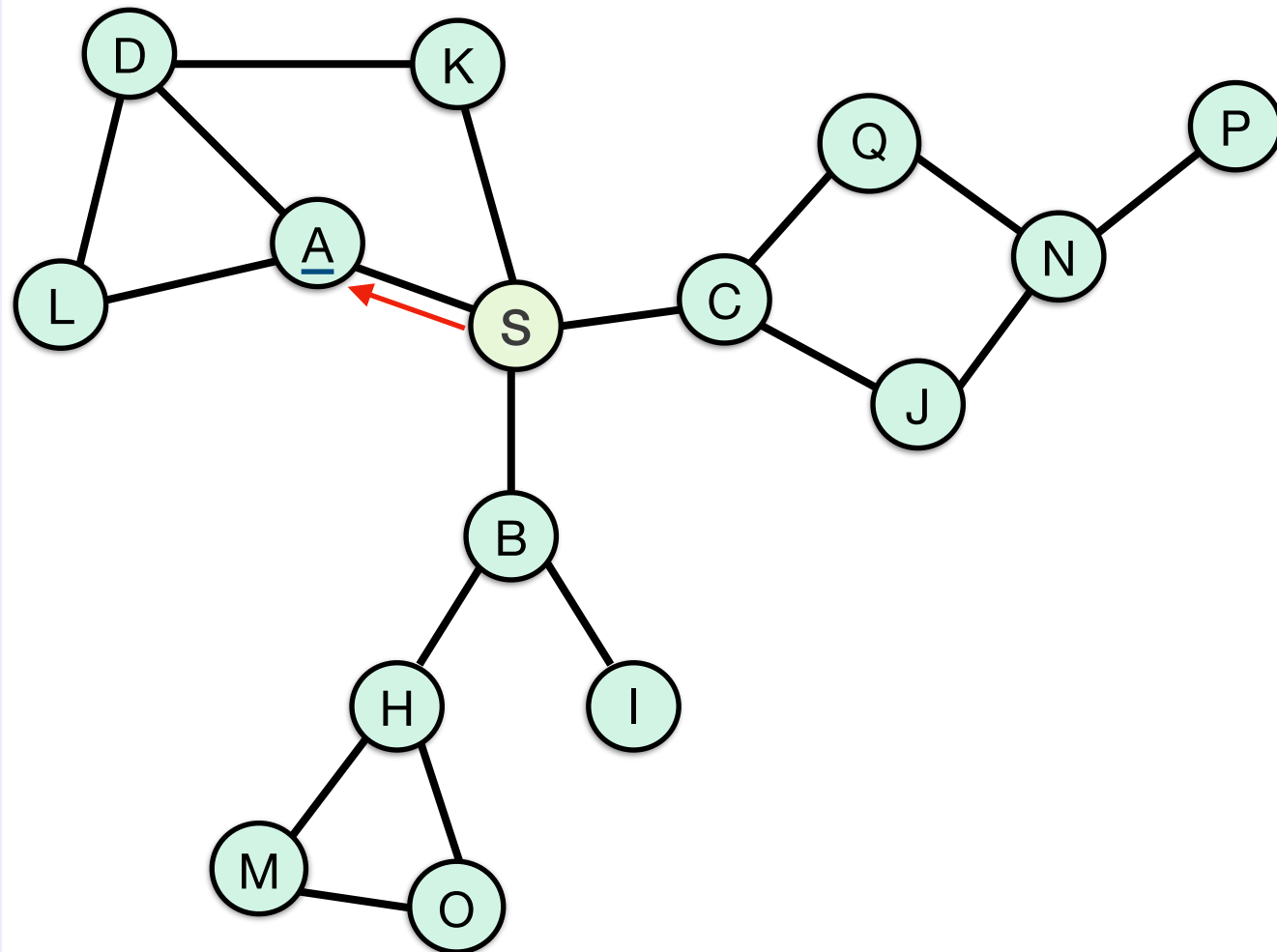
DFS tree
of an undirected graph

DFS Traversal and DFS Tree



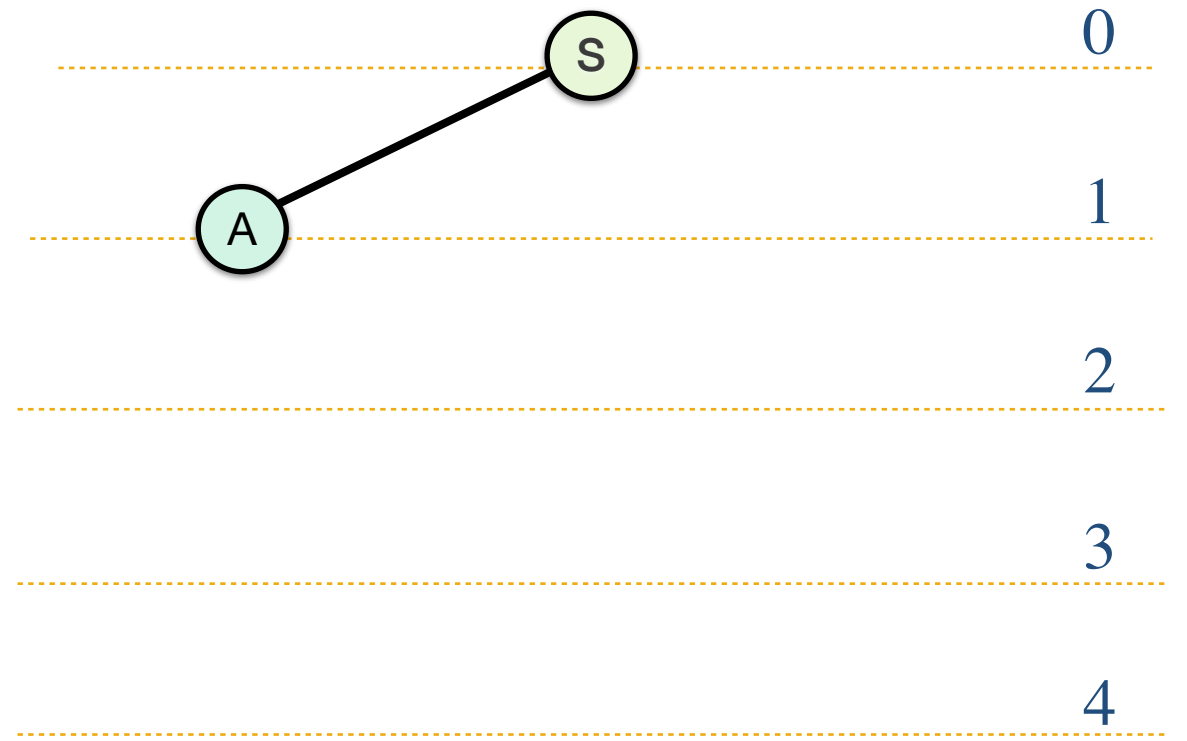
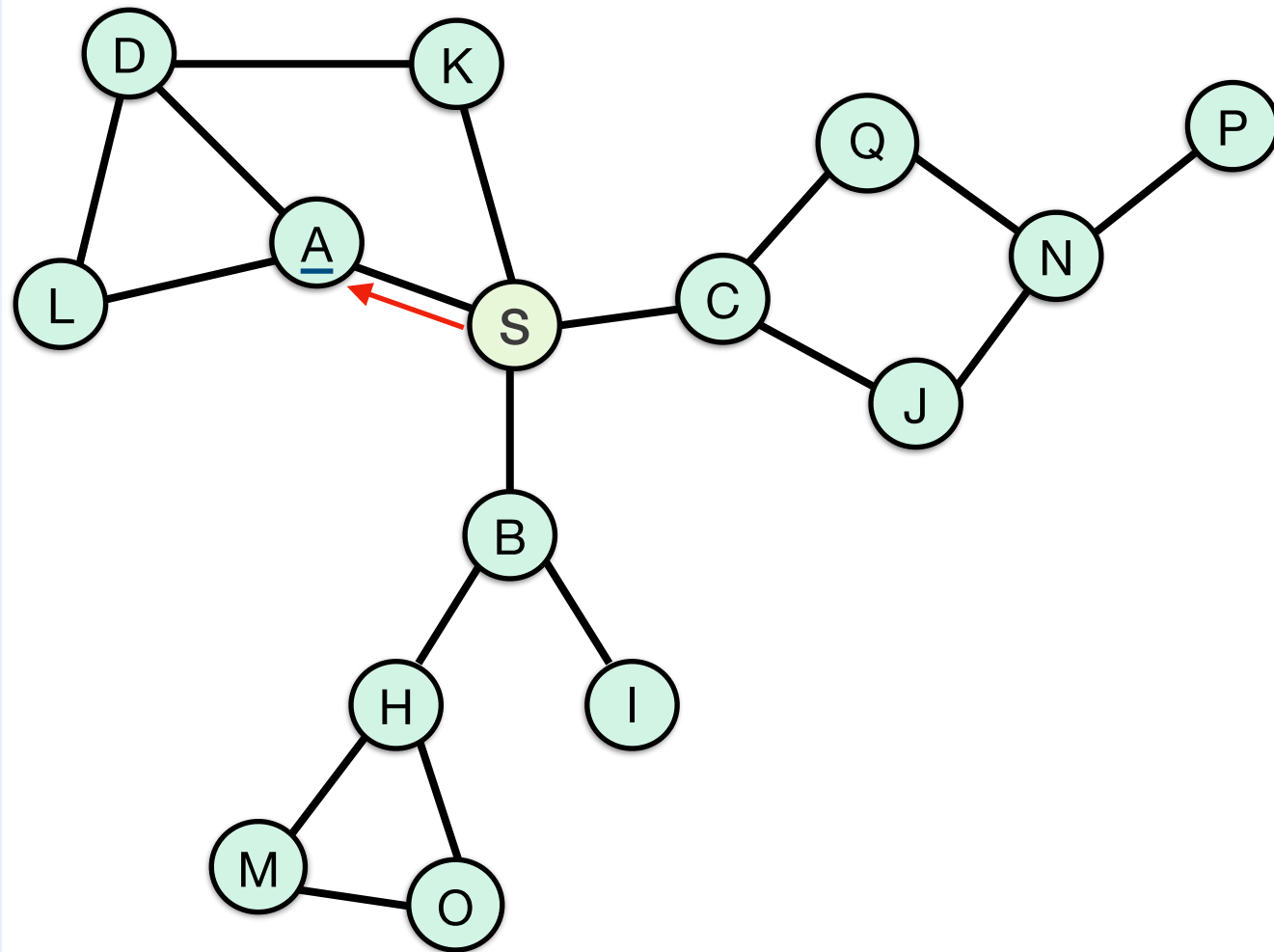
DFS tree

DFS Traversal and DFS Tree



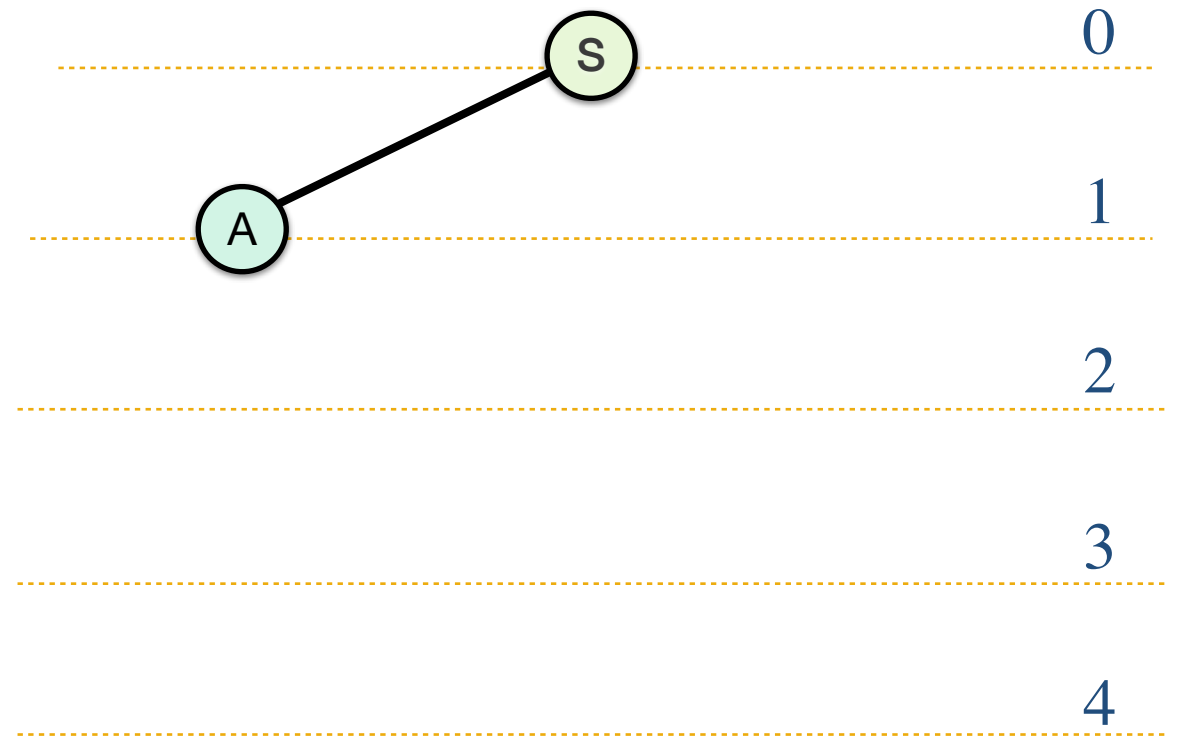
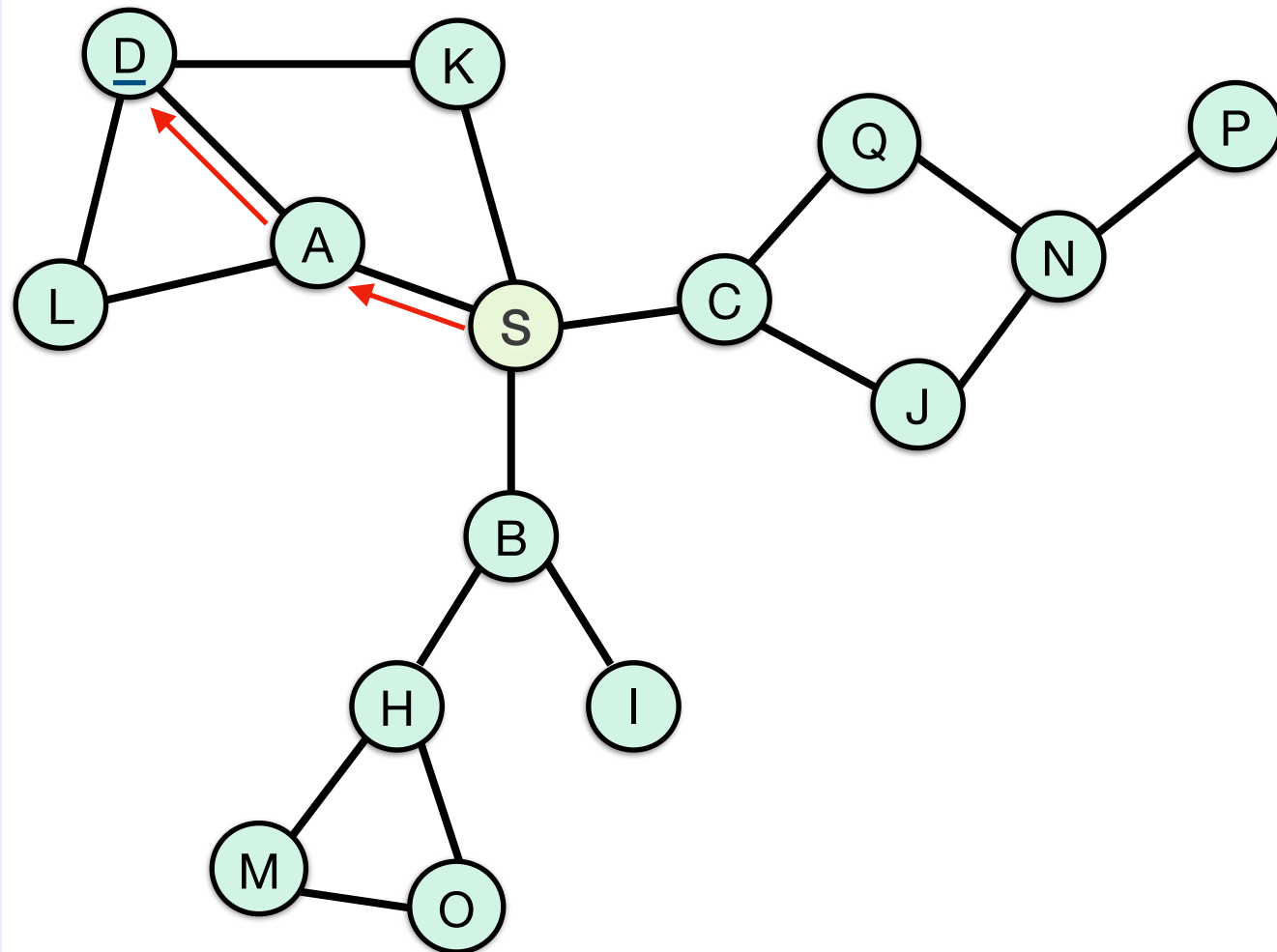
DFS tree

DFS Traversal and DFS Tree



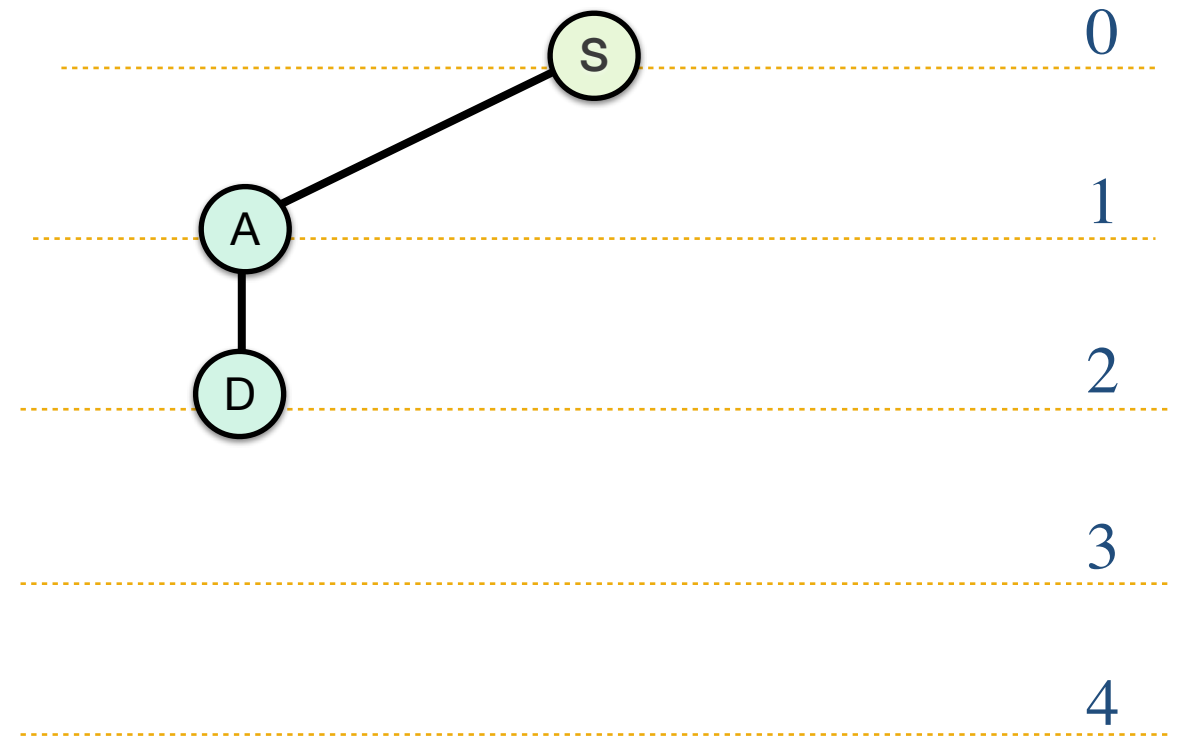
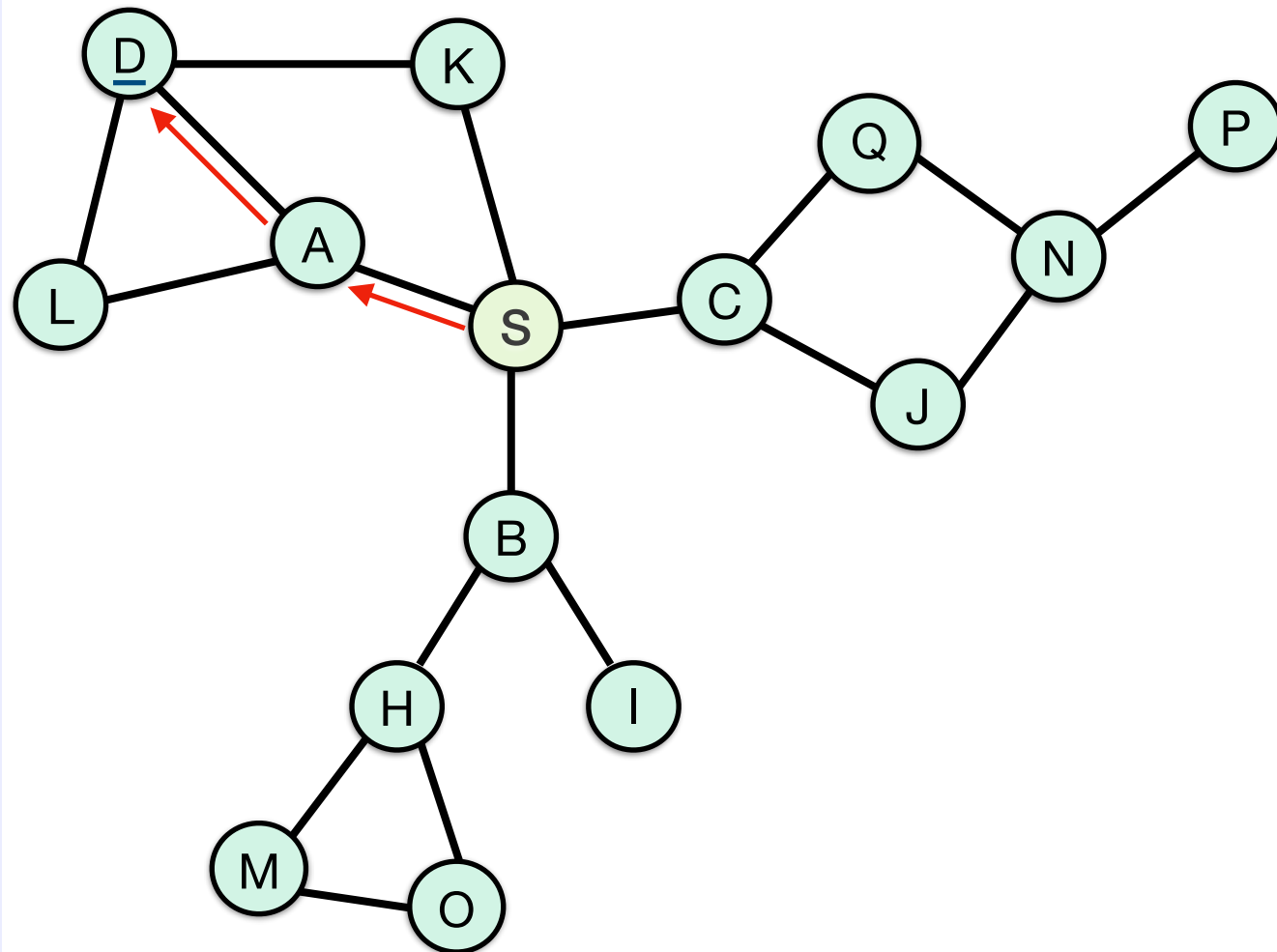
DFS tree

DFS Traversal and DFS Tree



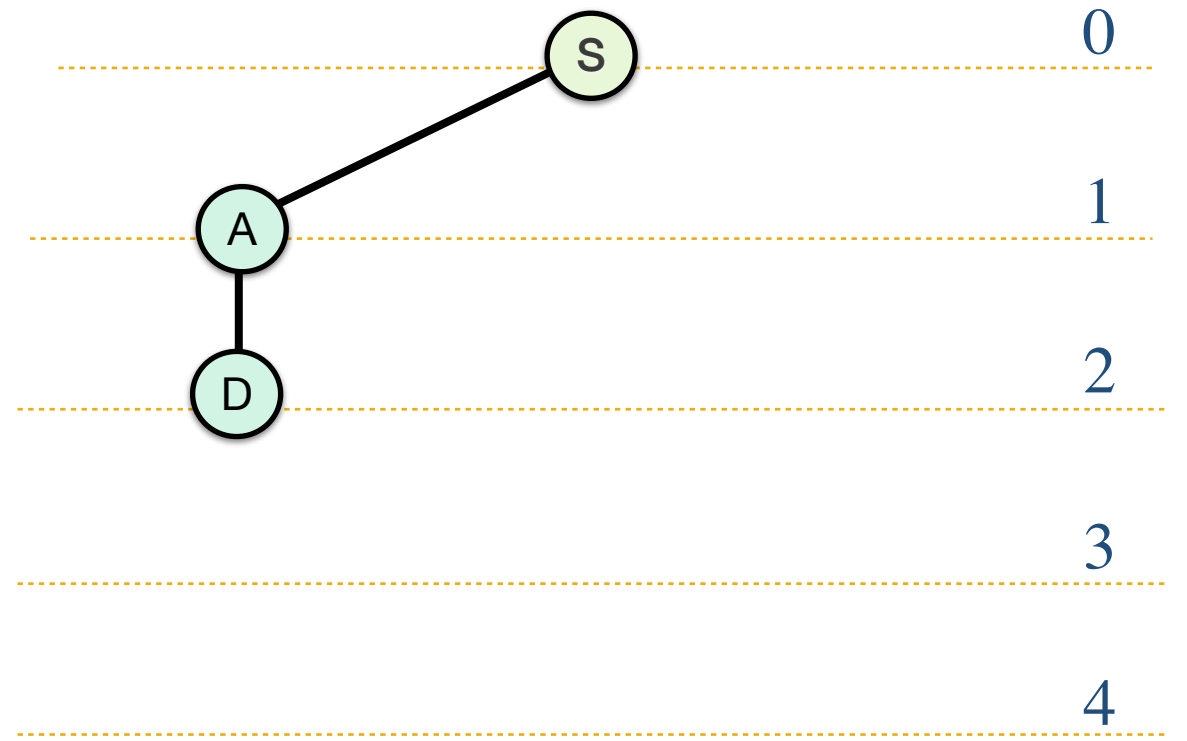
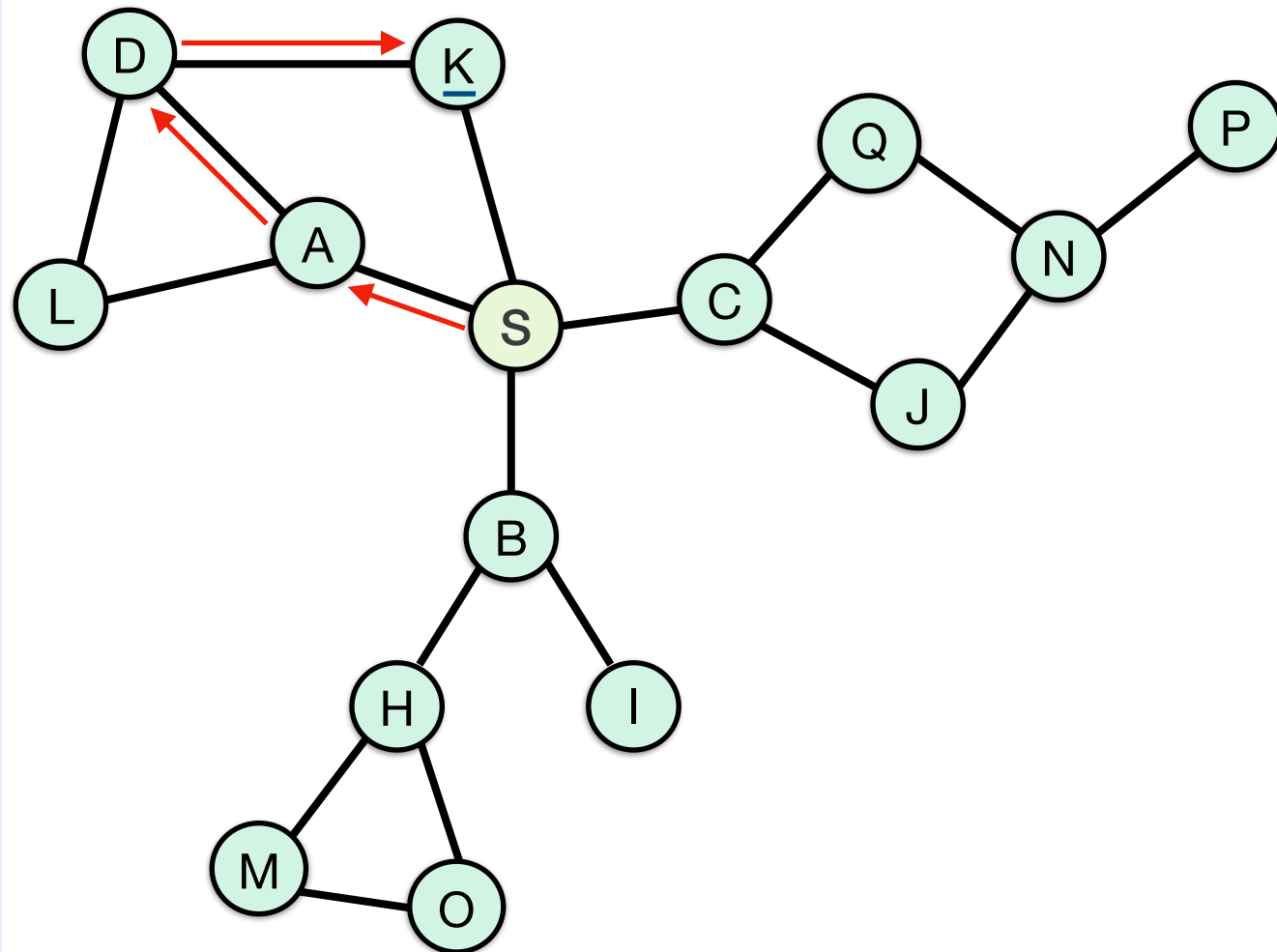
DFS tree

DFS Traversal and DFS Tree



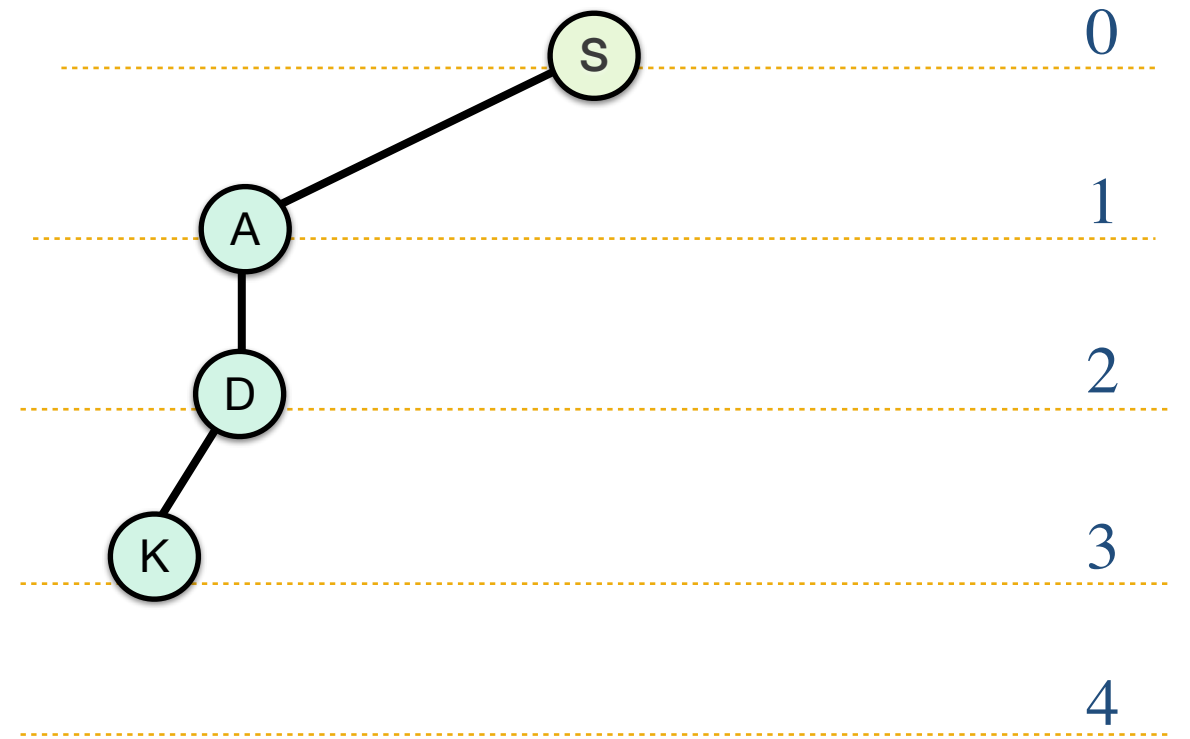
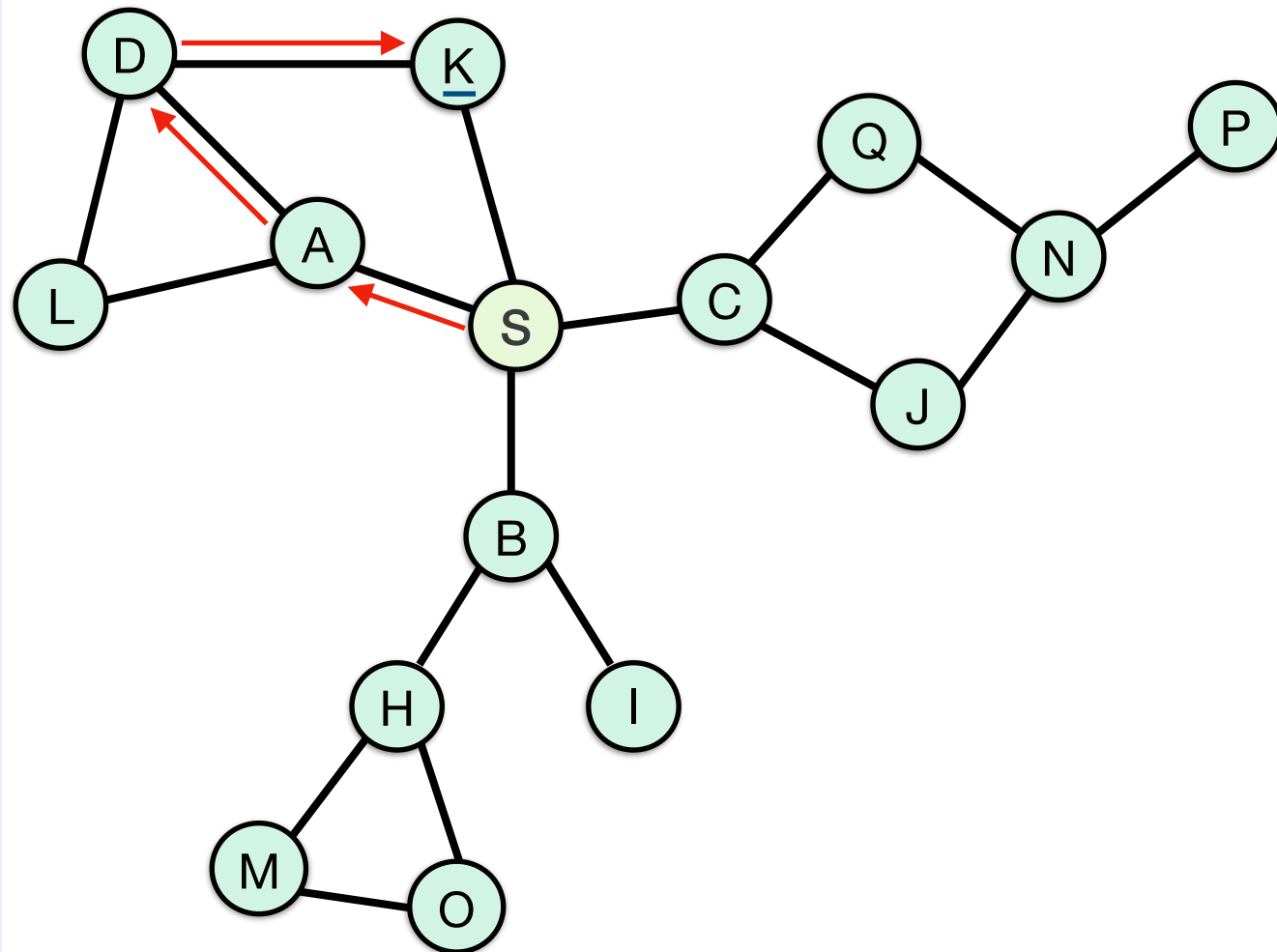
DFS tree

DFS Traversal and DFS Tree



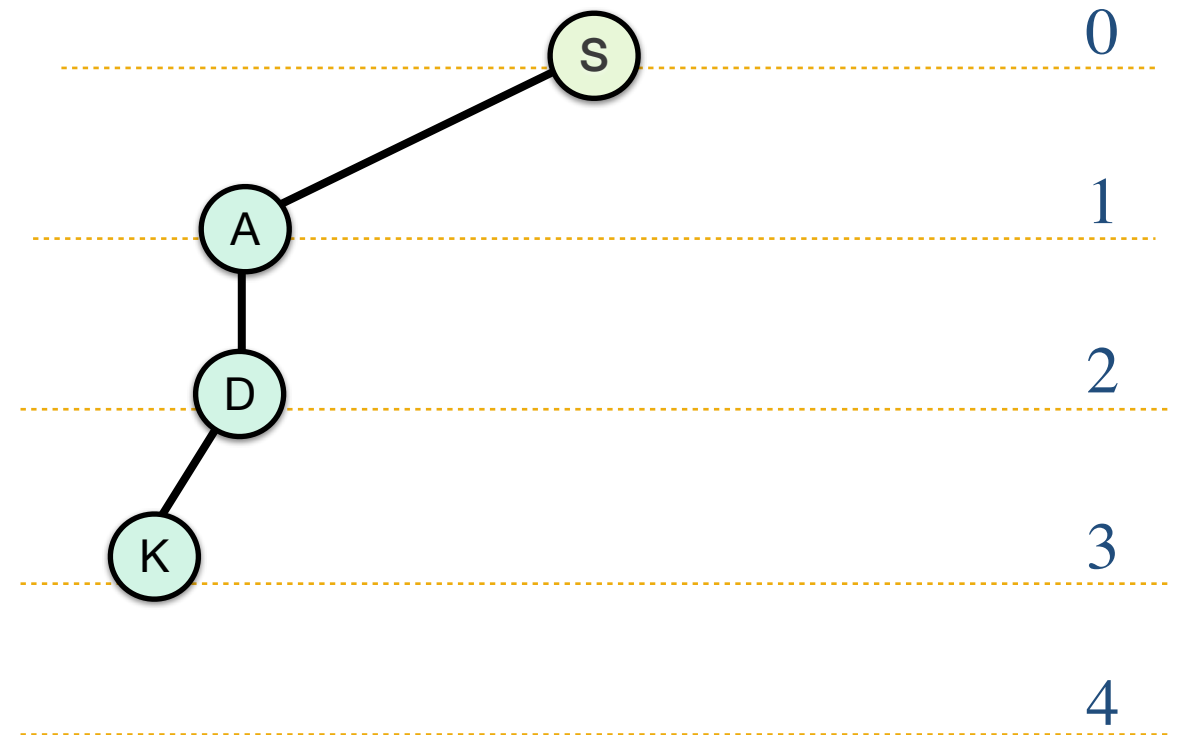
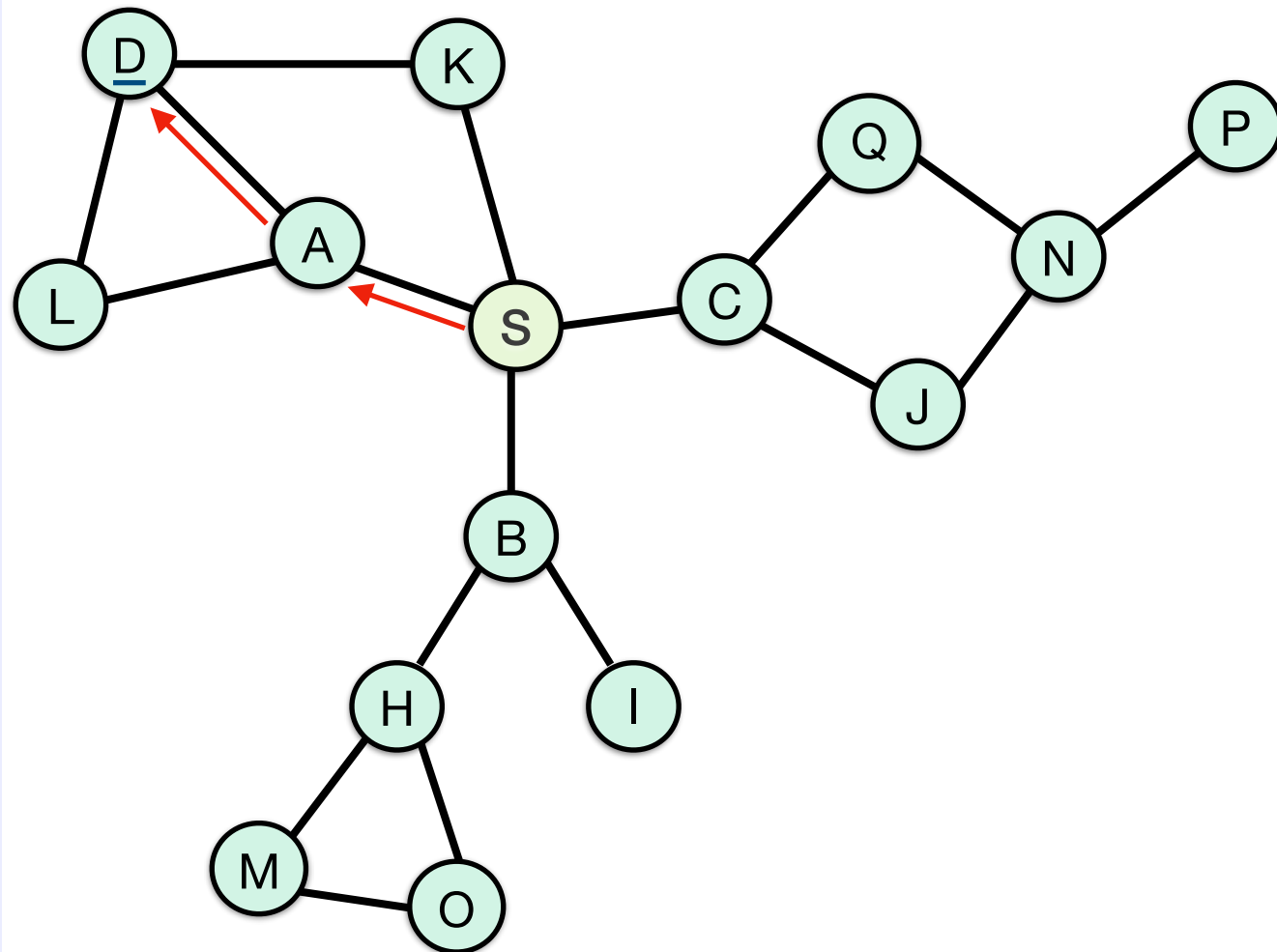
DFS tree

DFS Traversal and DFS Tree



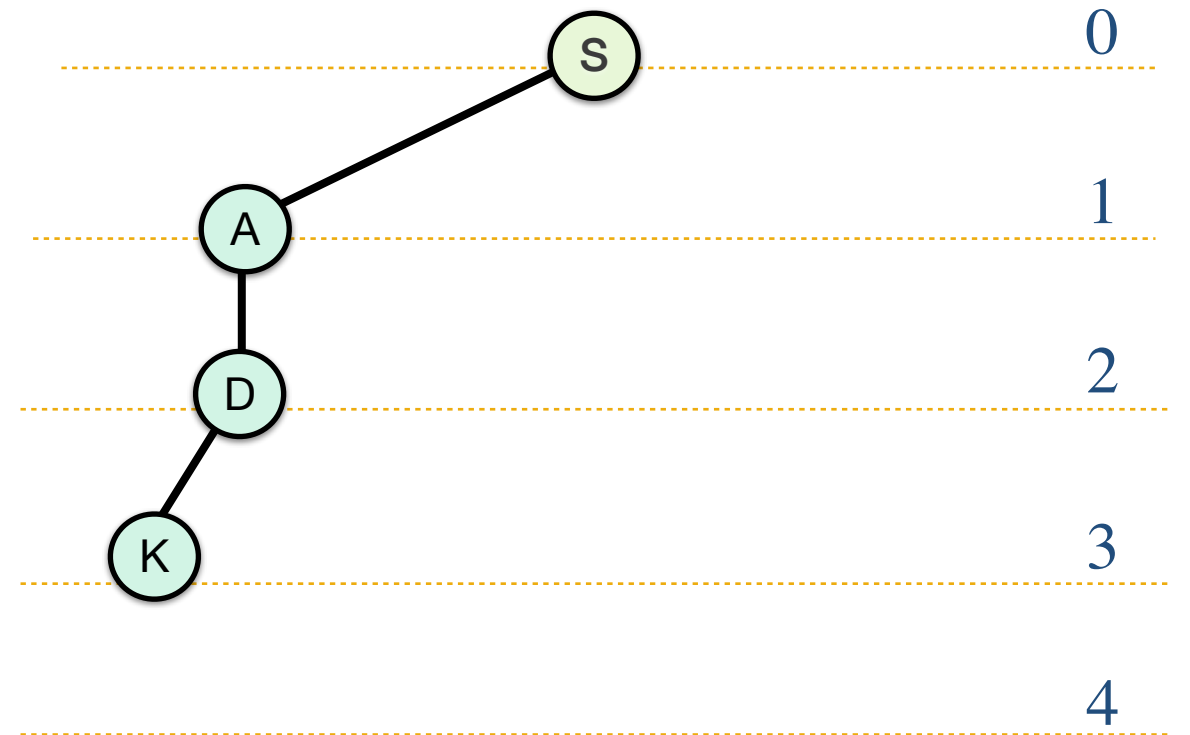
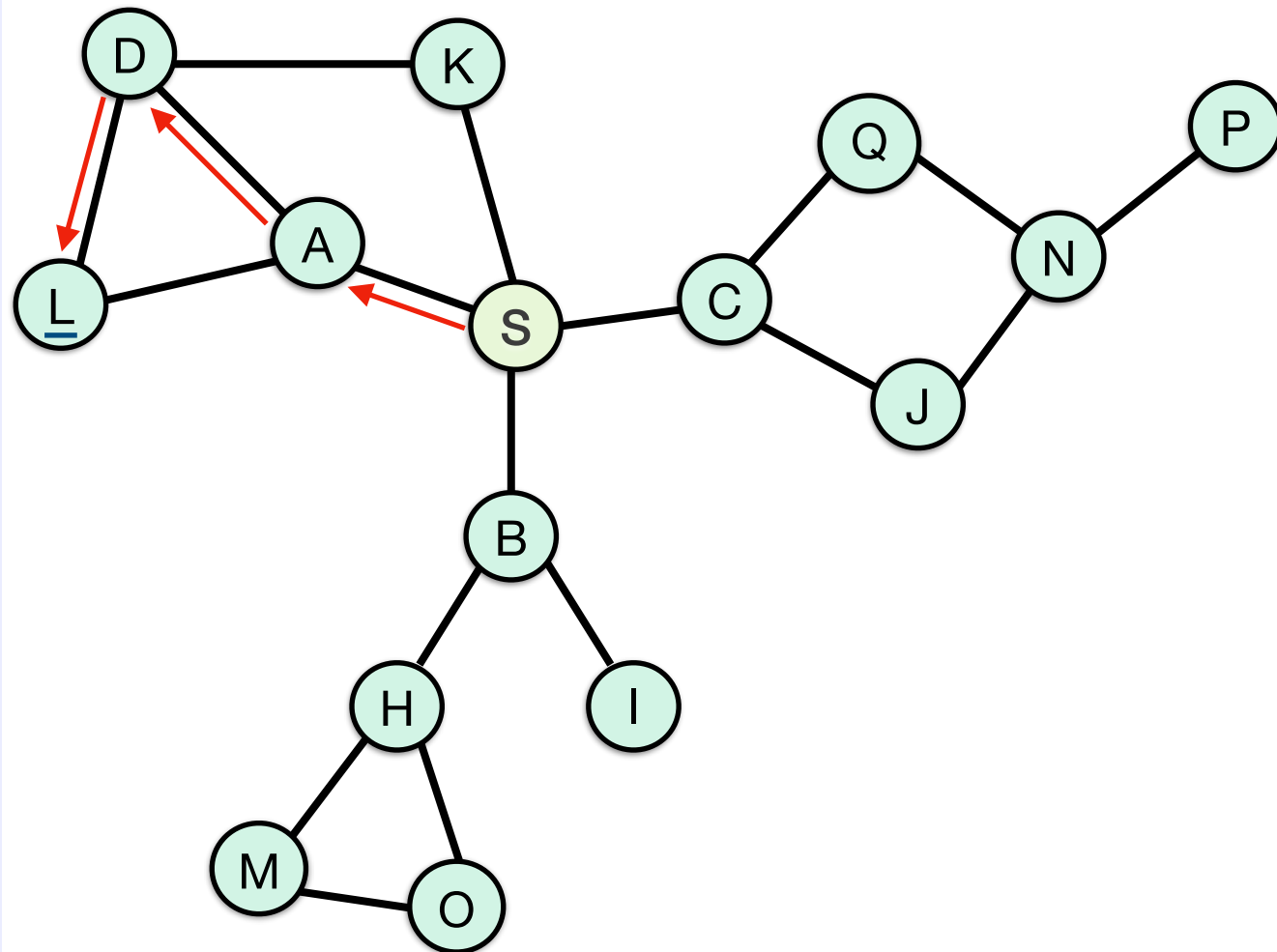
DFS tree

DFS Traversal and DFS Tree



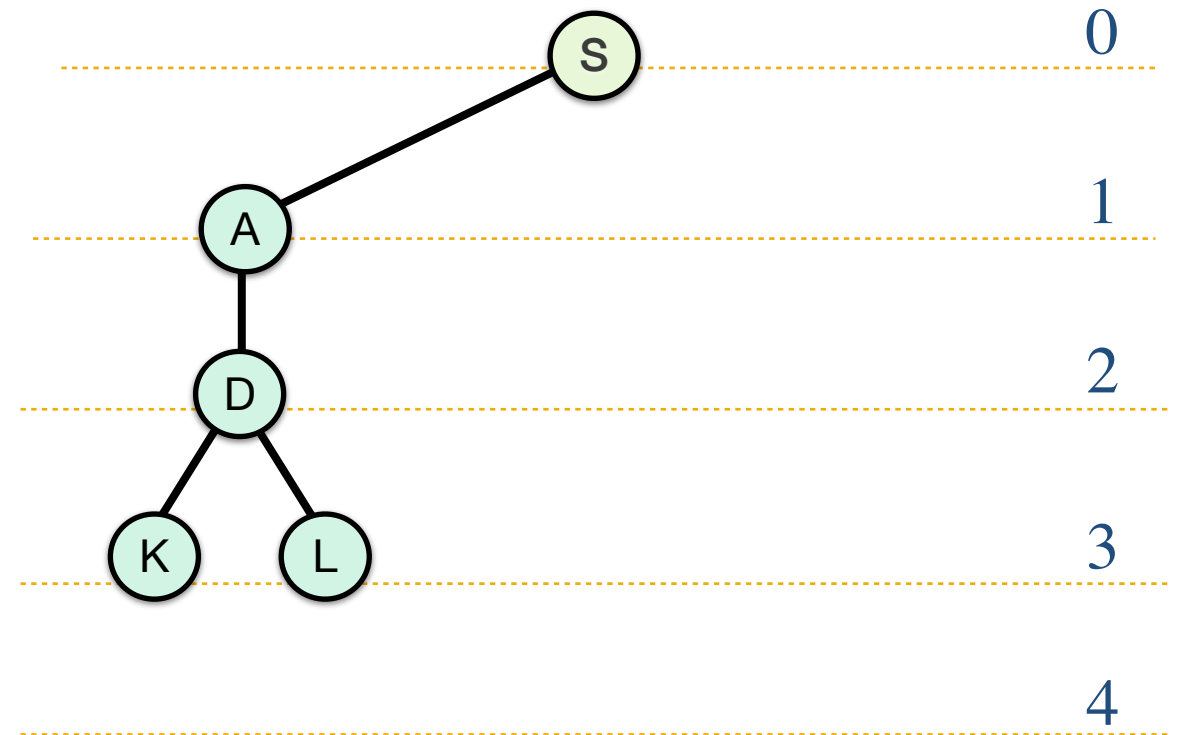
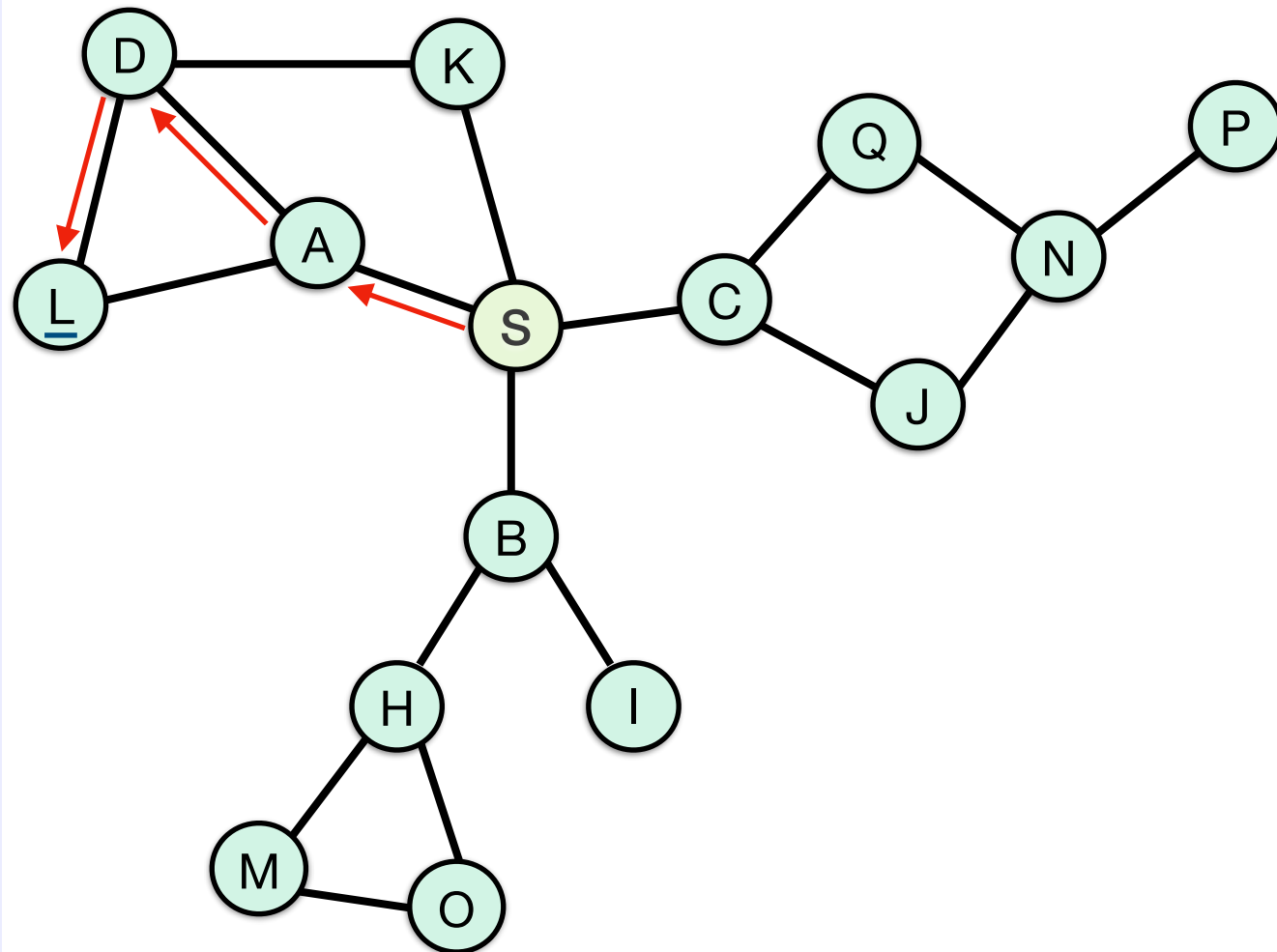
DFS tree

DFS Traversal and DFS Tree



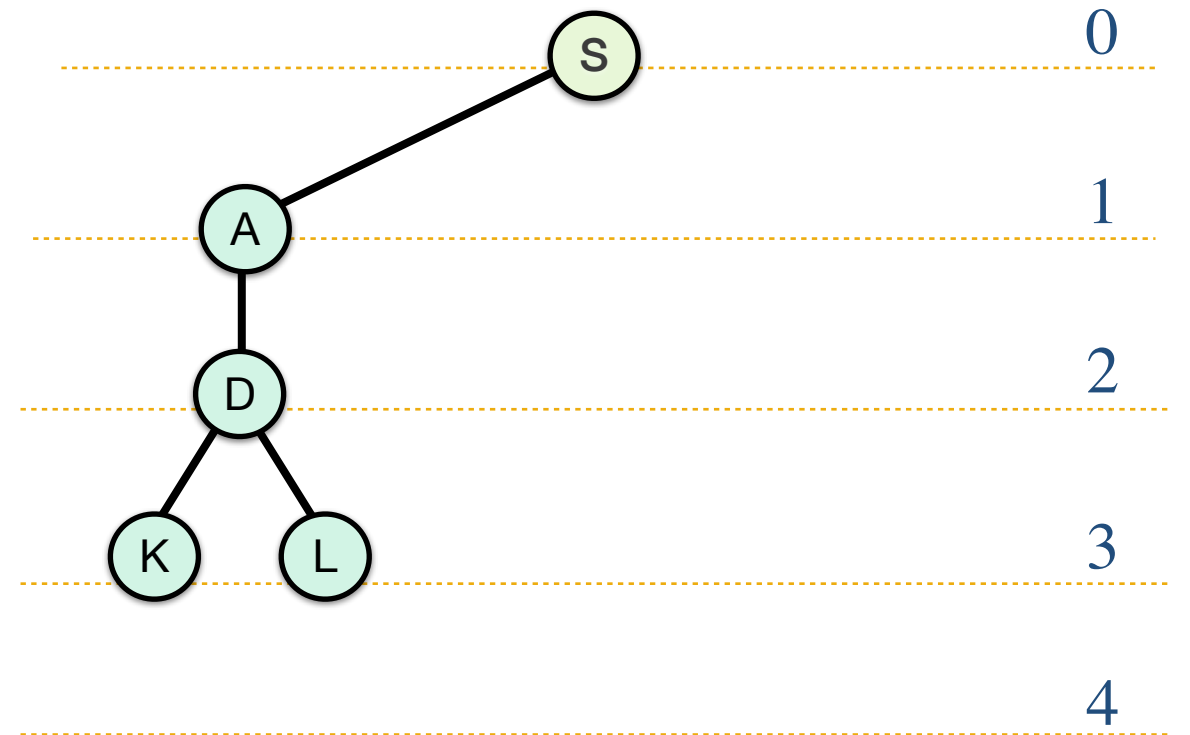
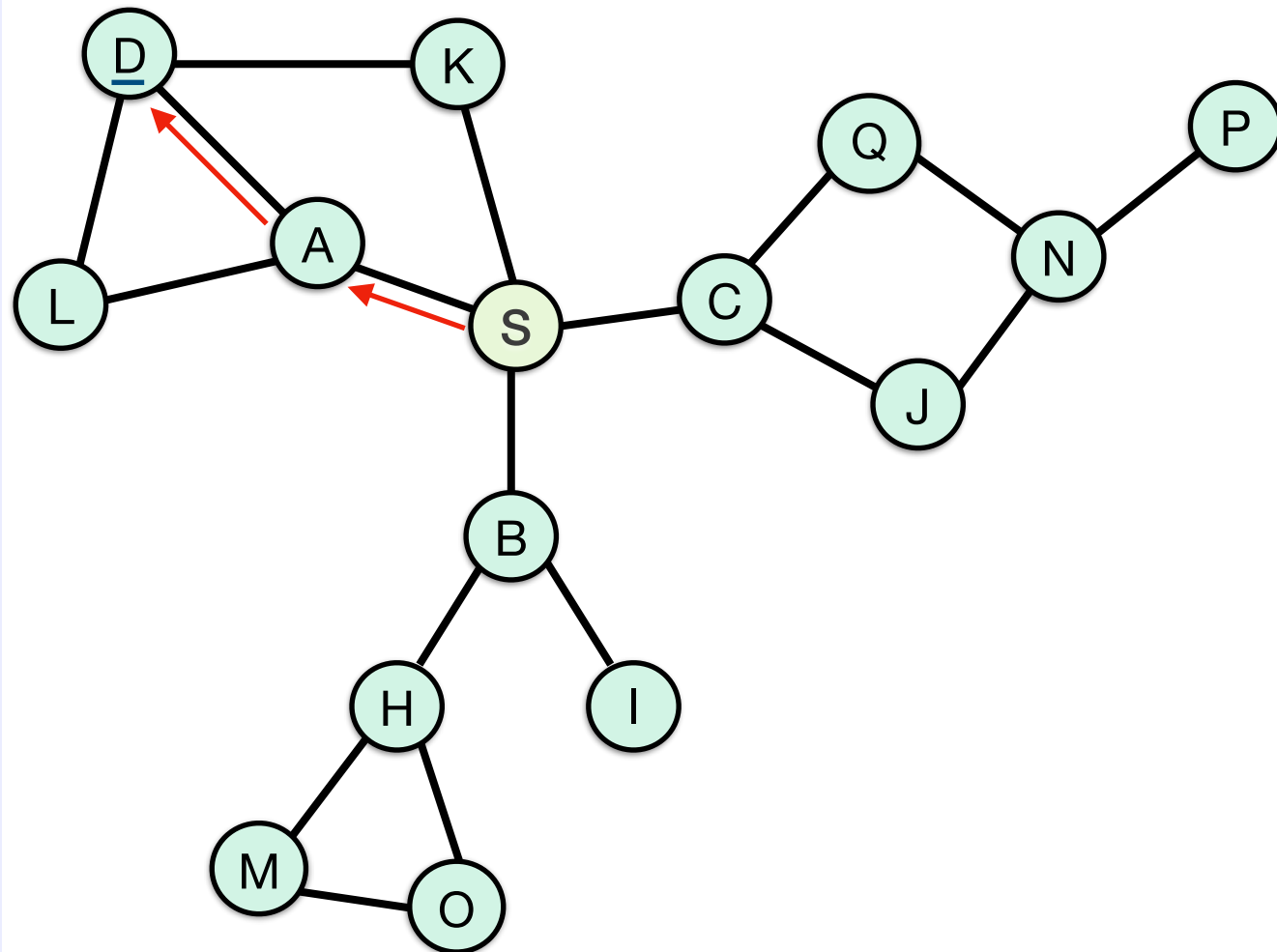
DFS tree

DFS Traversal and DFS Tree



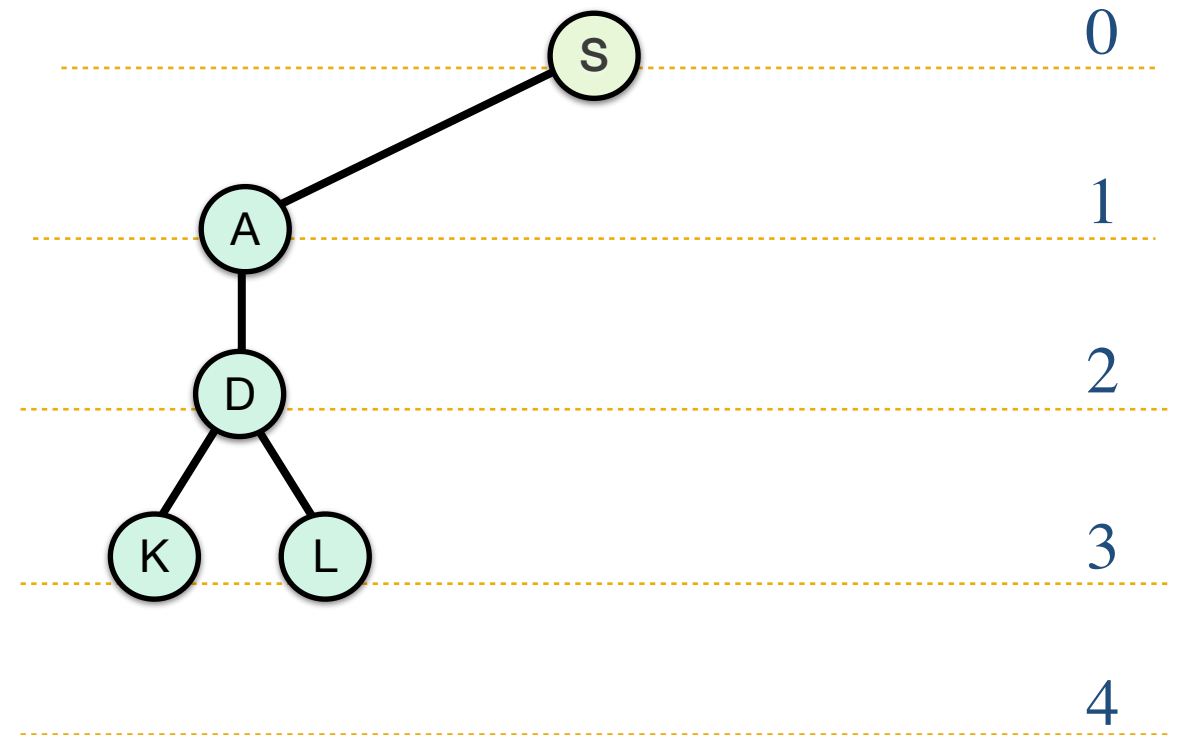
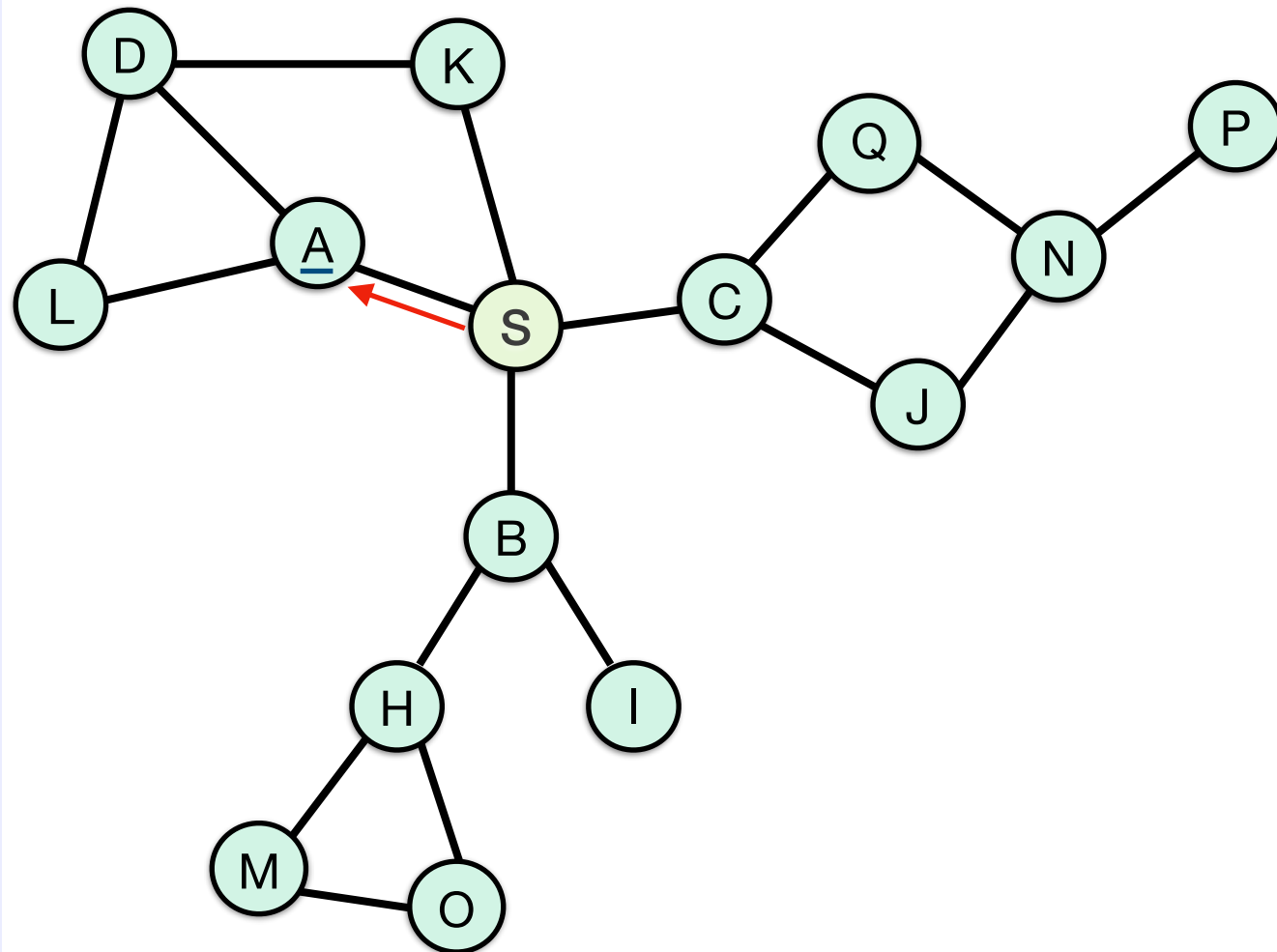
DFS tree

DFS Traversal and DFS Tree



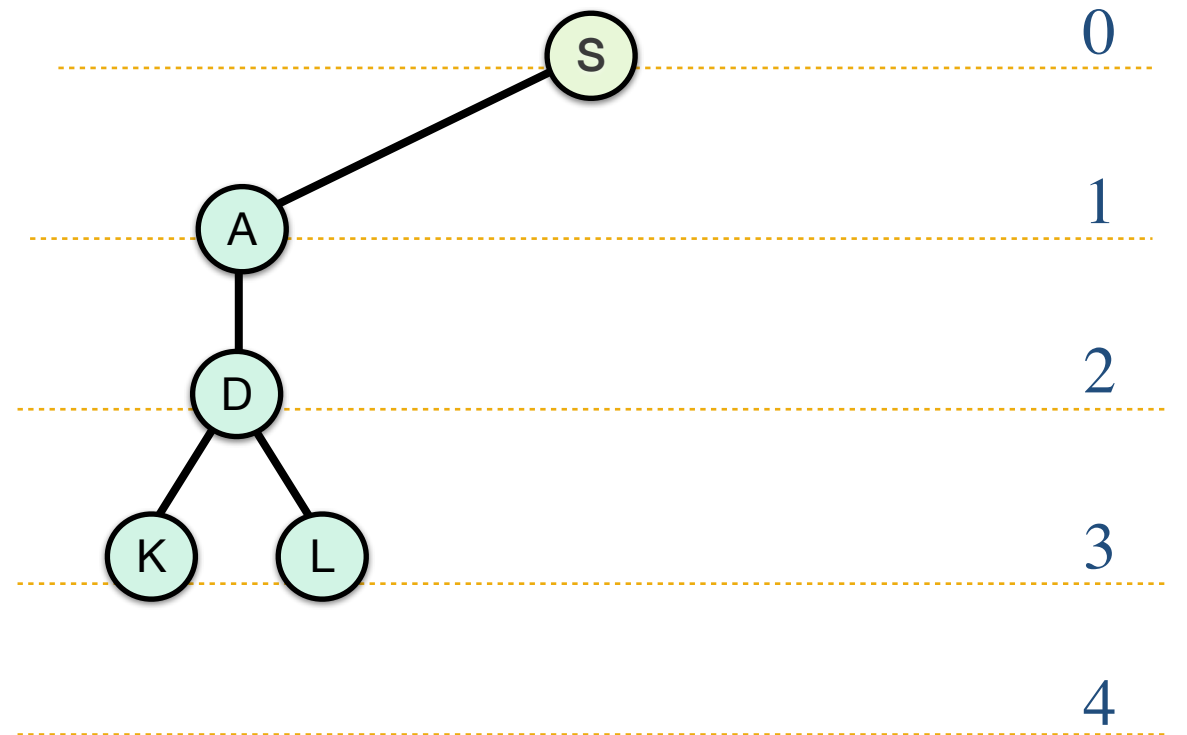
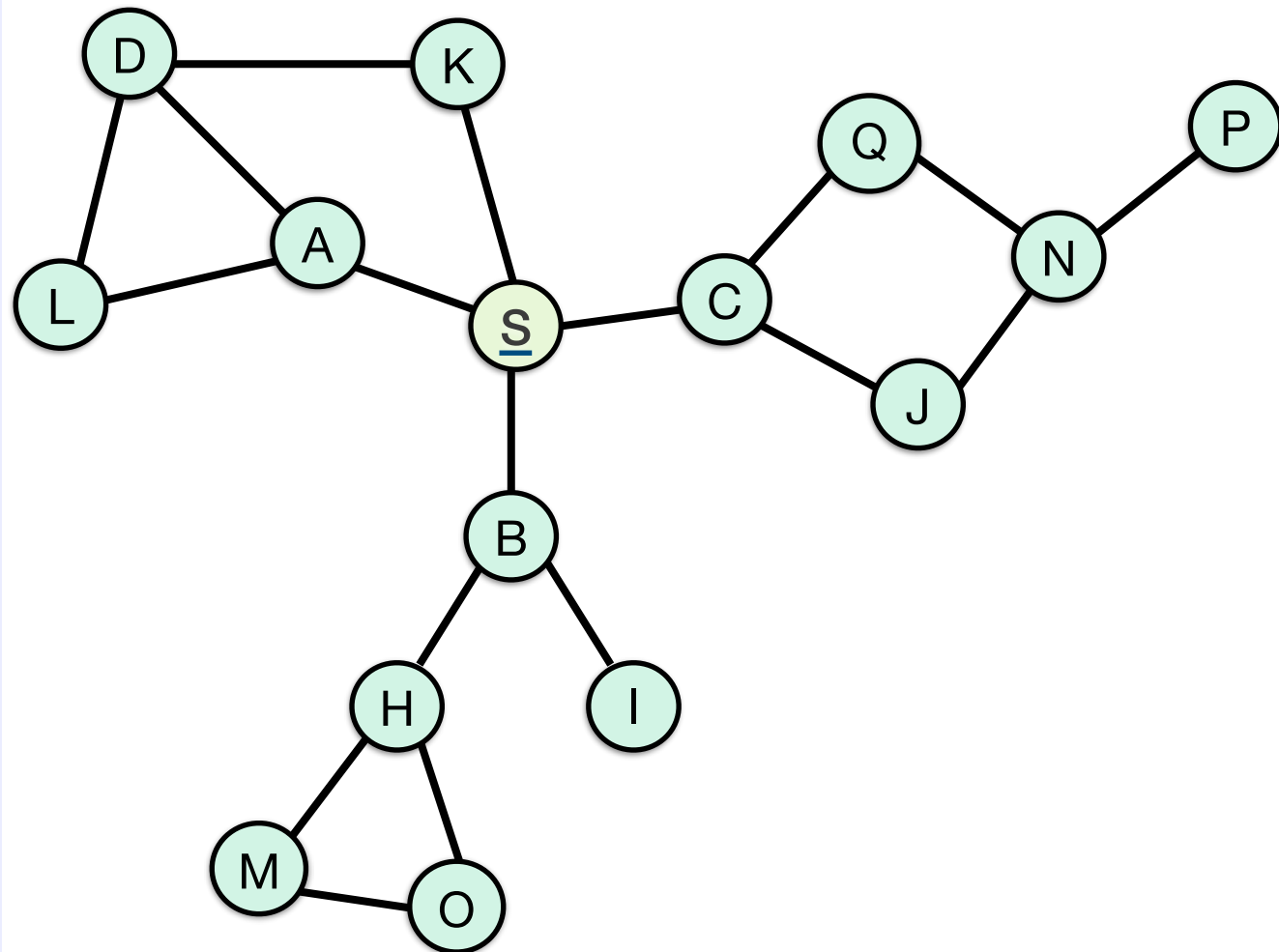
DFS tree

DFS Traversal and DFS Tree



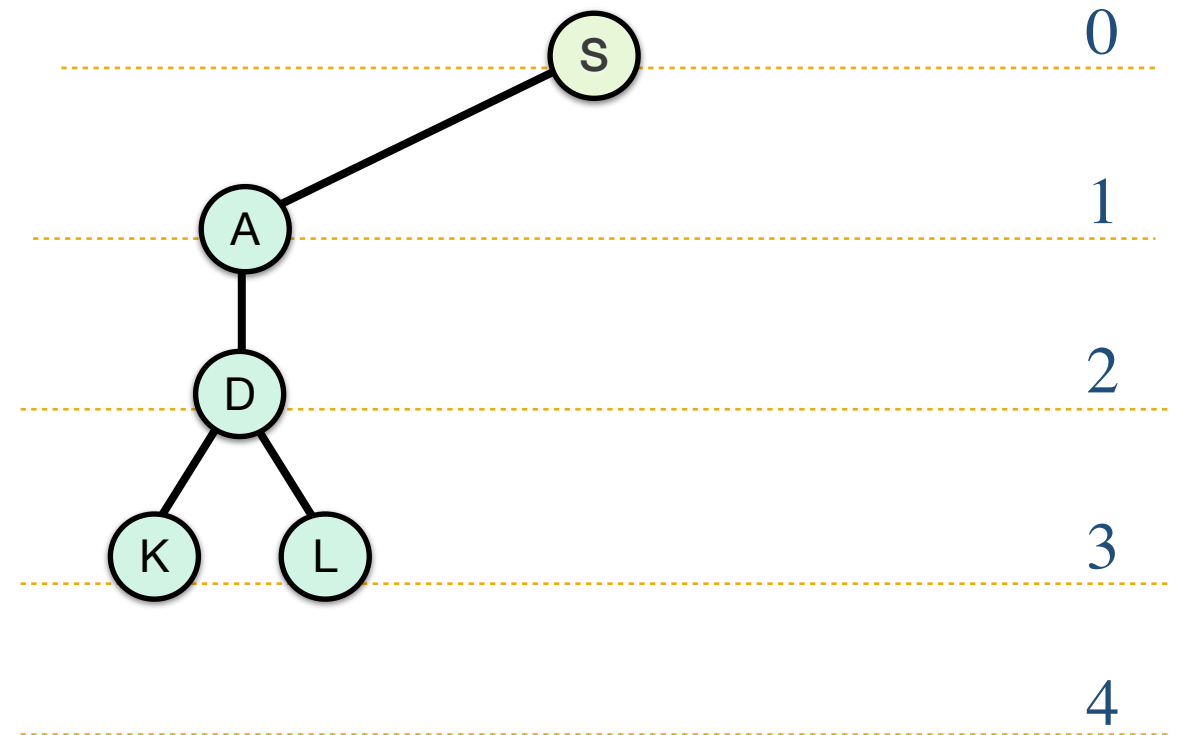
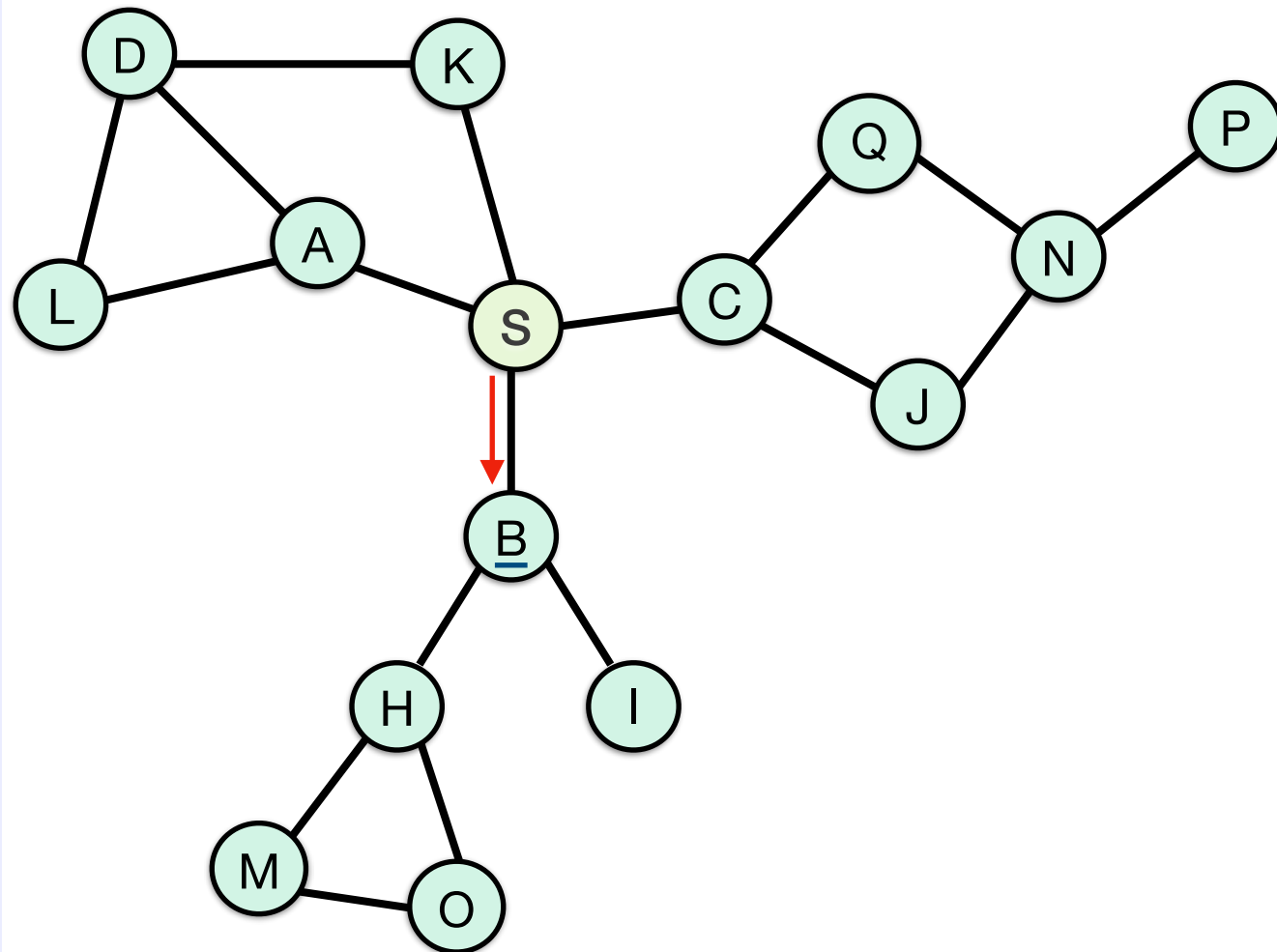
DFS tree

DFS Traversal and DFS Tree



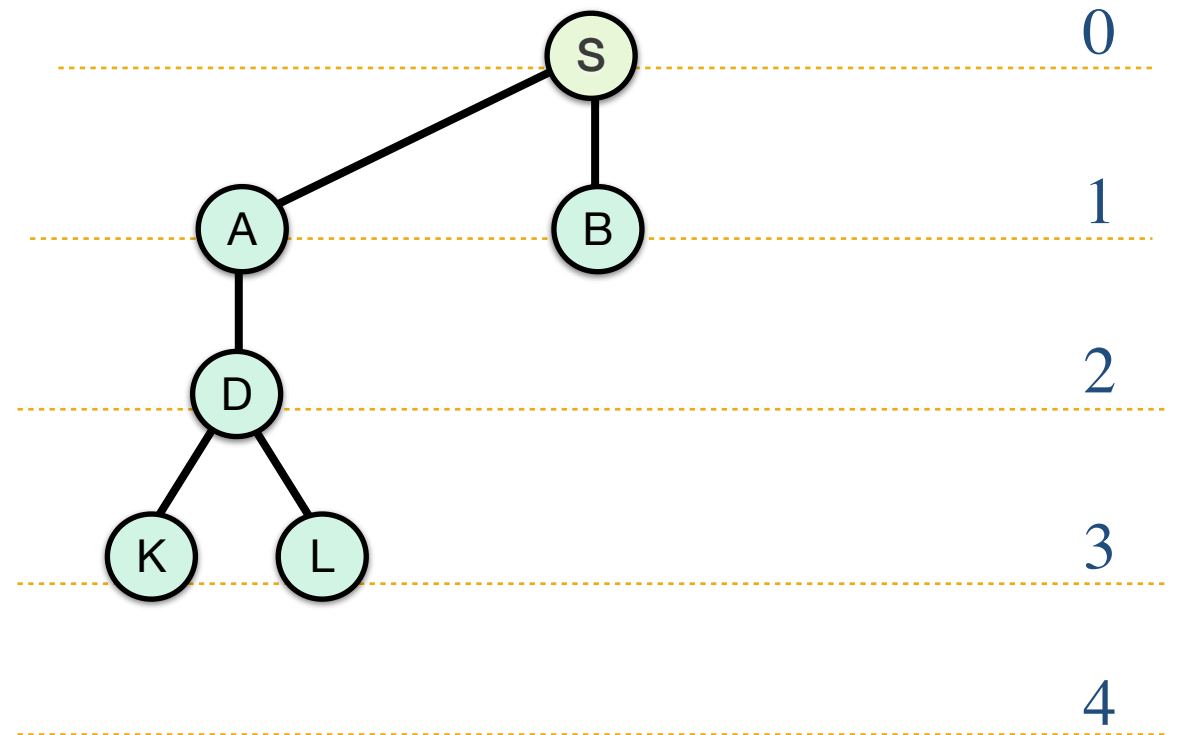
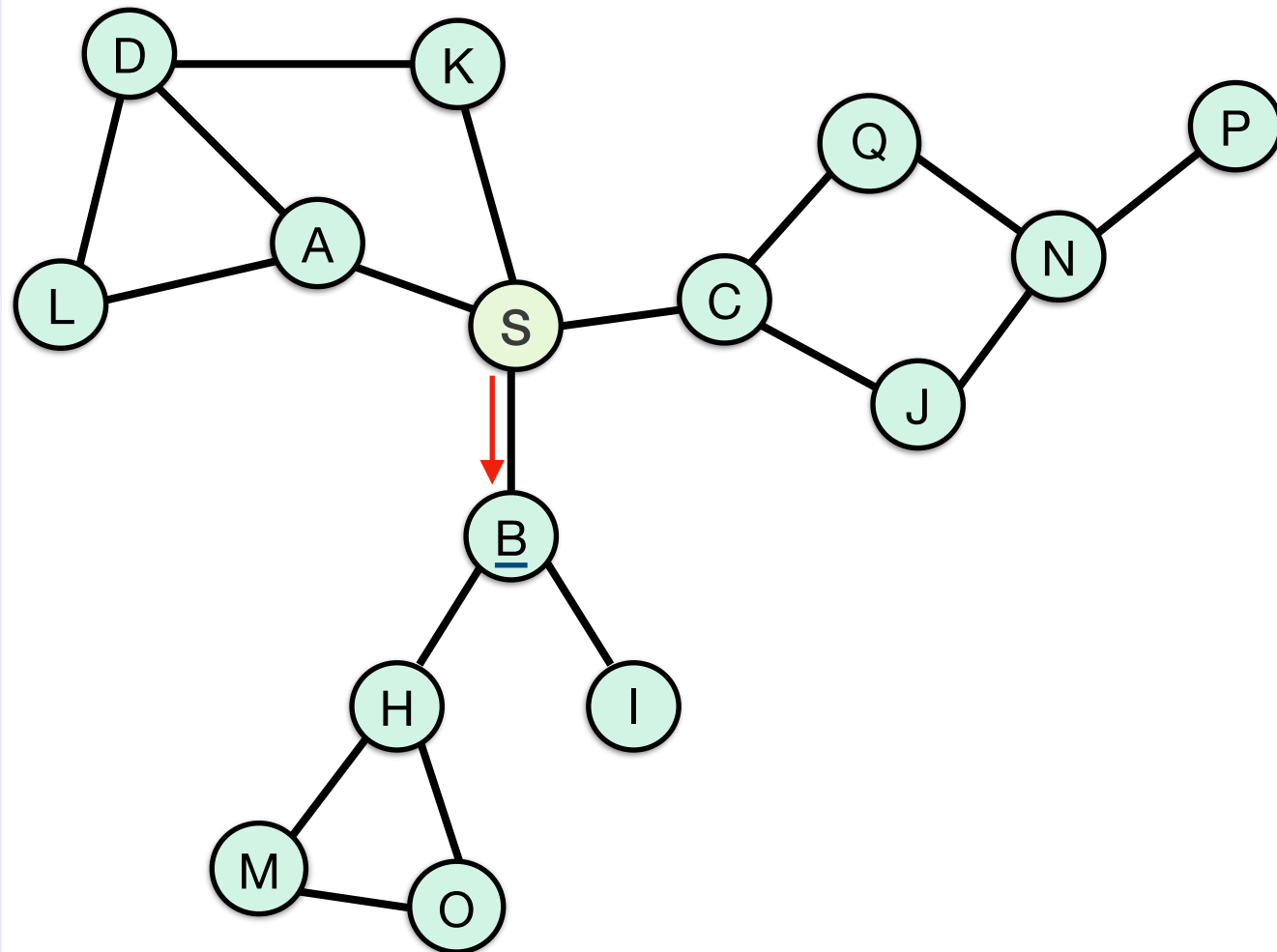
DFS tree

DFS Traversal and DFS Tree



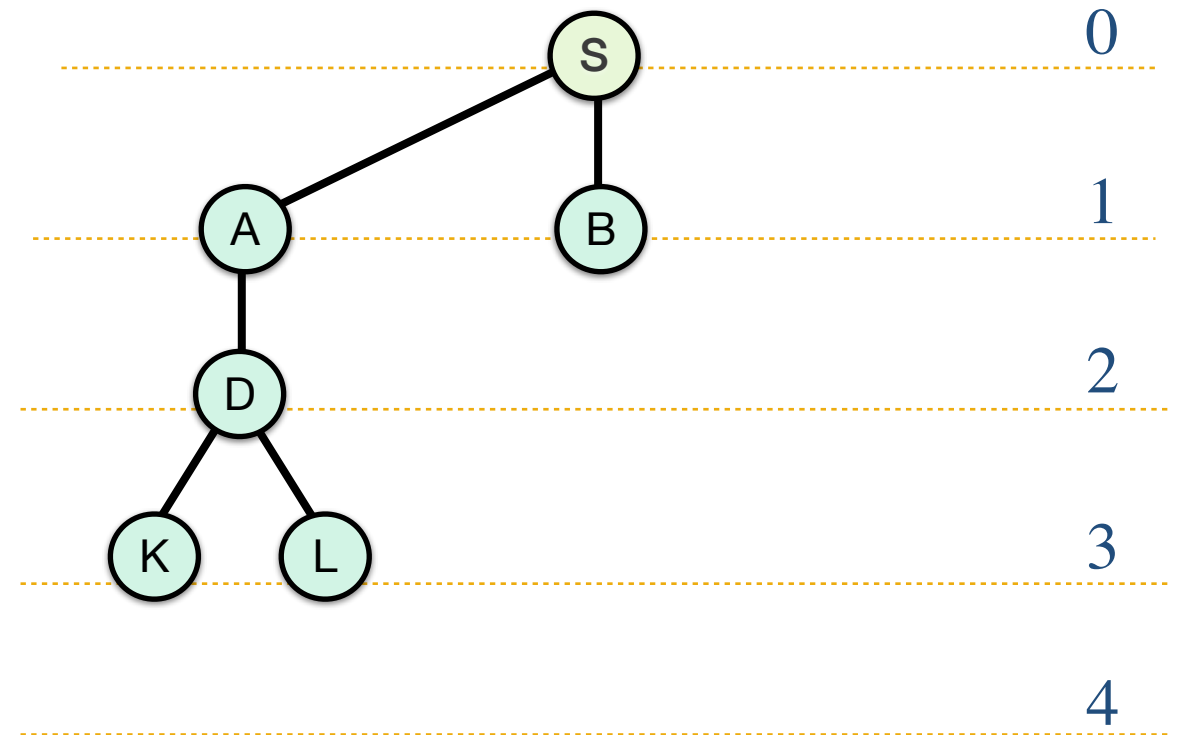
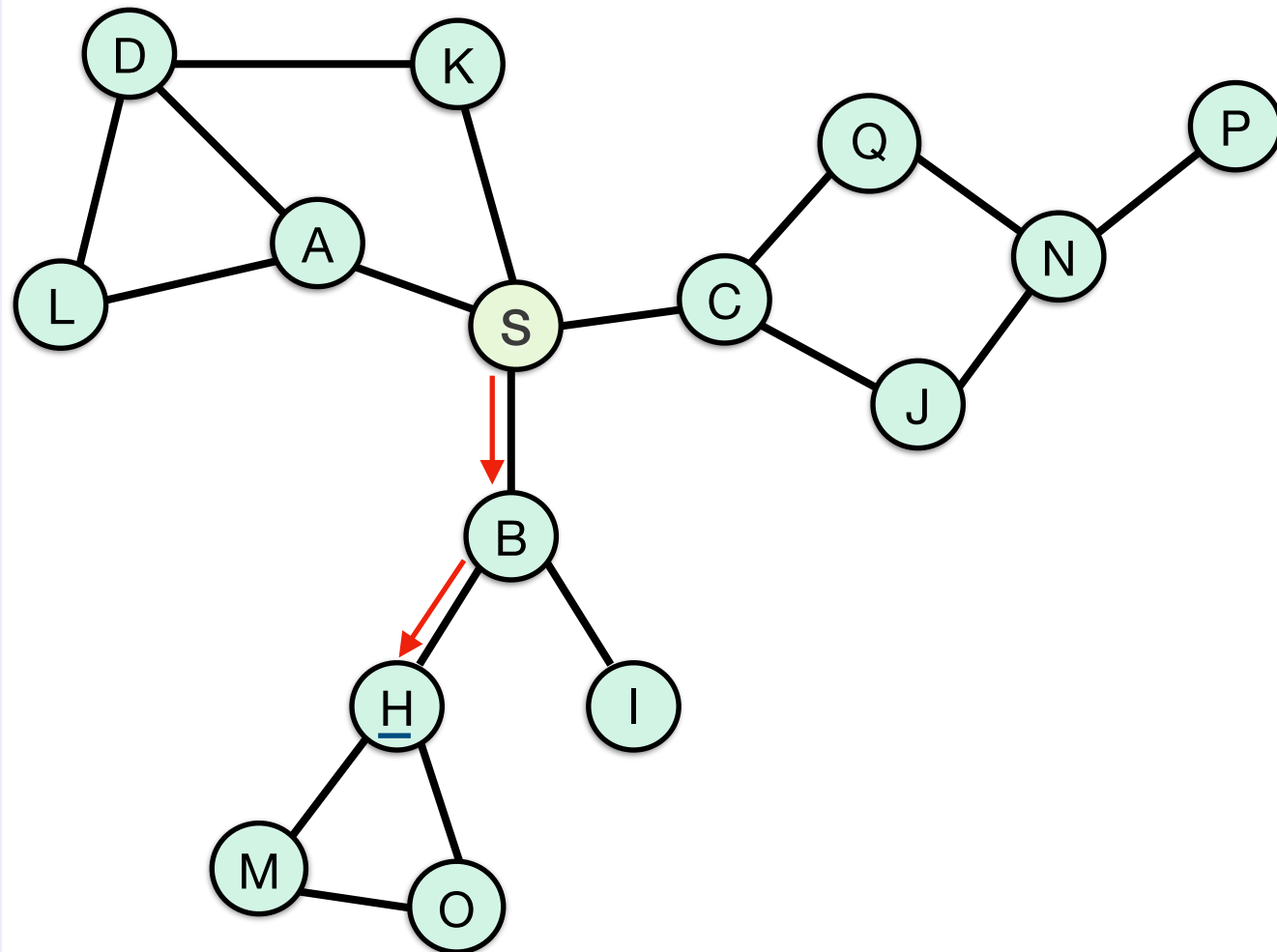
DFS tree

DFS Traversal and DFS Tree



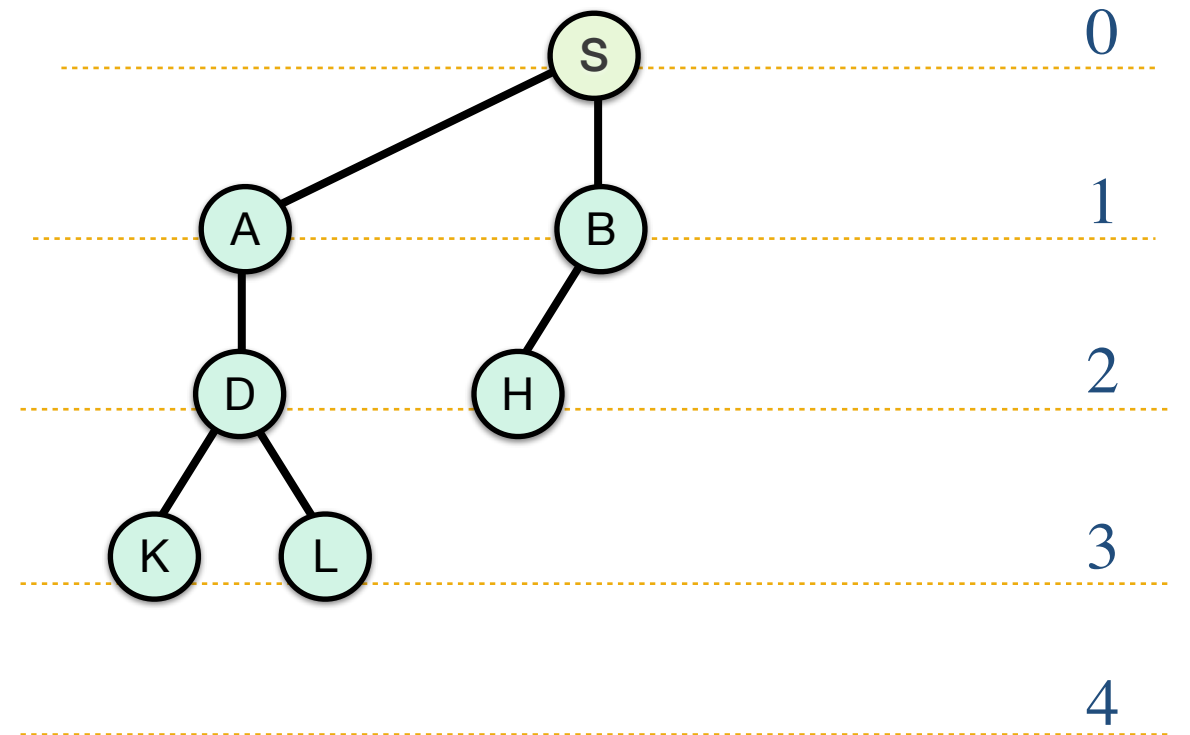
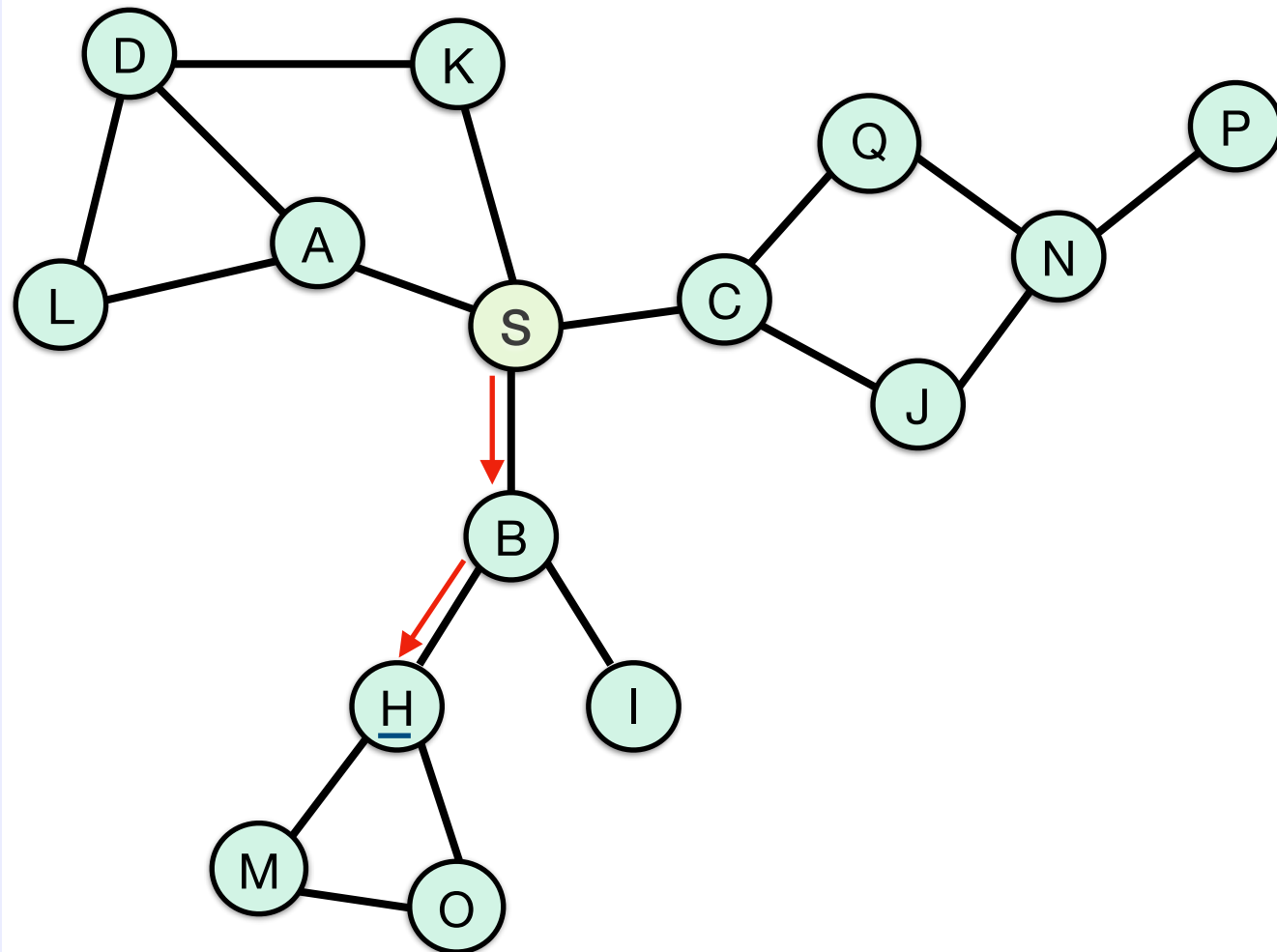
DFS tree

DFS Traversal and DFS Tree



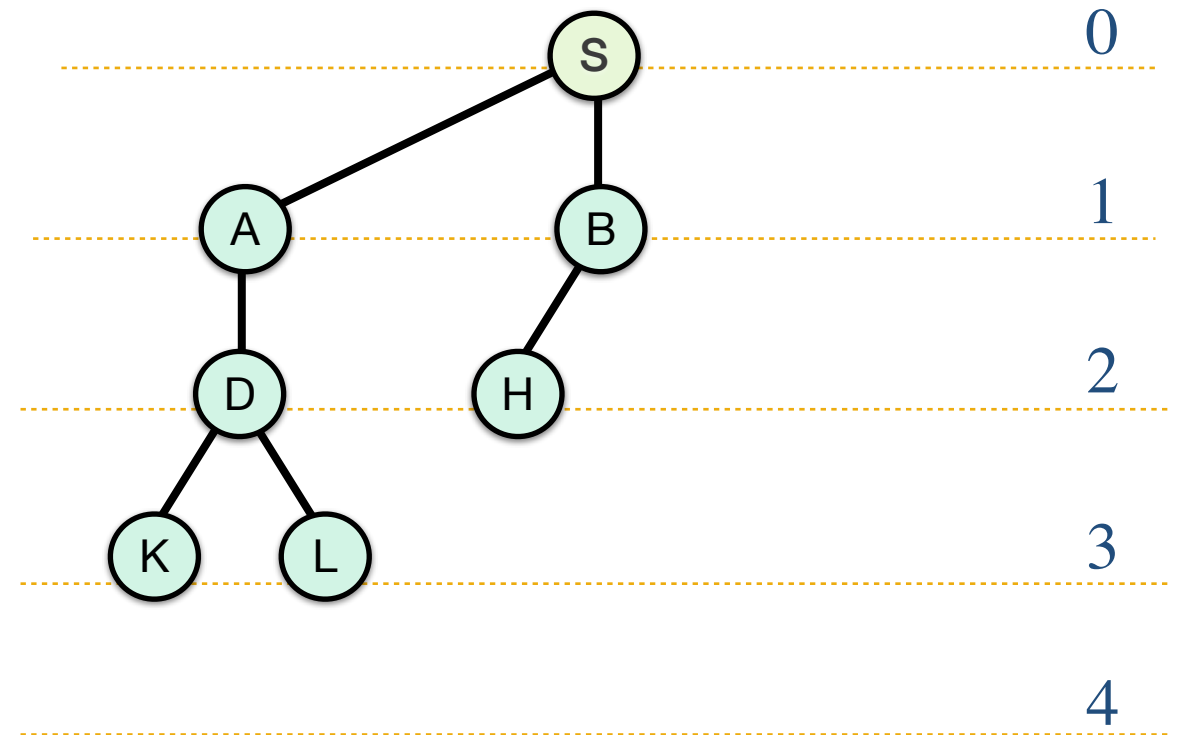
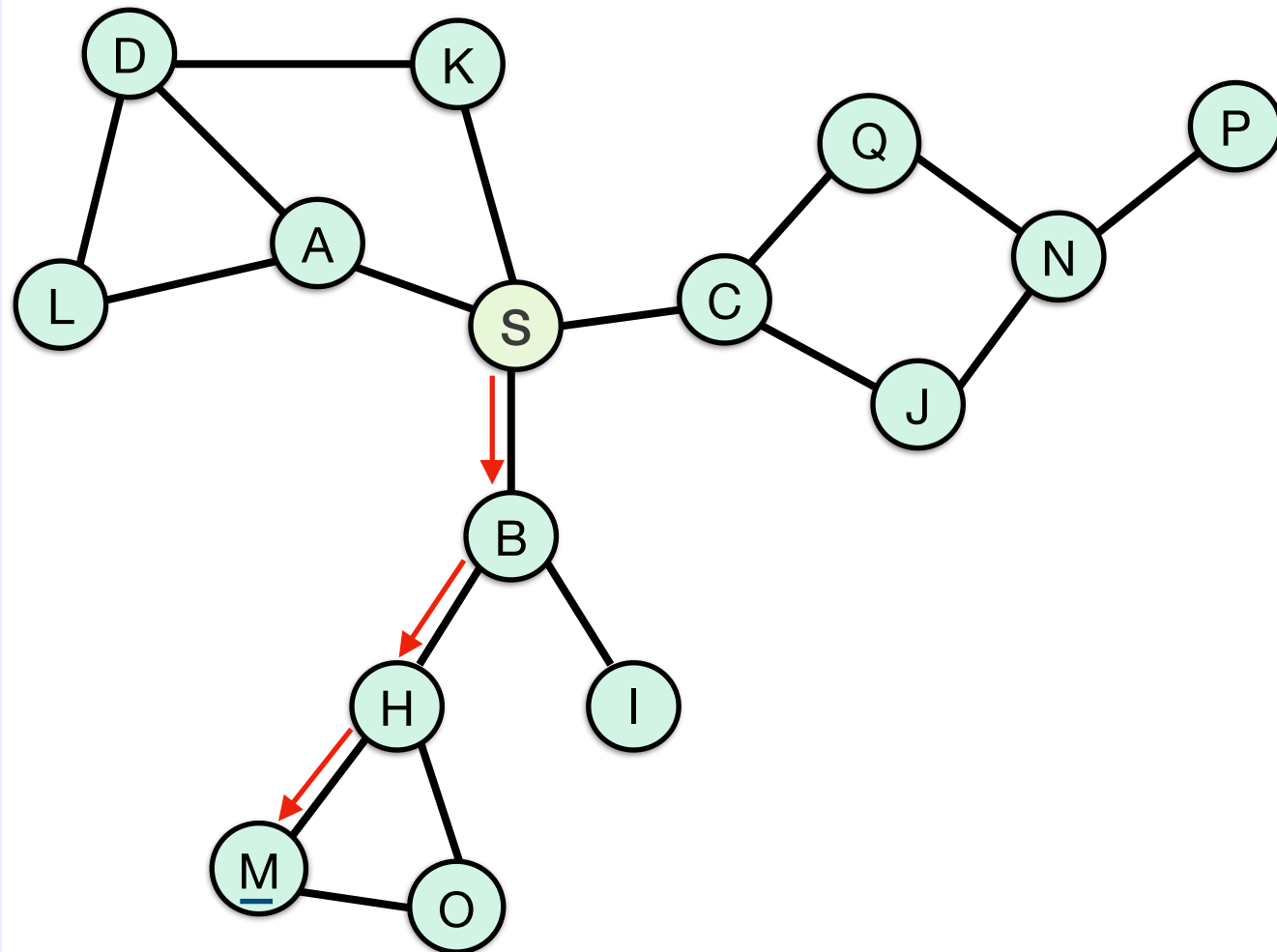
DFS tree

DFS Traversal and DFS Tree



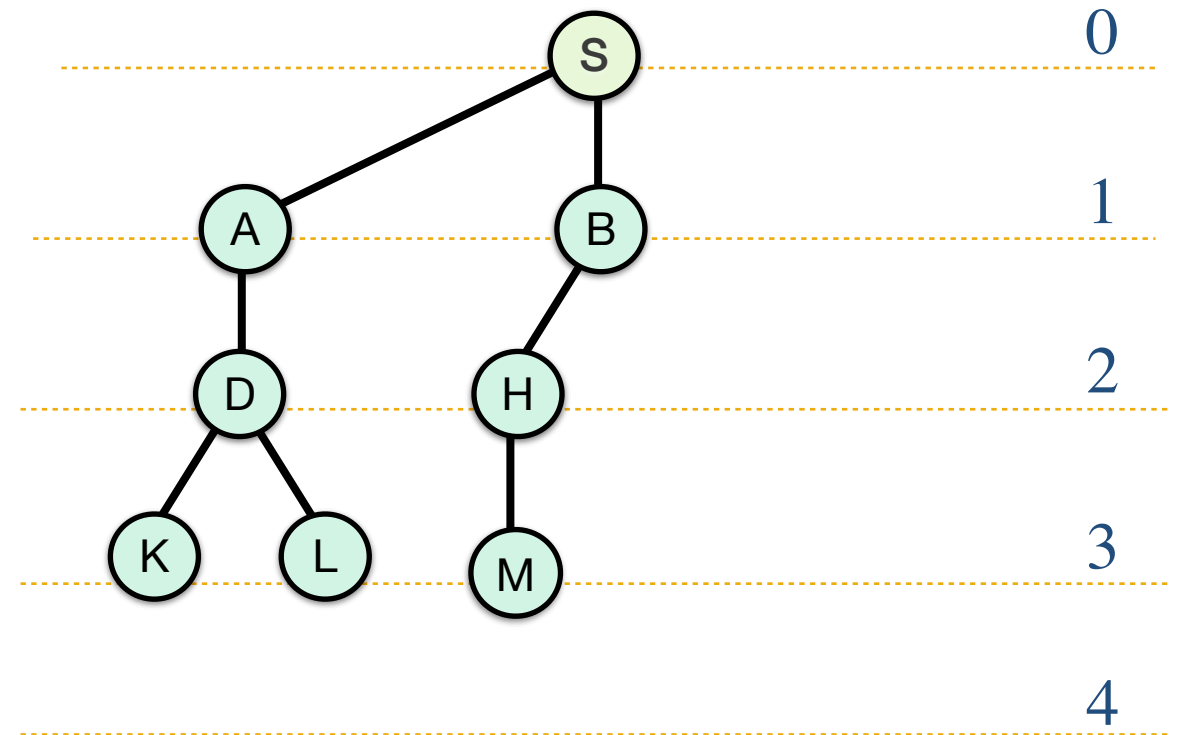
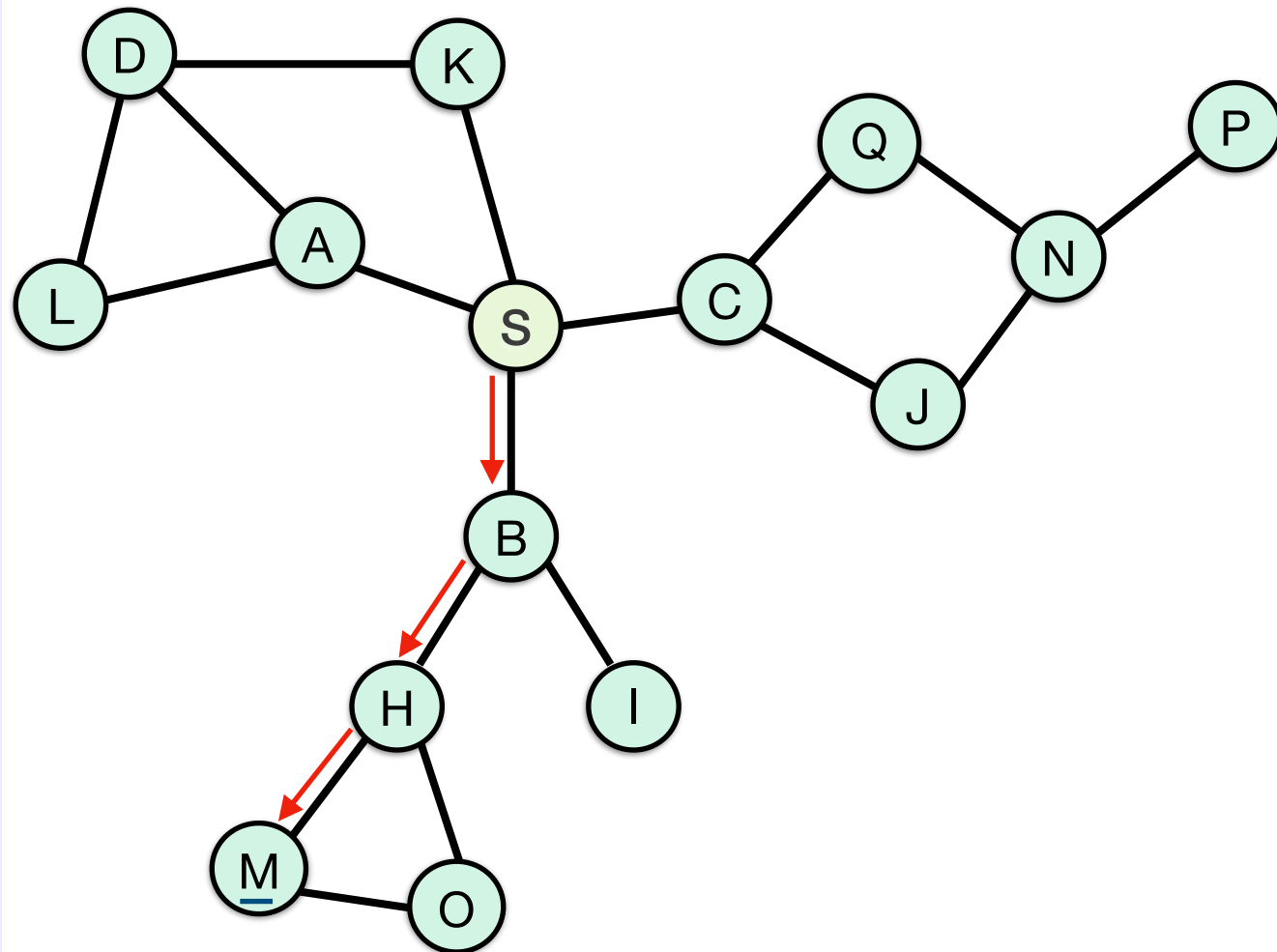
DFS tree

DFS Traversal and DFS Tree



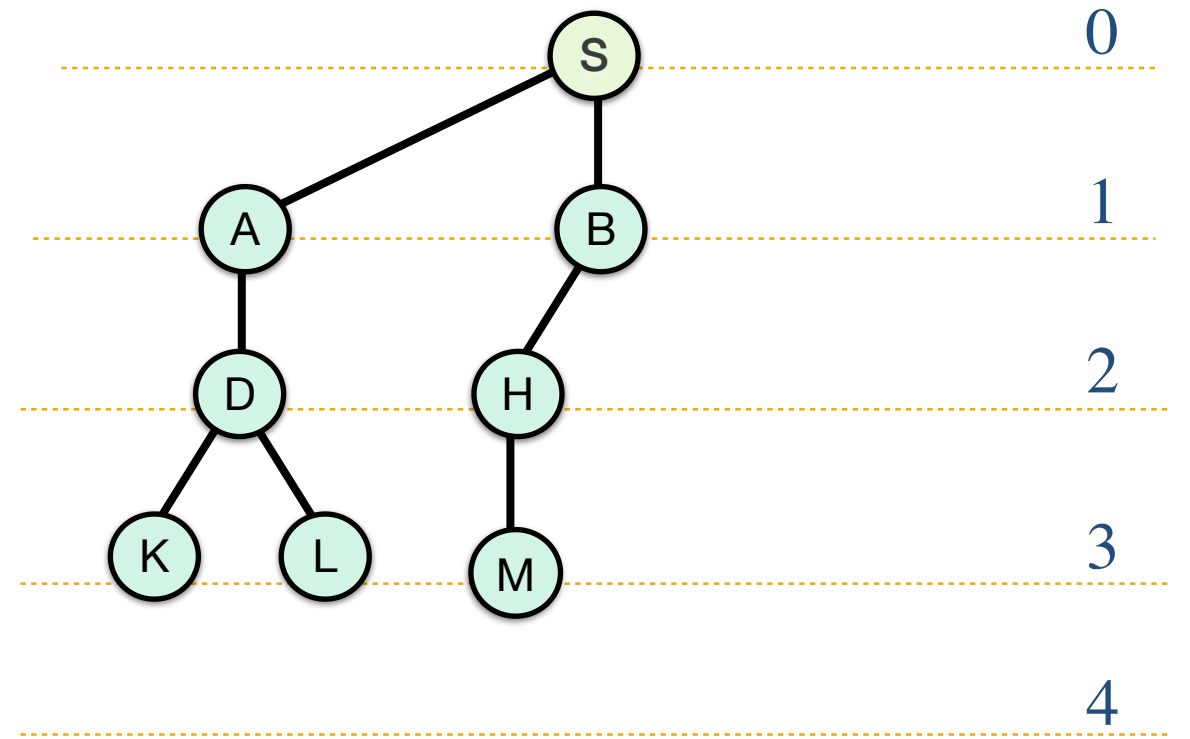
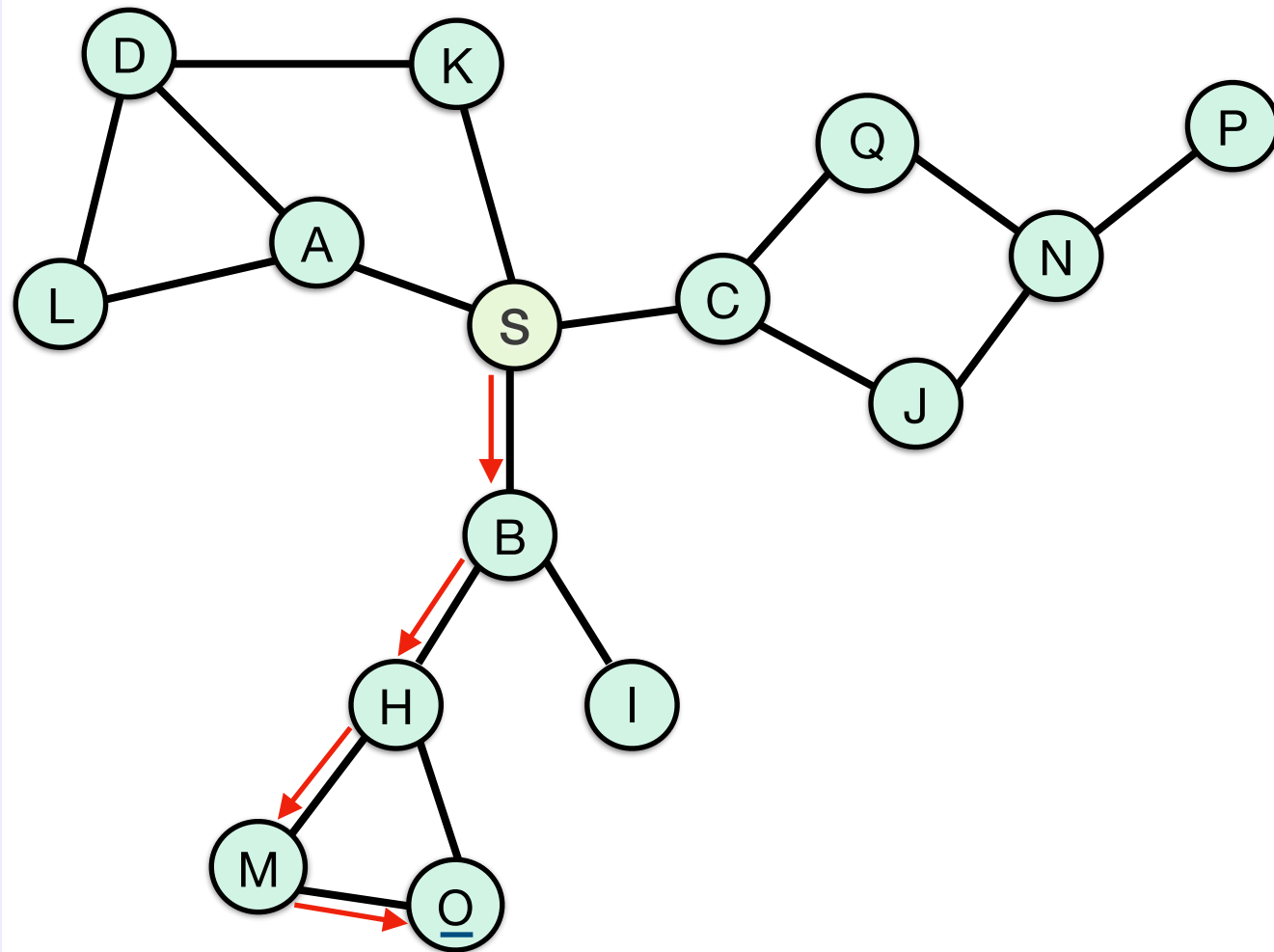
DFS tree

DFS Traversal and DFS Tree



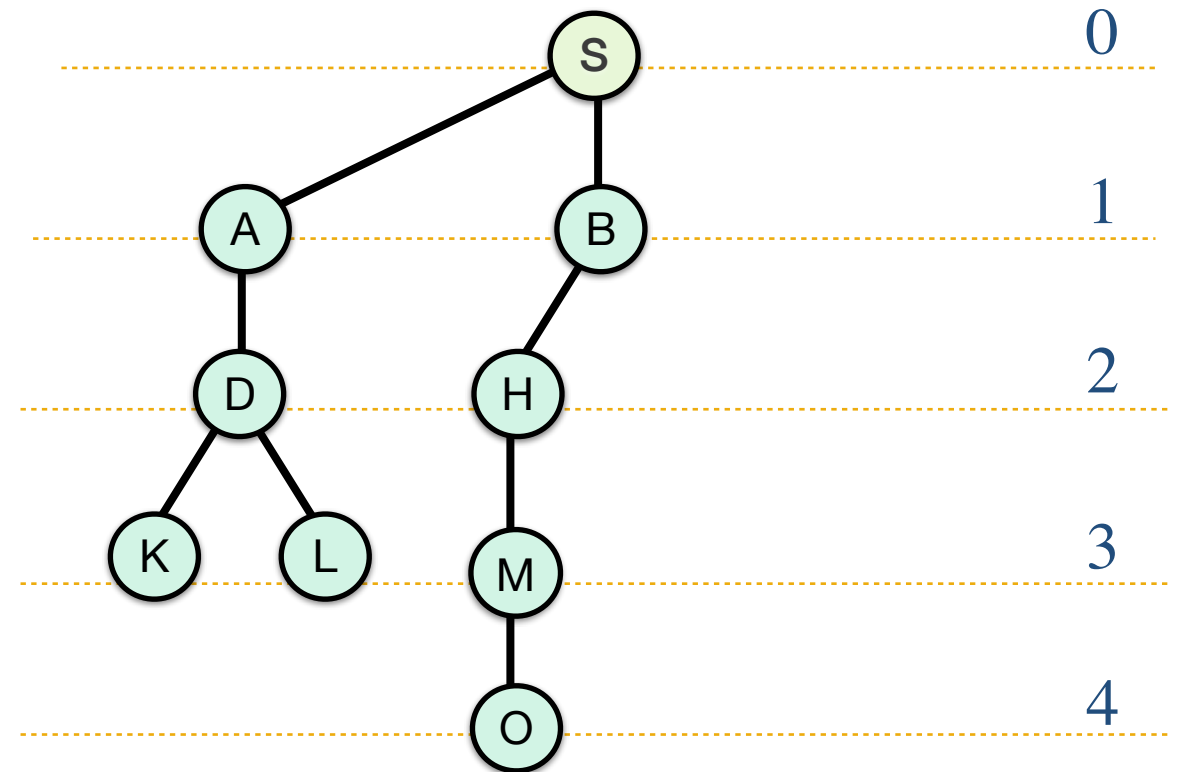
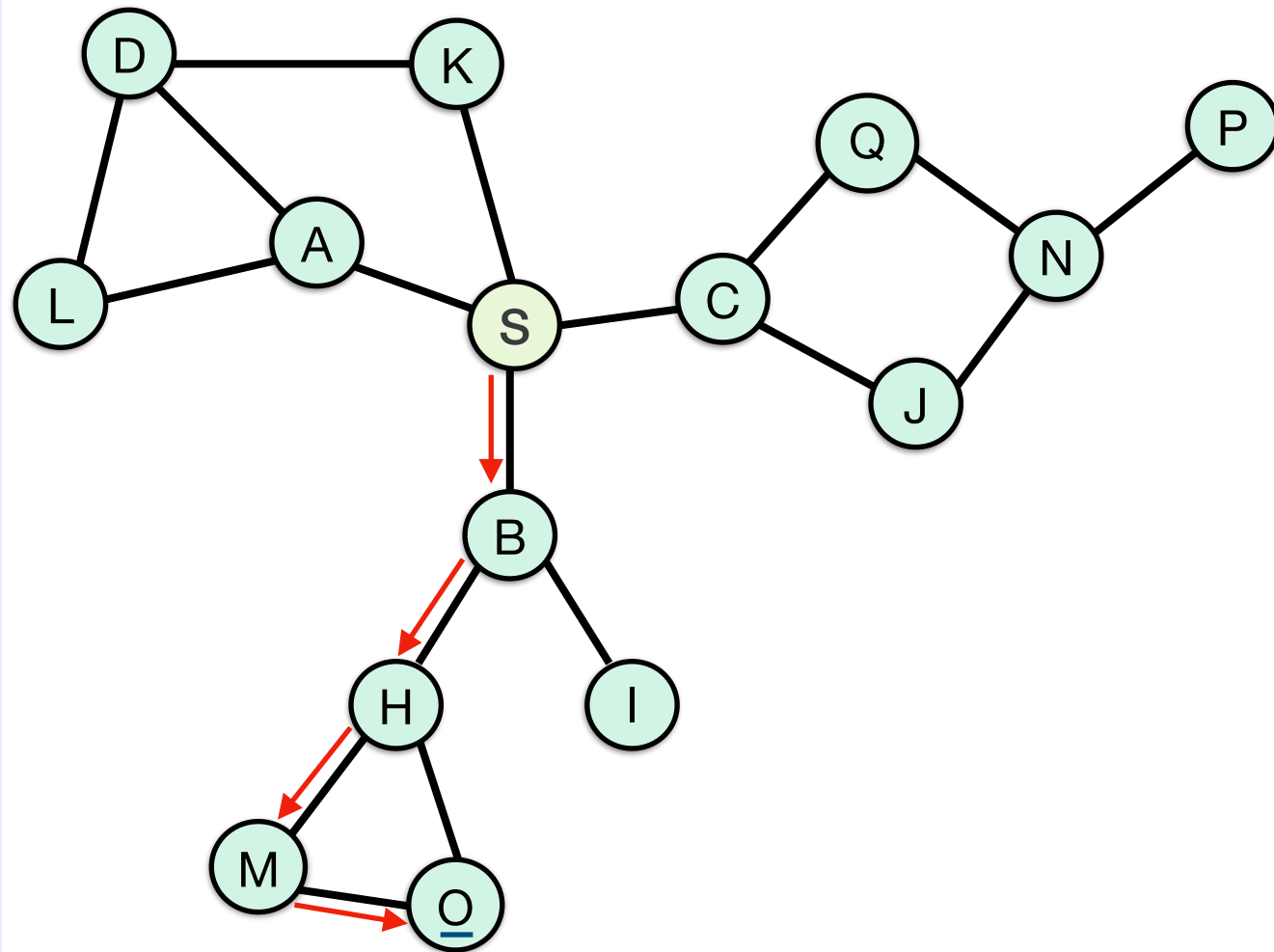
DFS tree

DFS Traversal and DFS Tree



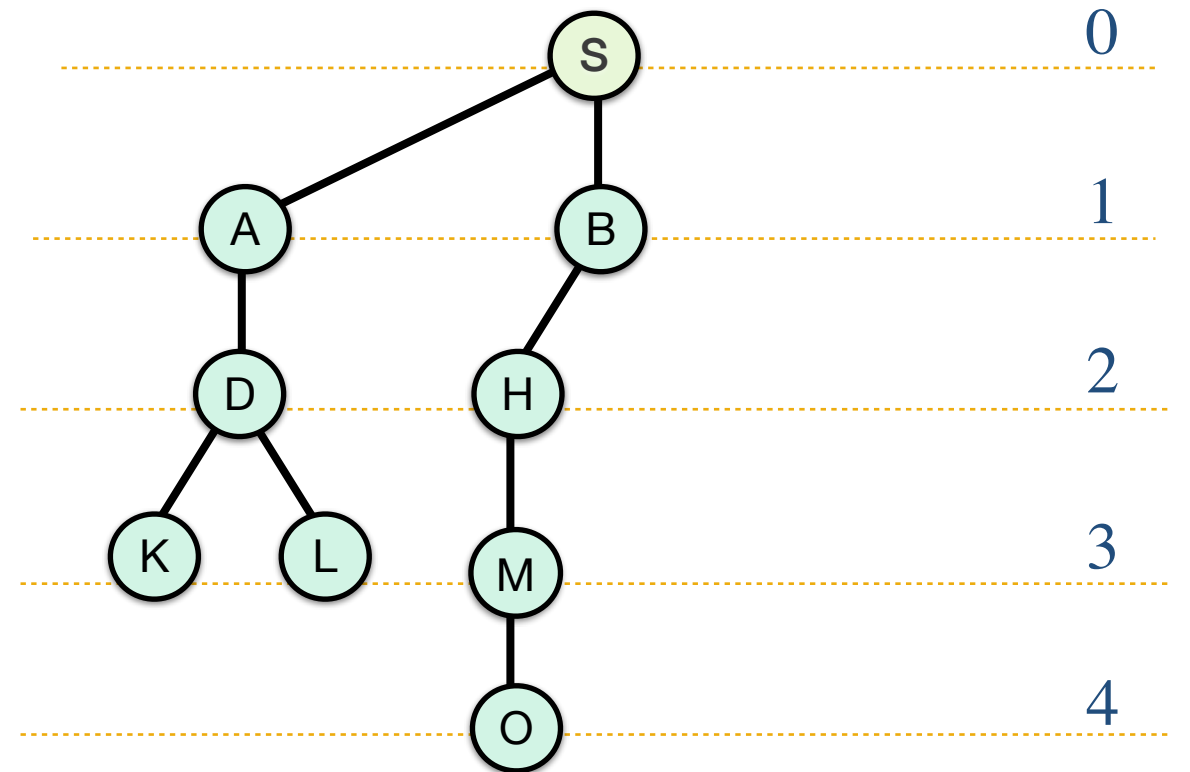
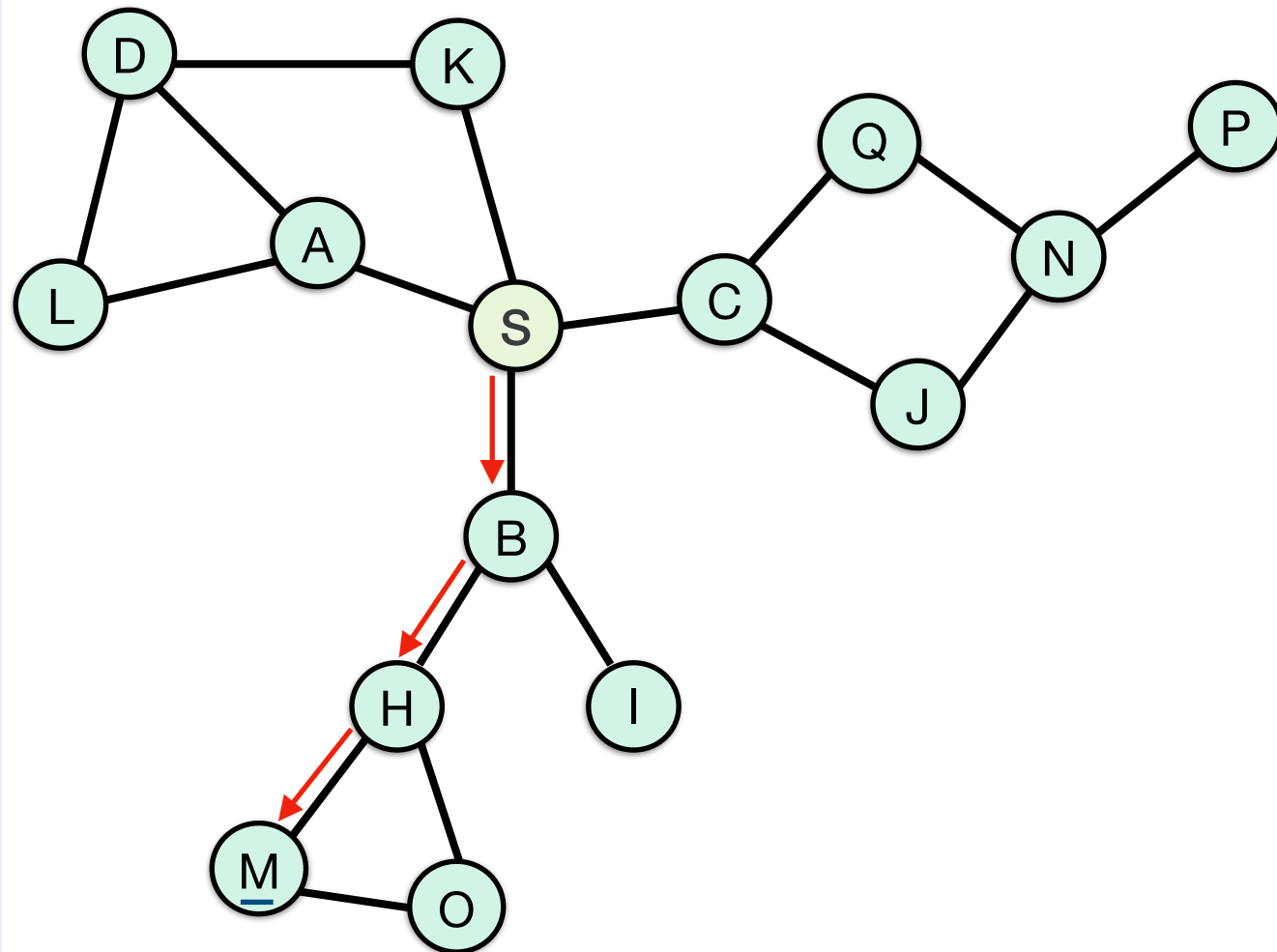
DFS tree

DFS Traversal and DFS Tree



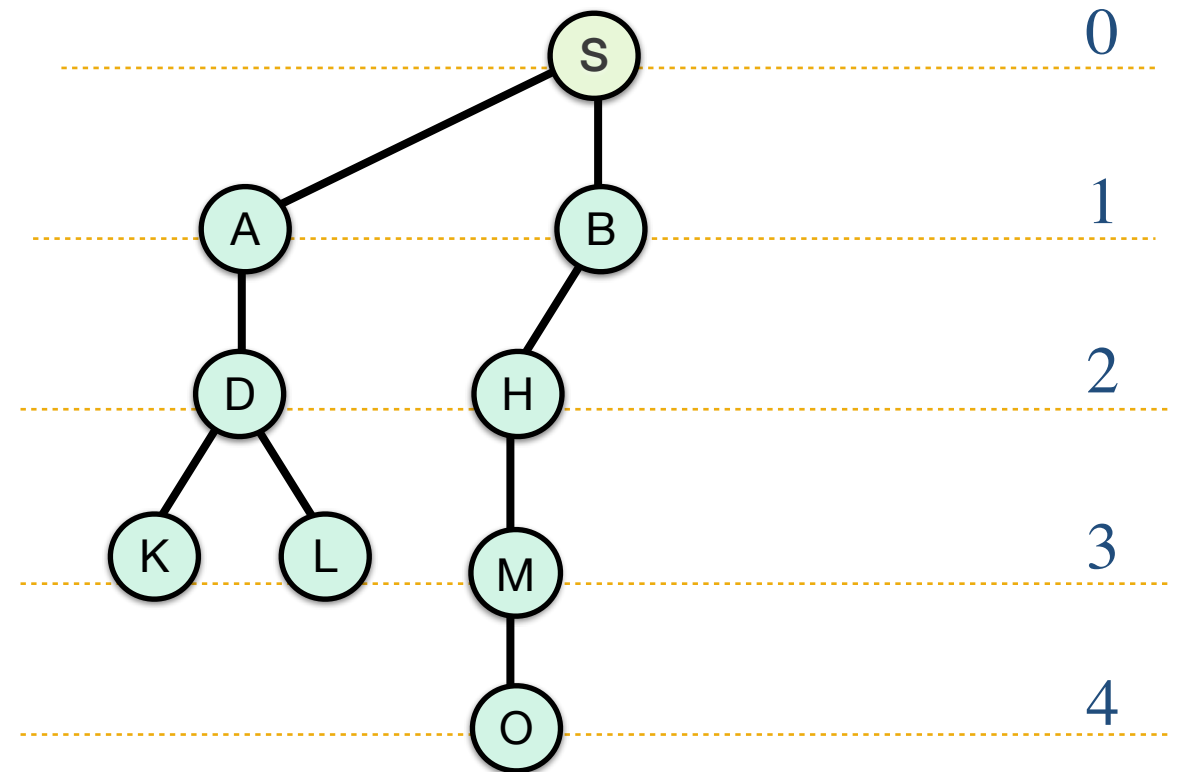
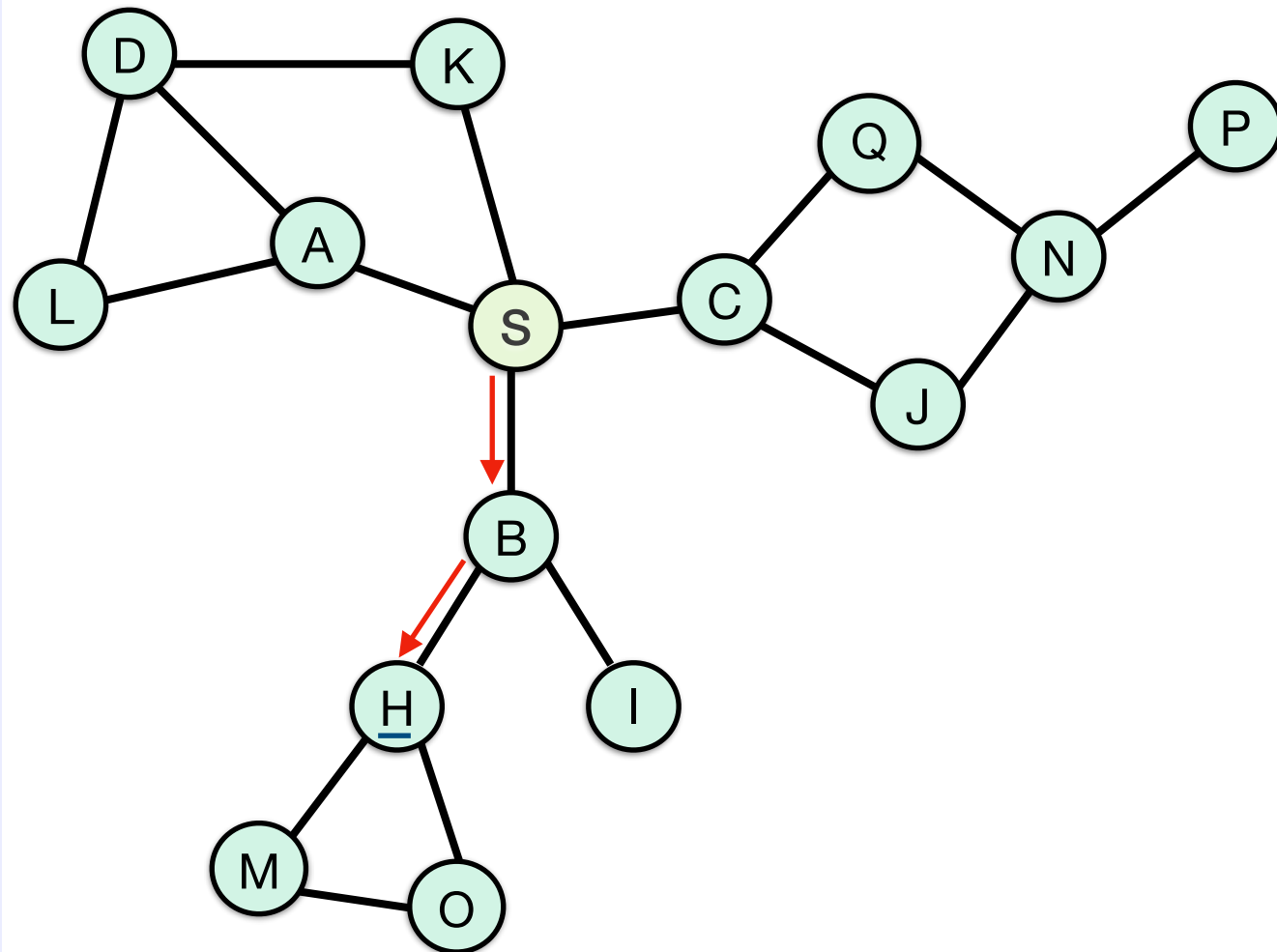
DFS tree

DFS Traversal and DFS Tree



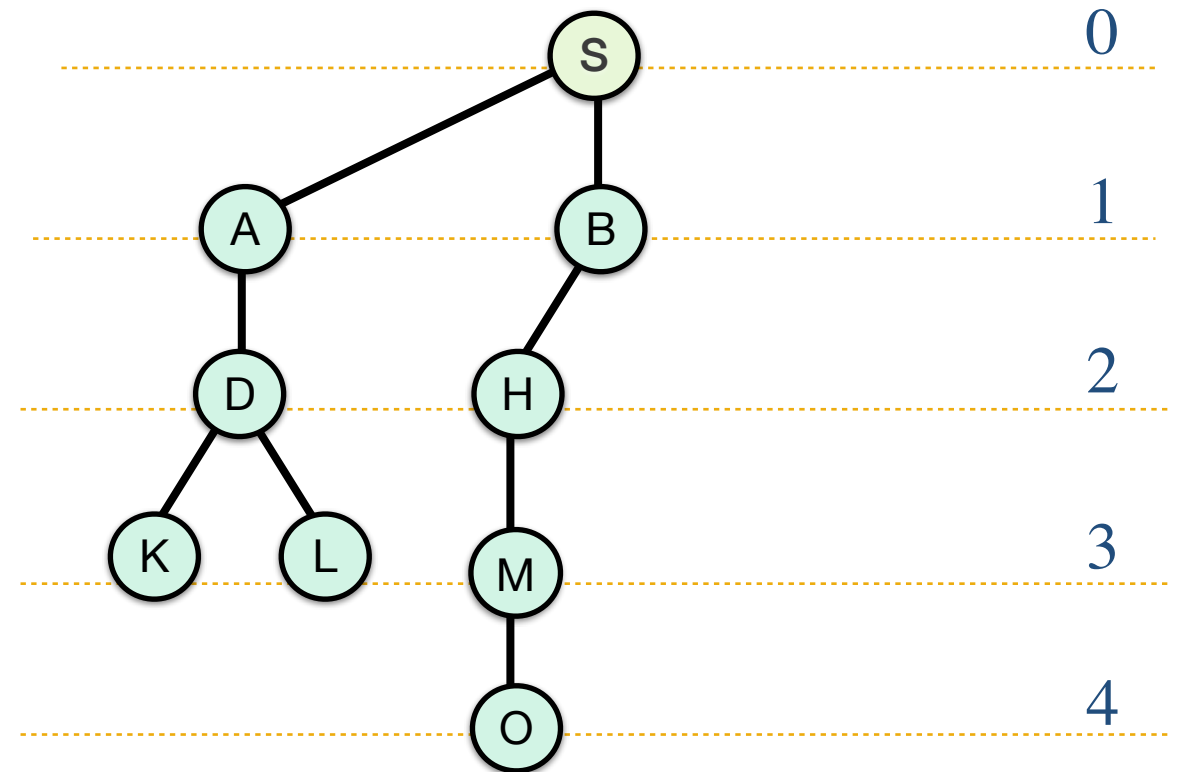
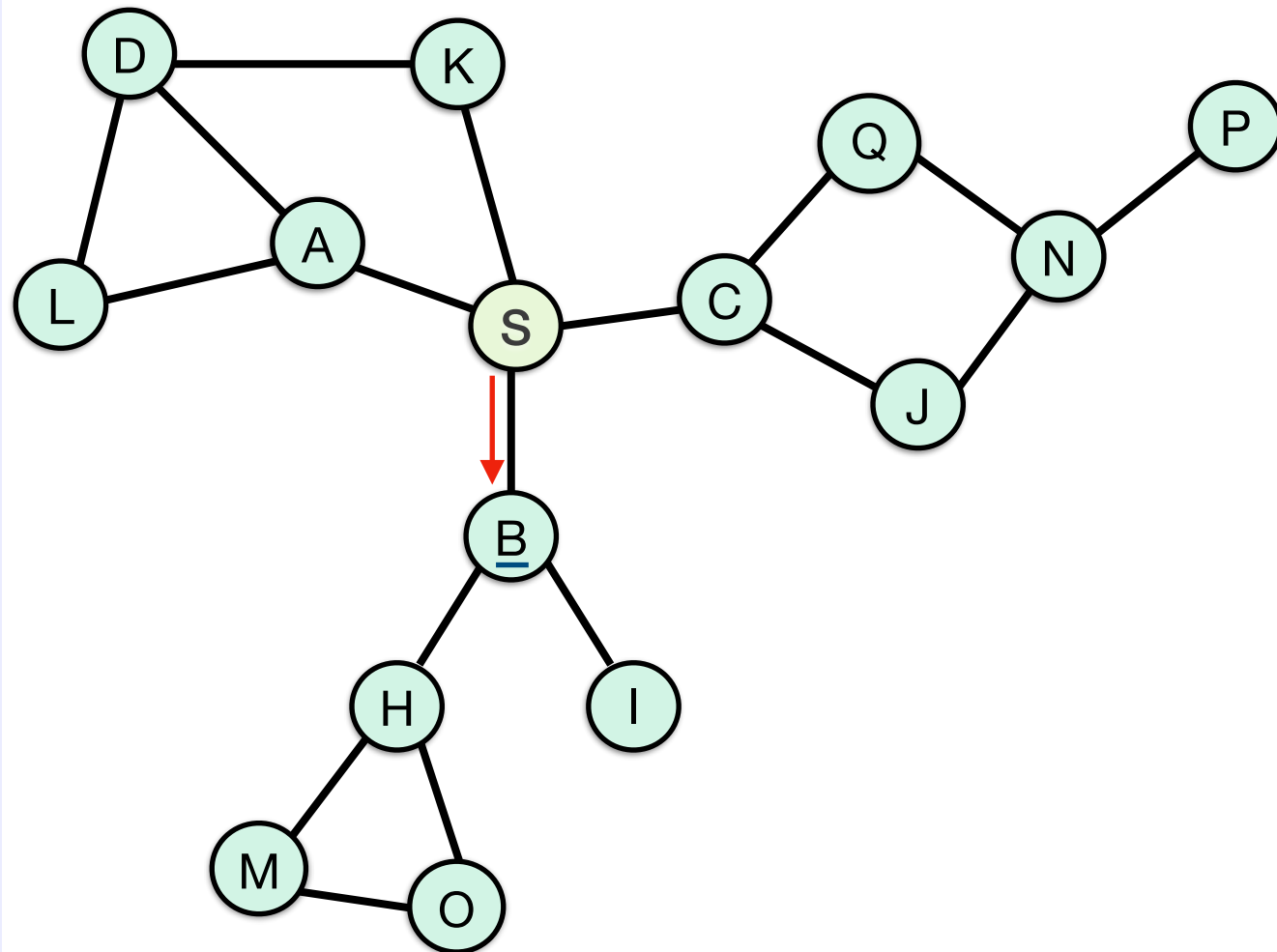
DFS tree

DFS Traversal and DFS Tree



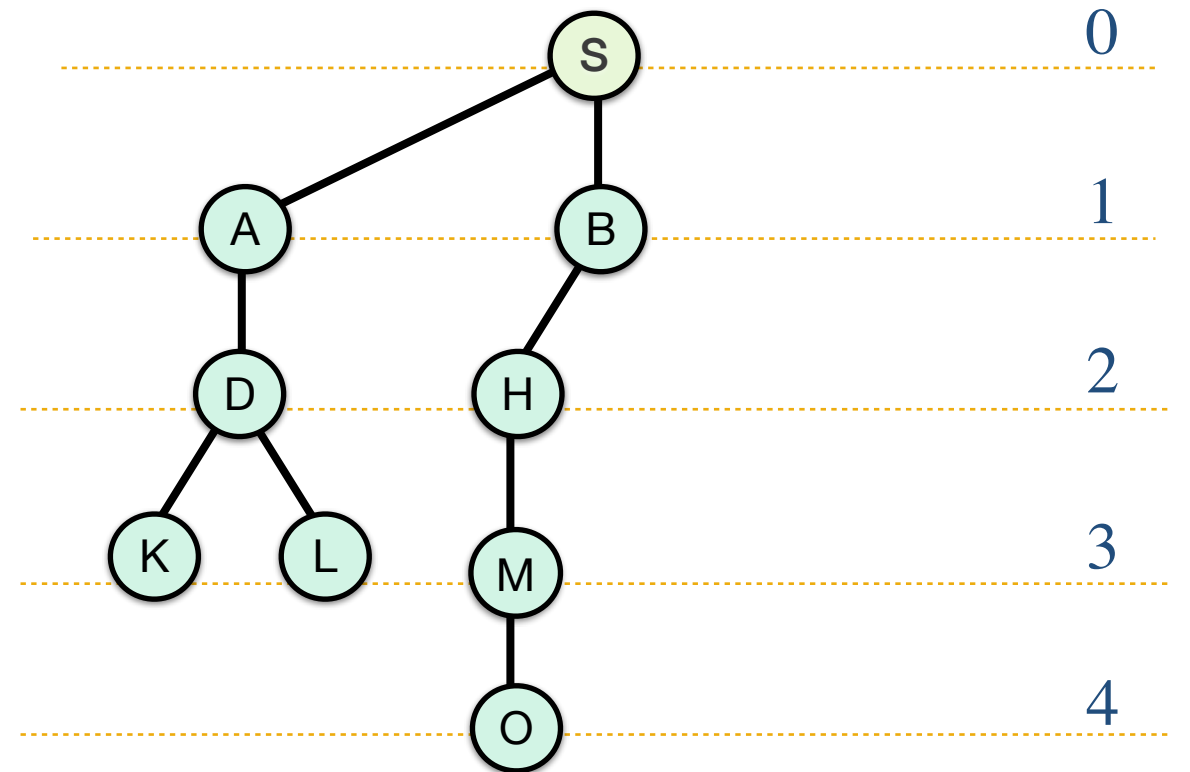
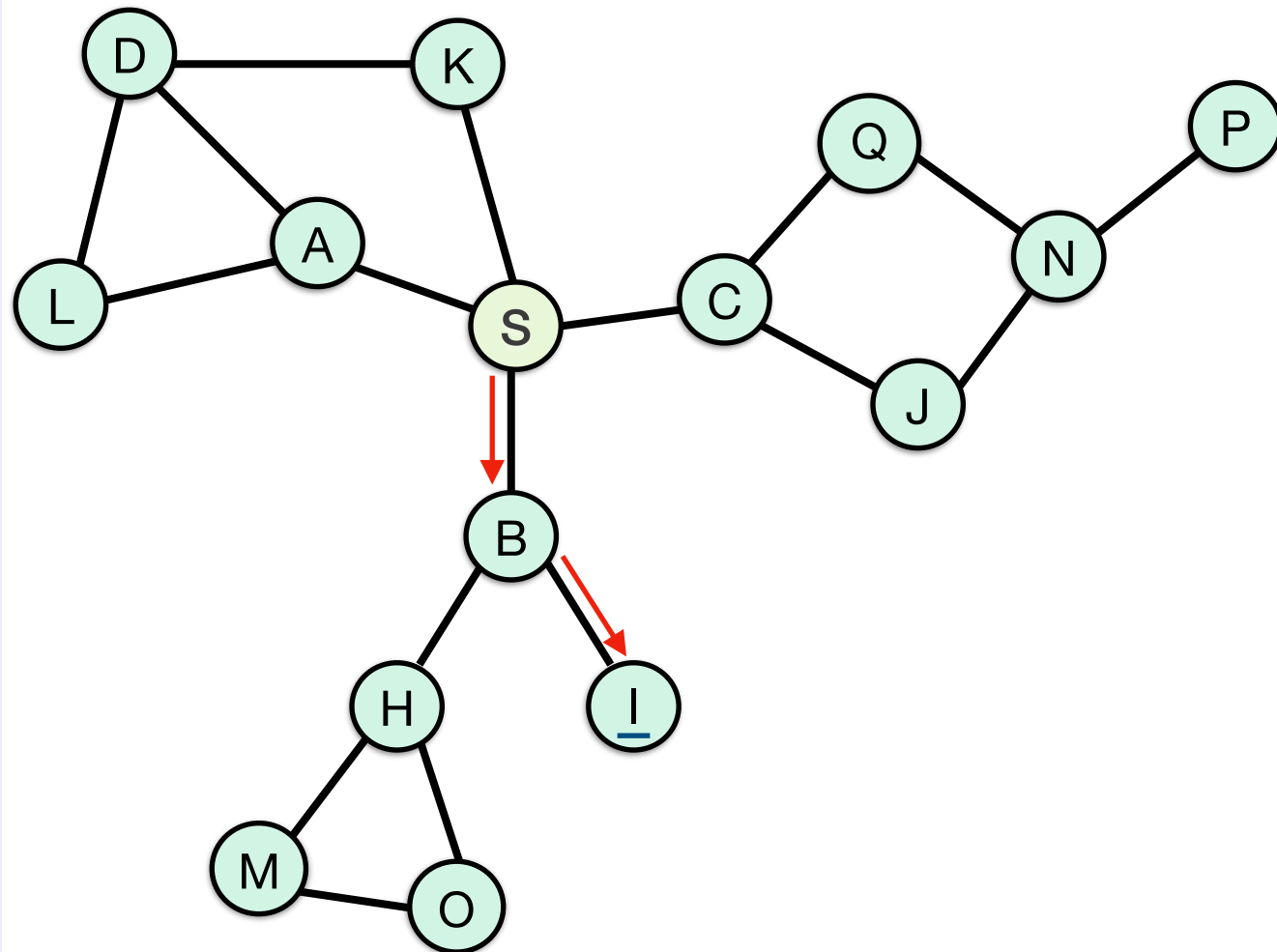
DFS tree

DFS Traversal and DFS Tree



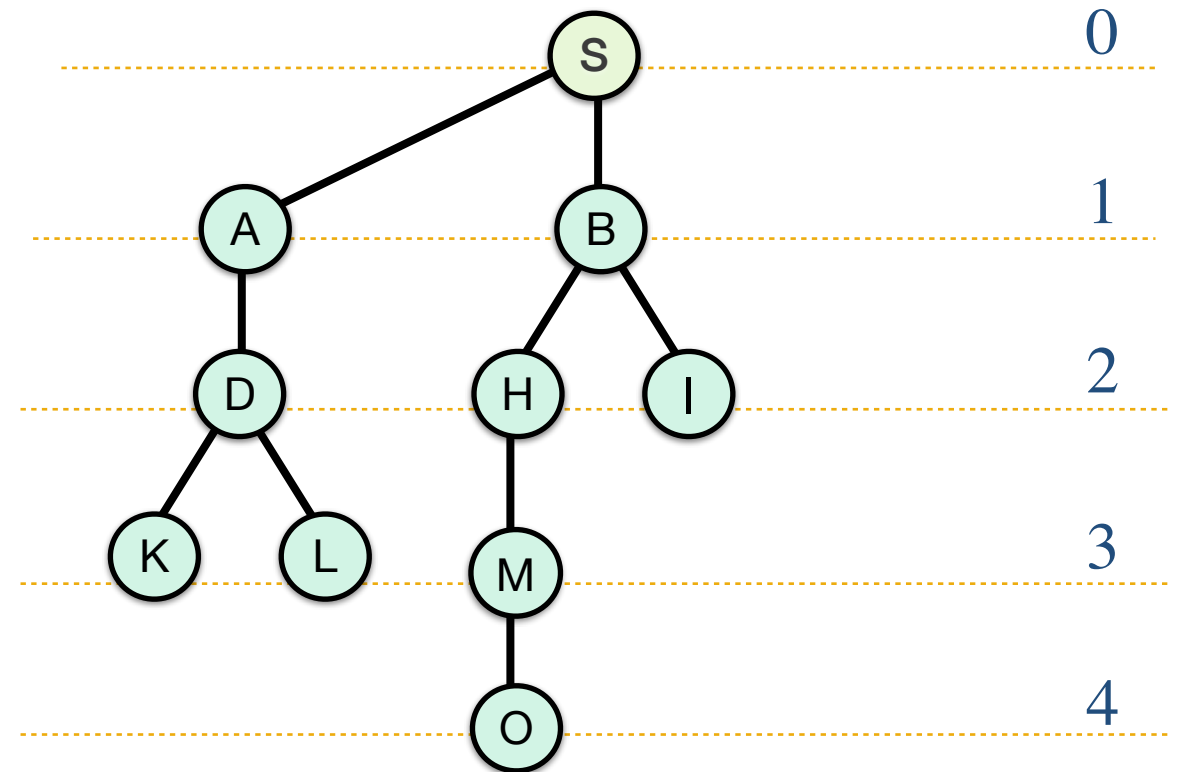
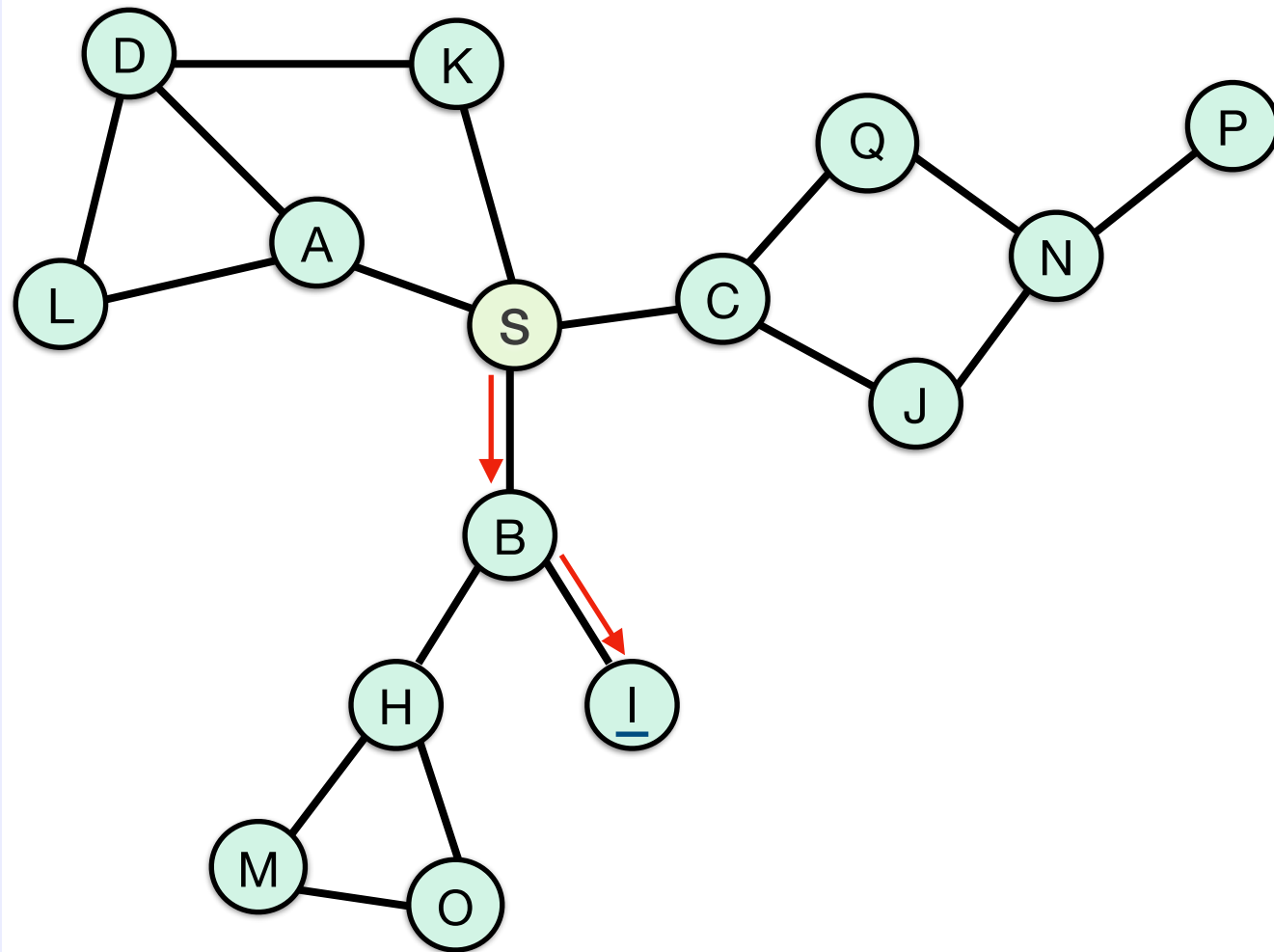
DFS tree

DFS Traversal and DFS Tree



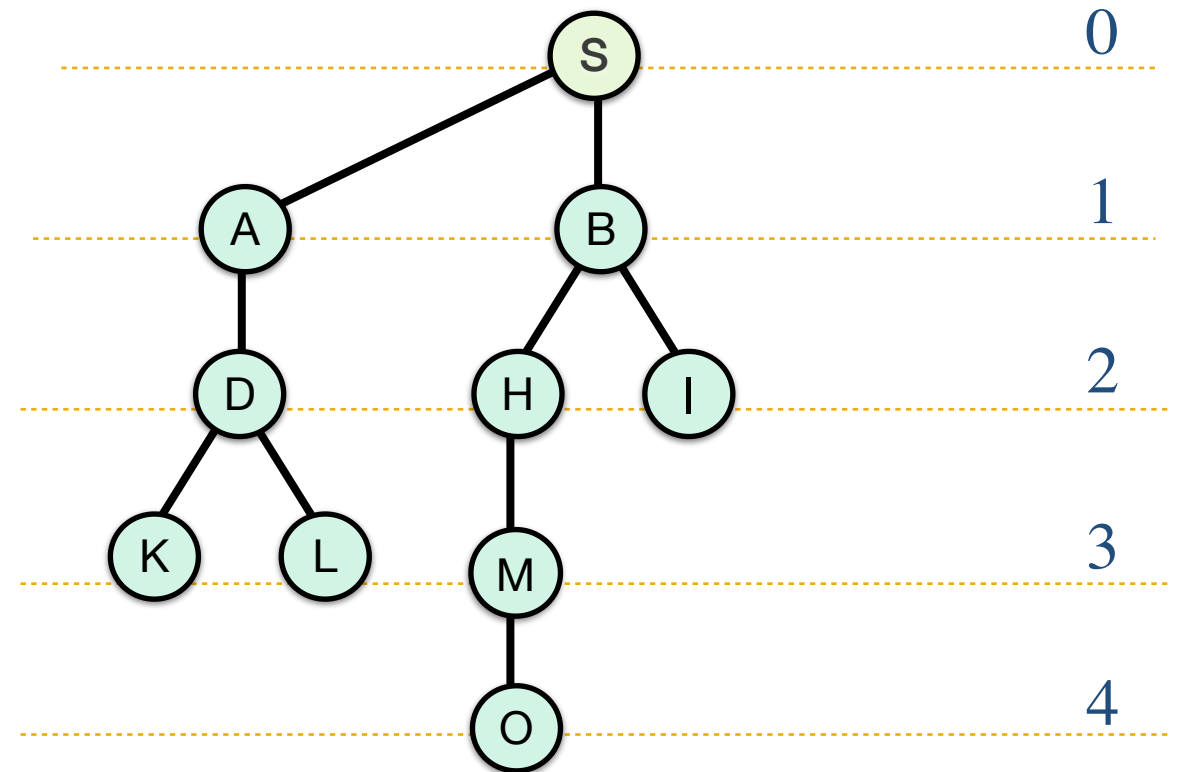
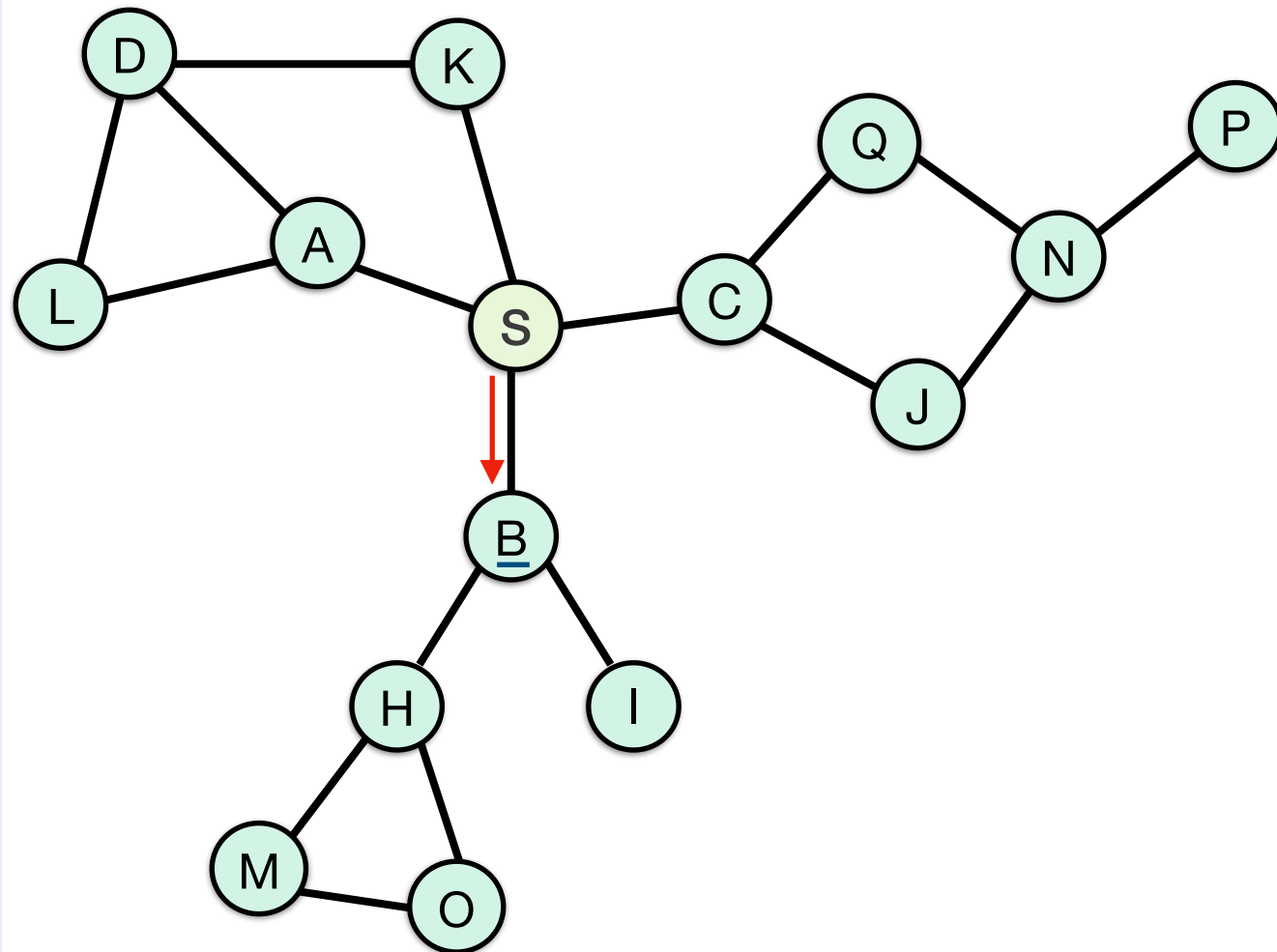
DFS tree

DFS Traversal and DFS Tree



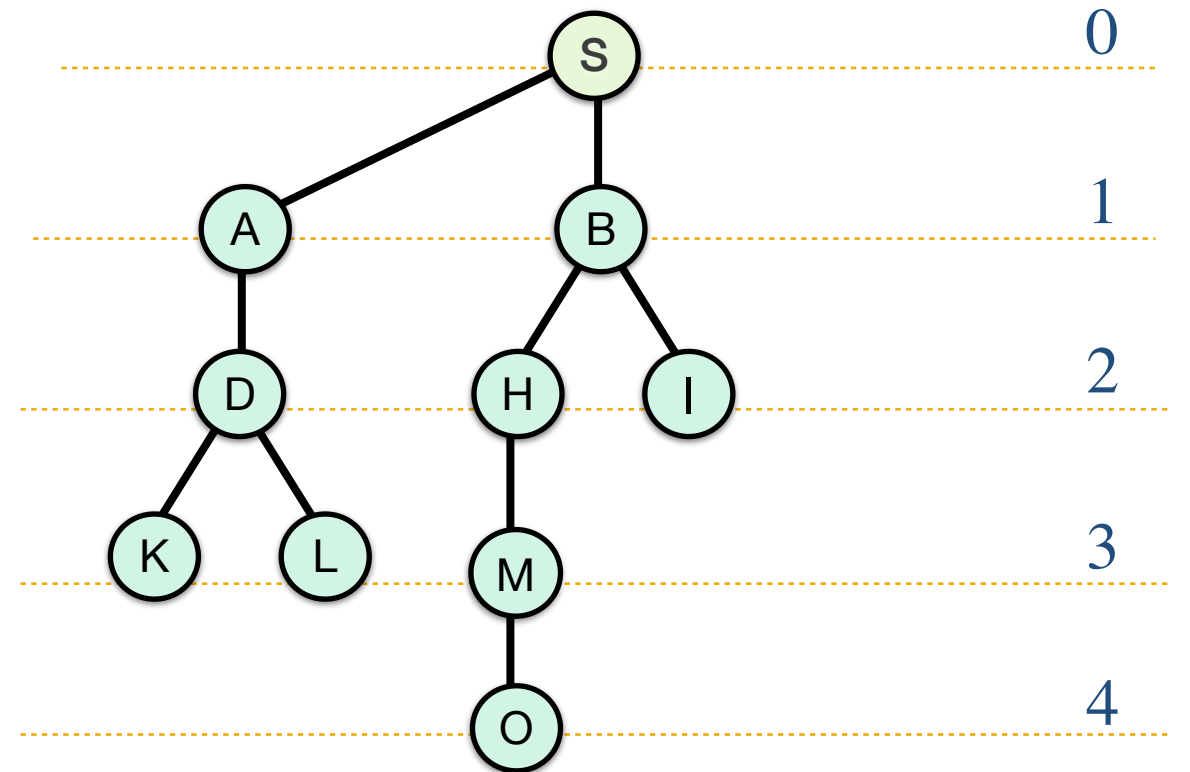
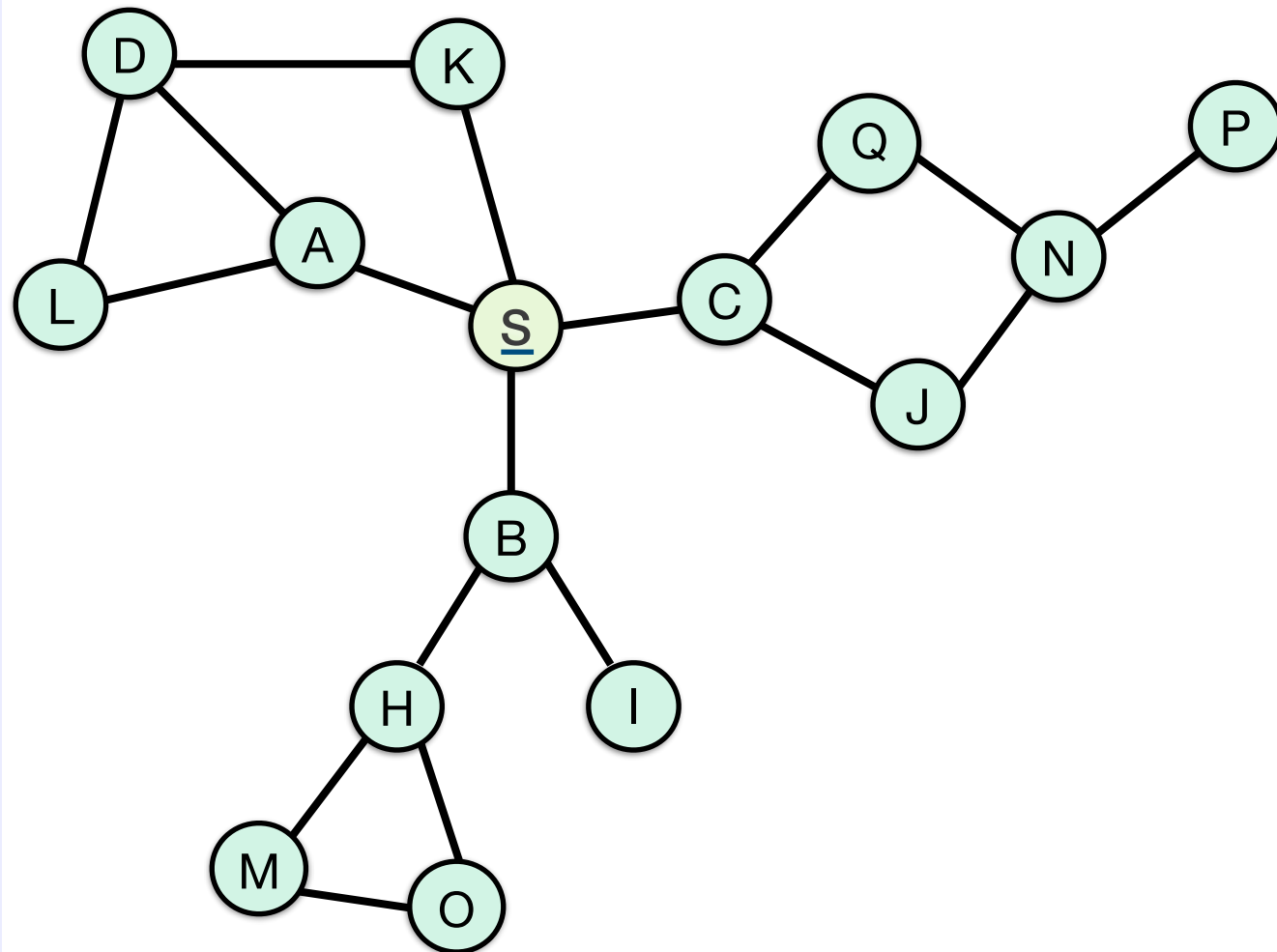
DFS tree

DFS Traversal and DFS Tree



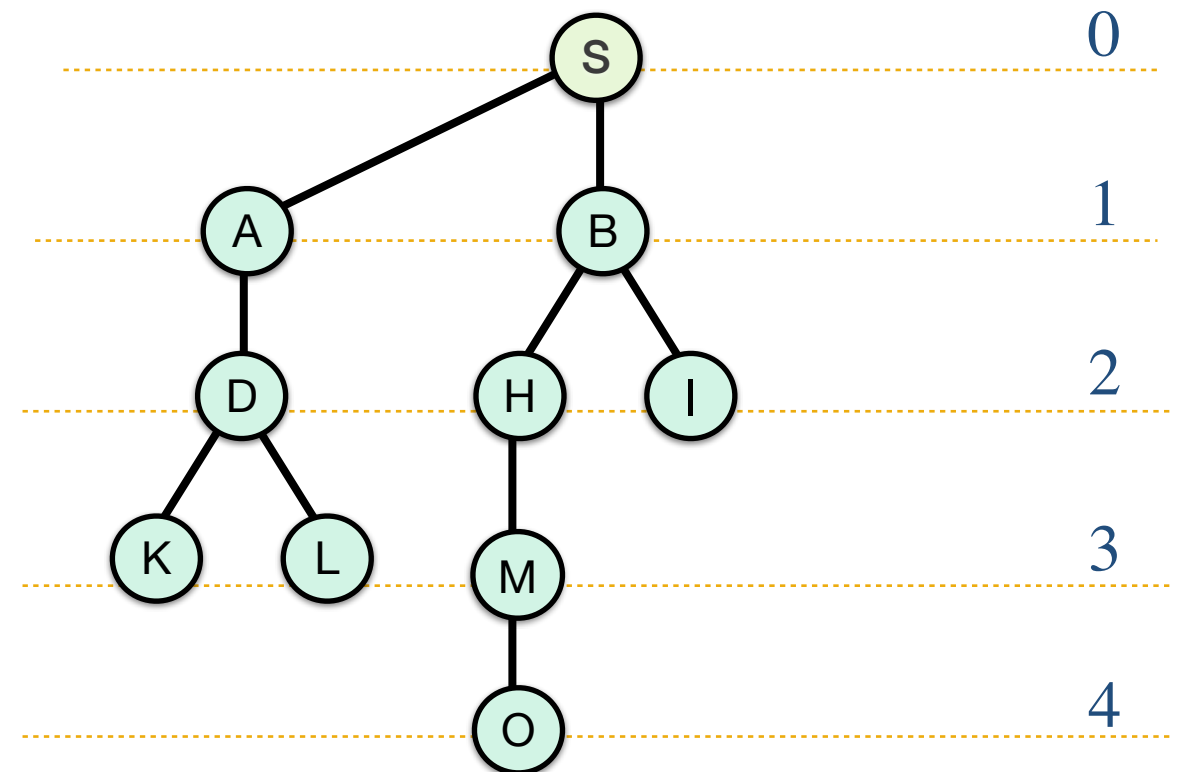
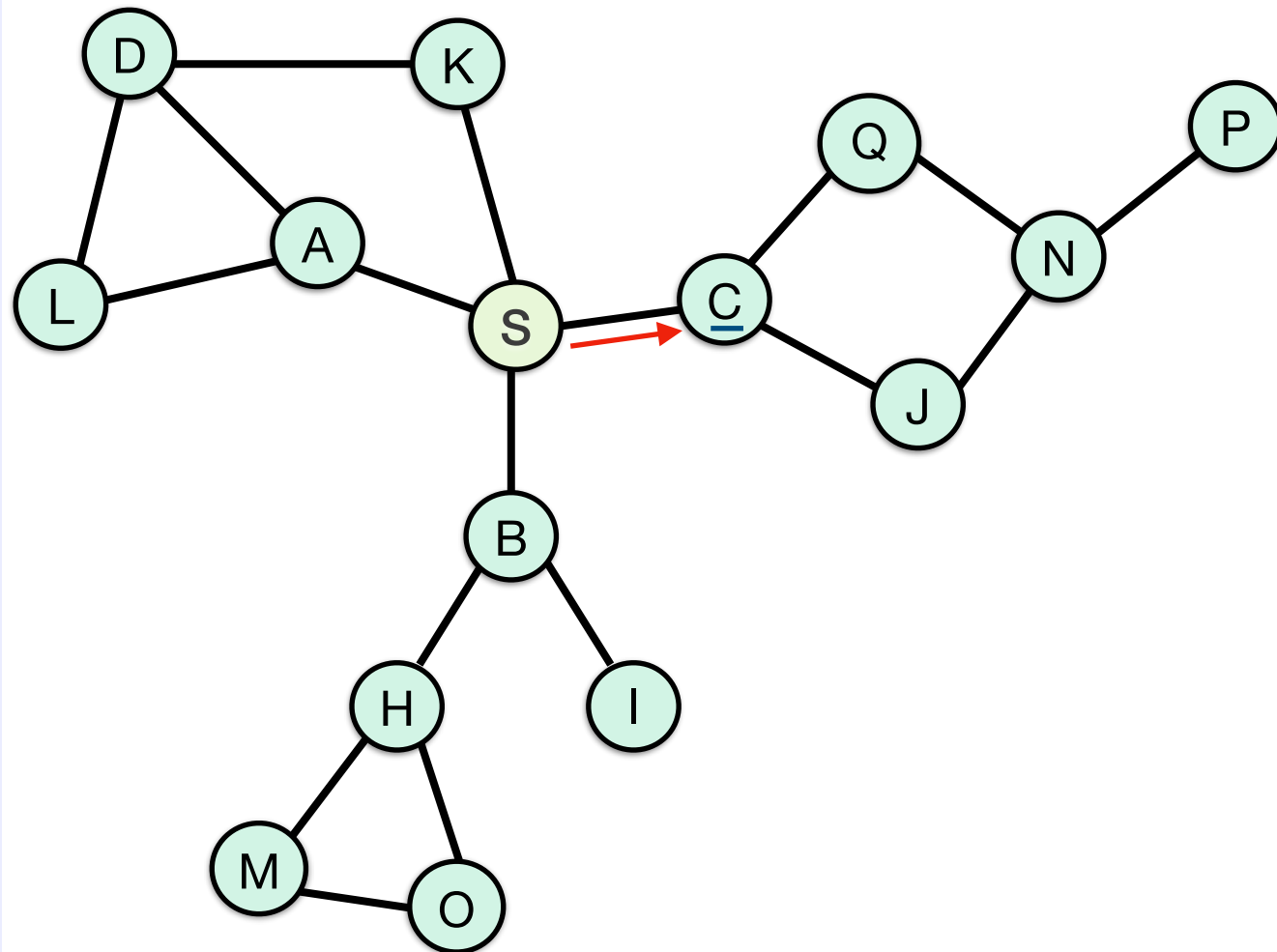
DFS tree

DFS Traversal and DFS Tree



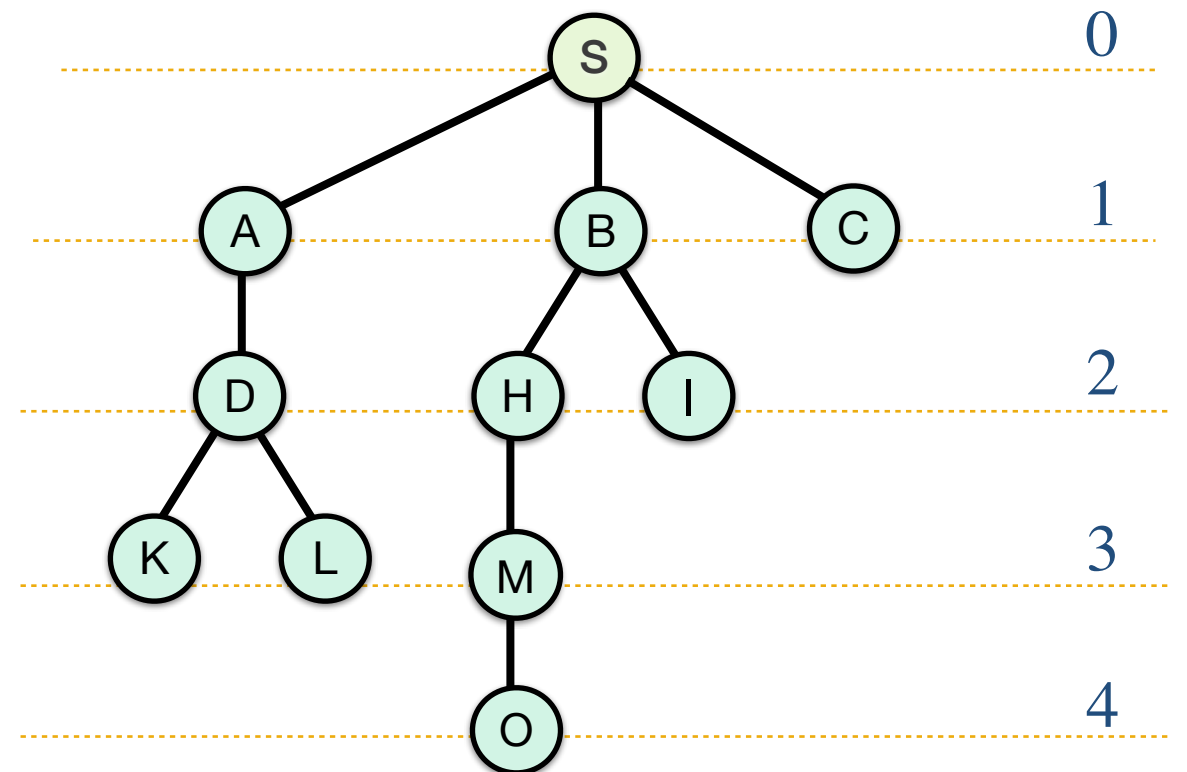
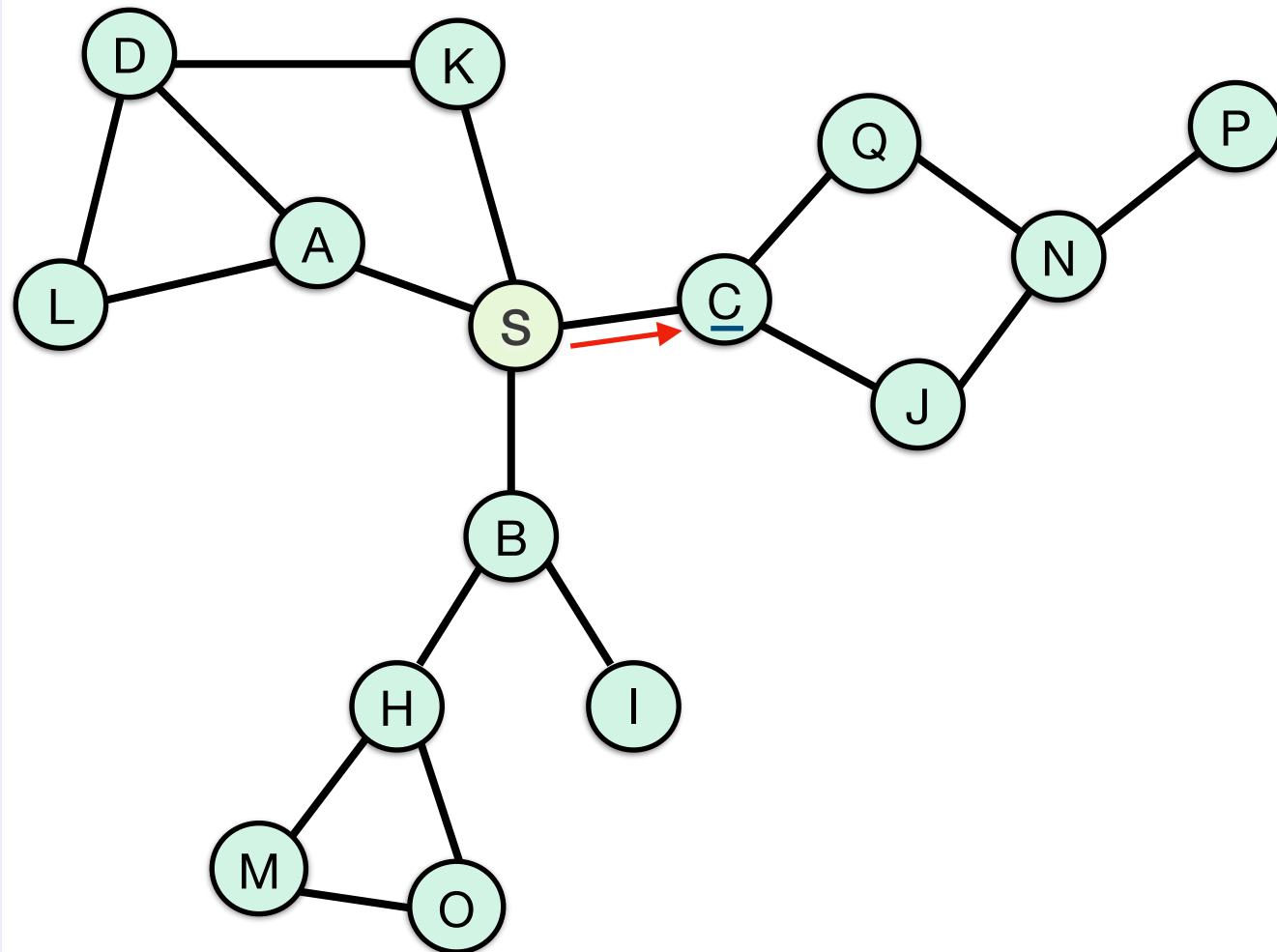
DFS tree

DFS Traversal and DFS Tree



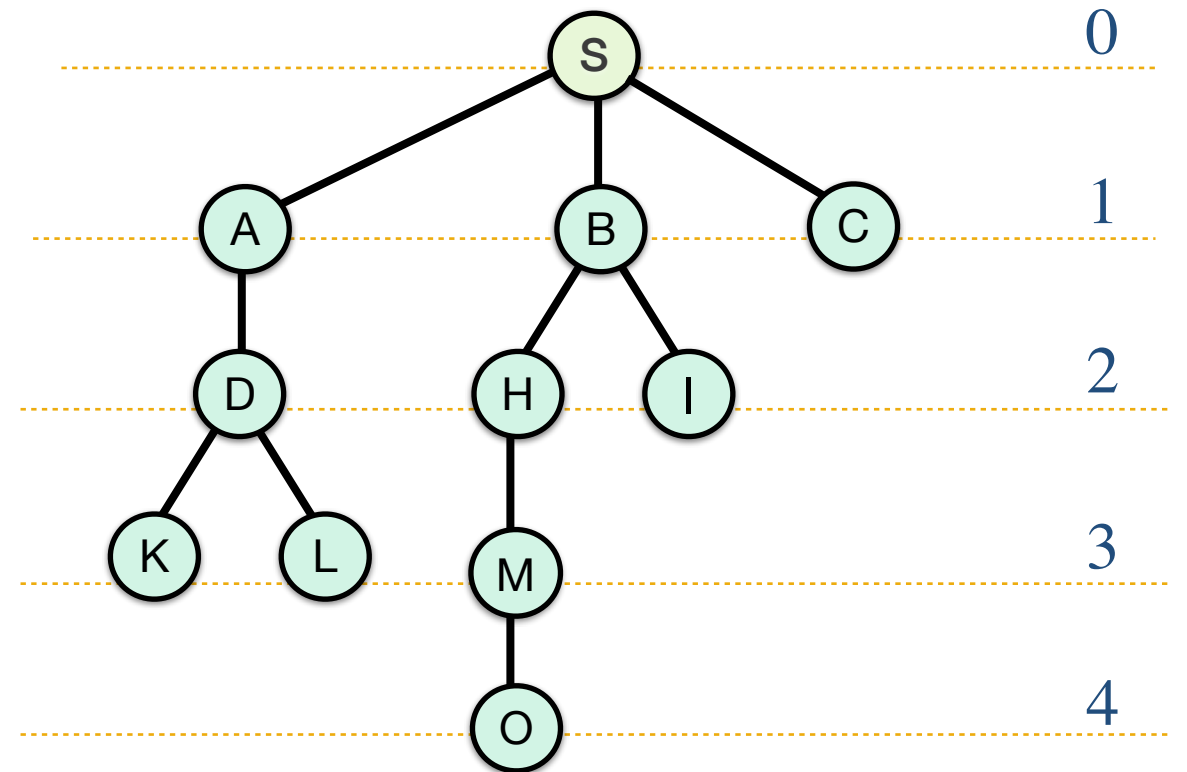
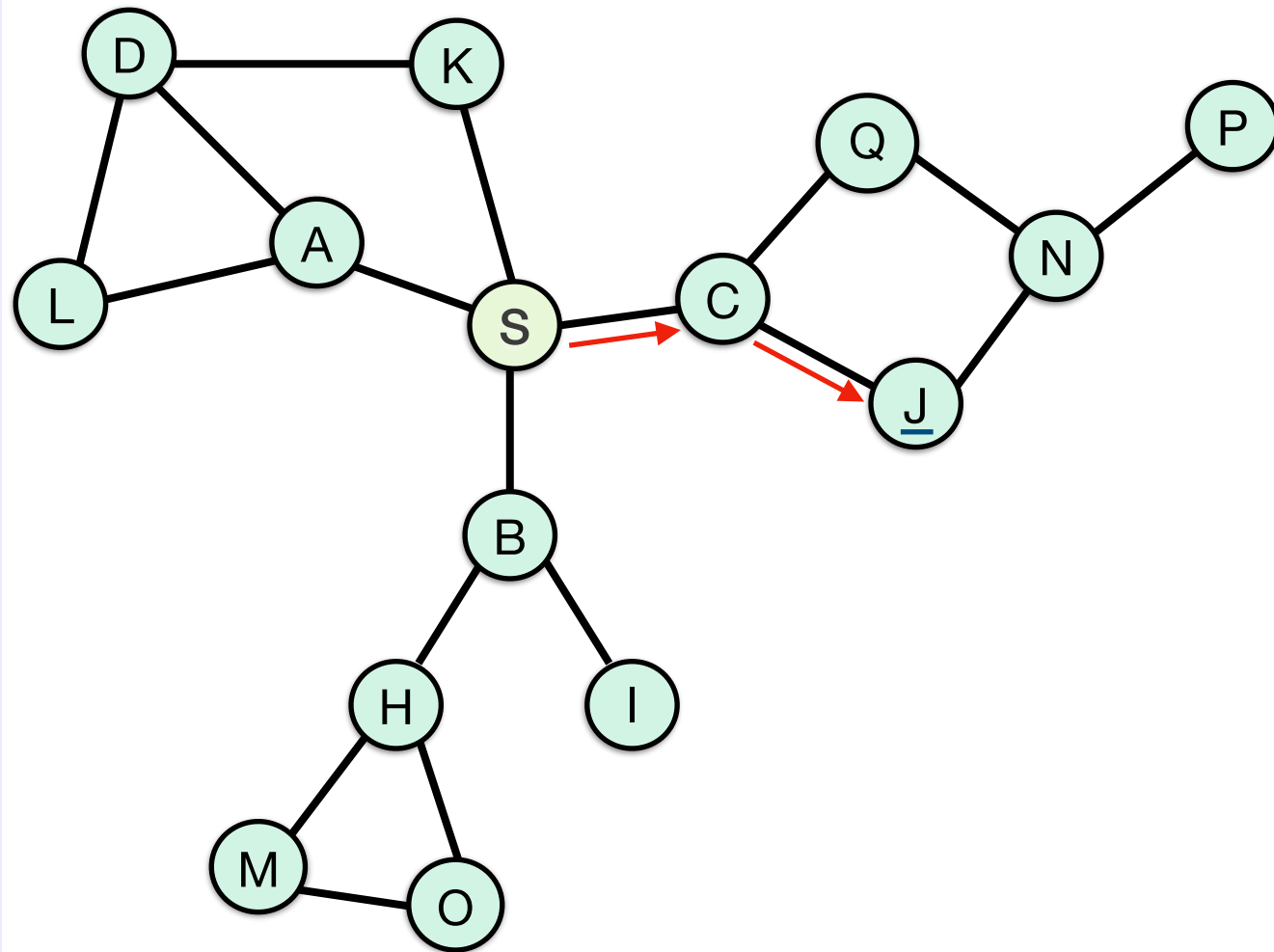
DFS tree

DFS Traversal and DFS Tree



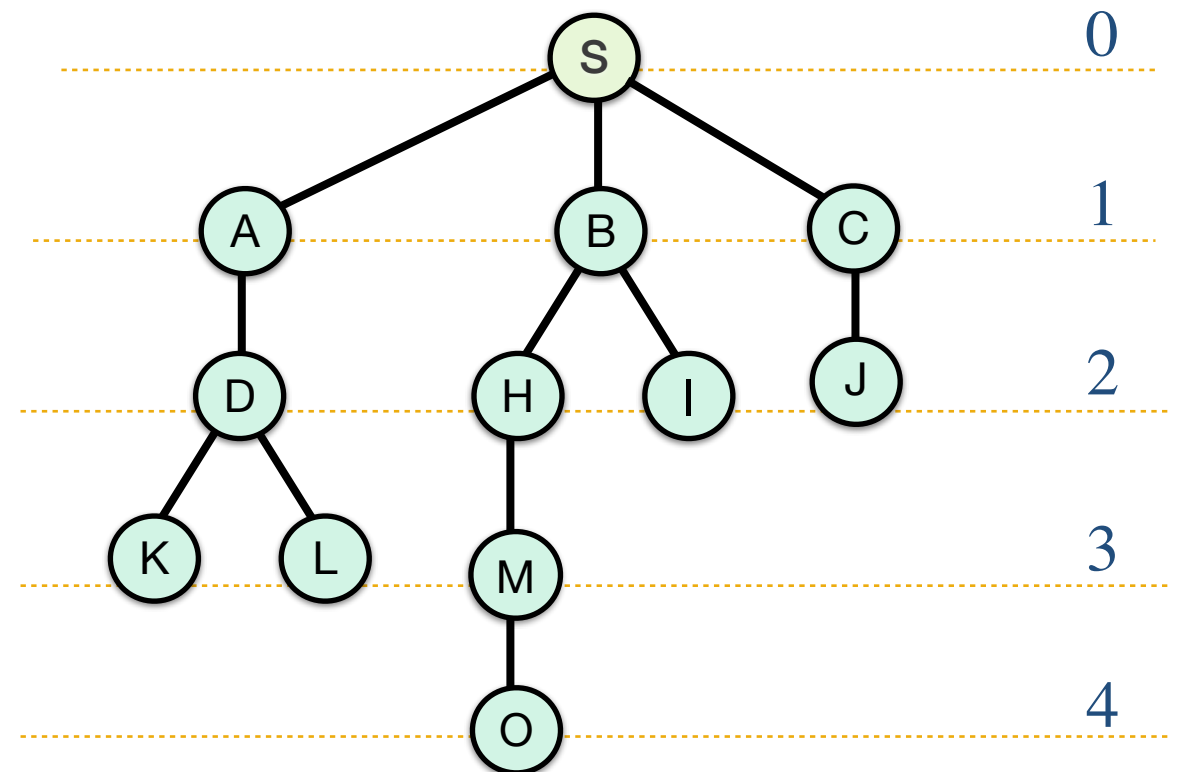
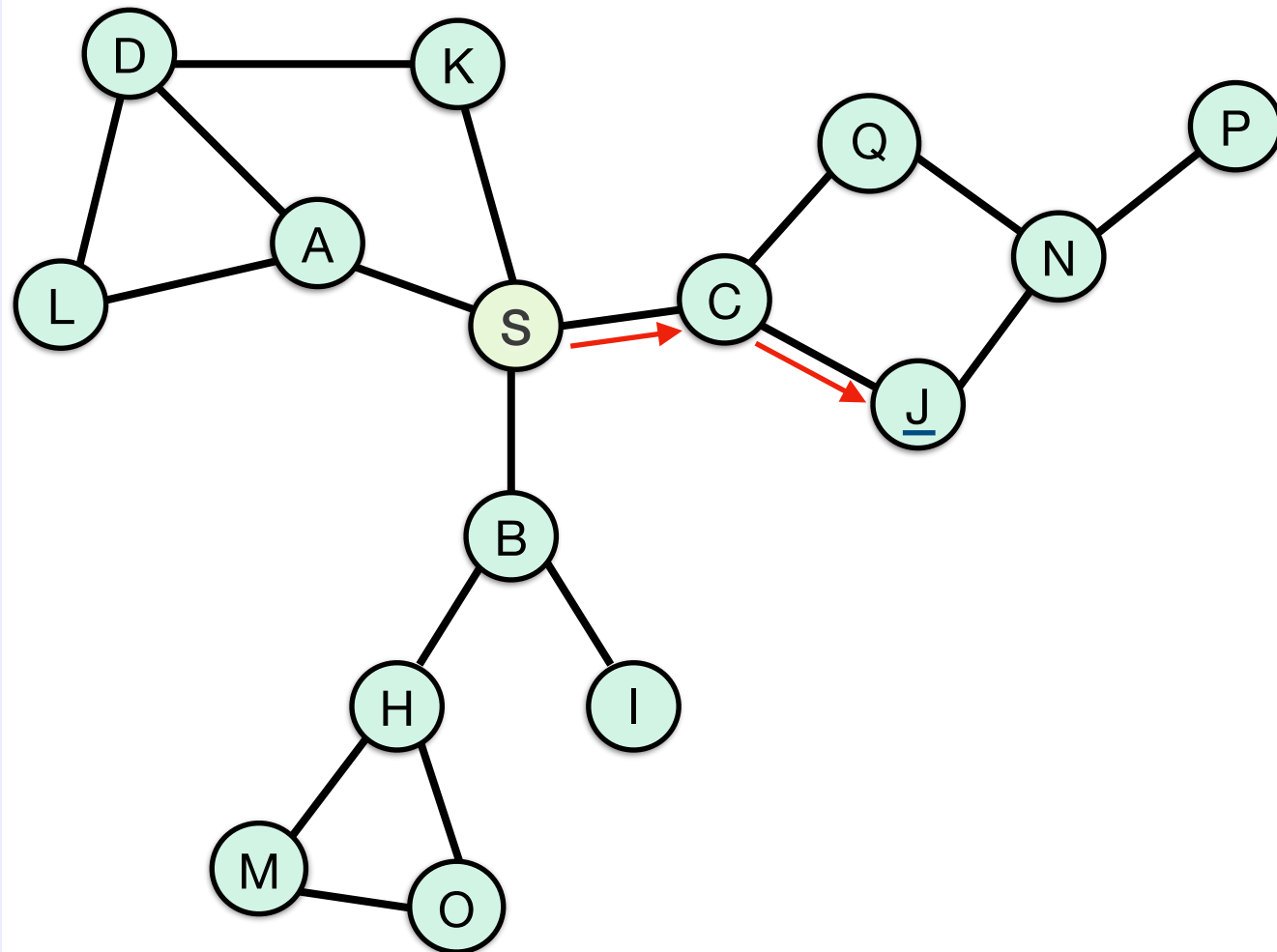
DFS tree

DFS Traversal and DFS Tree



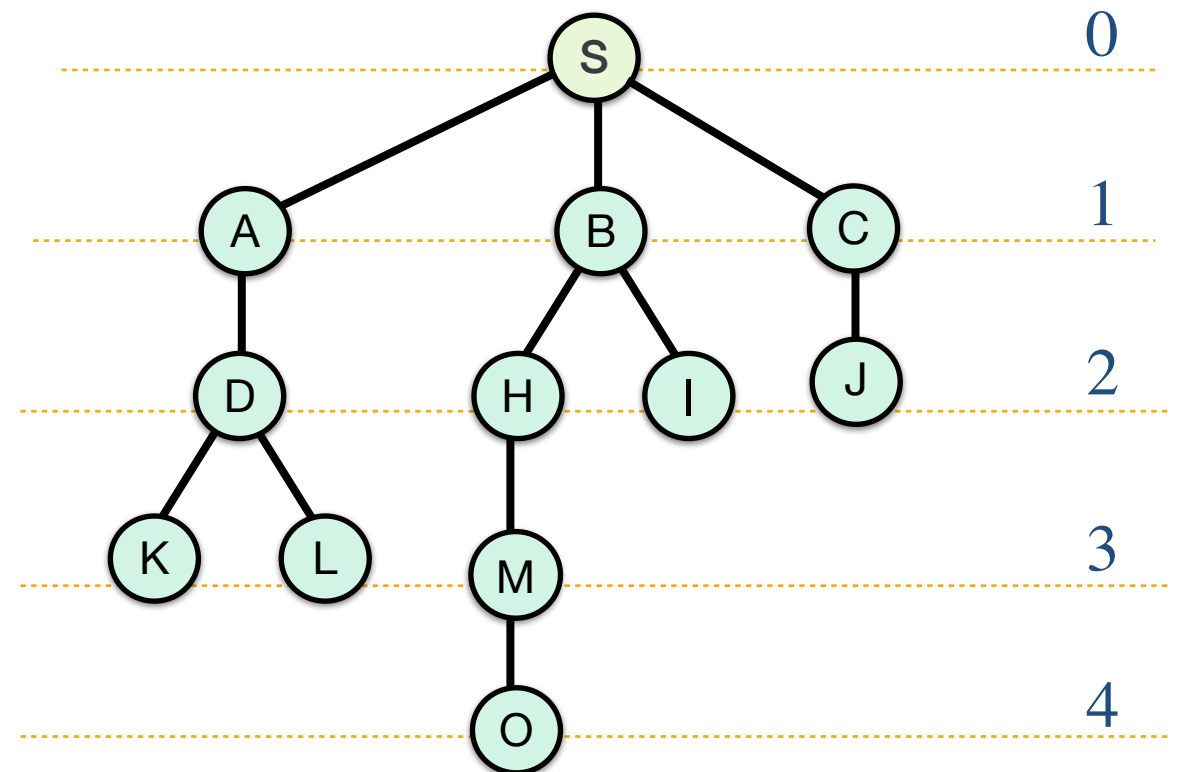
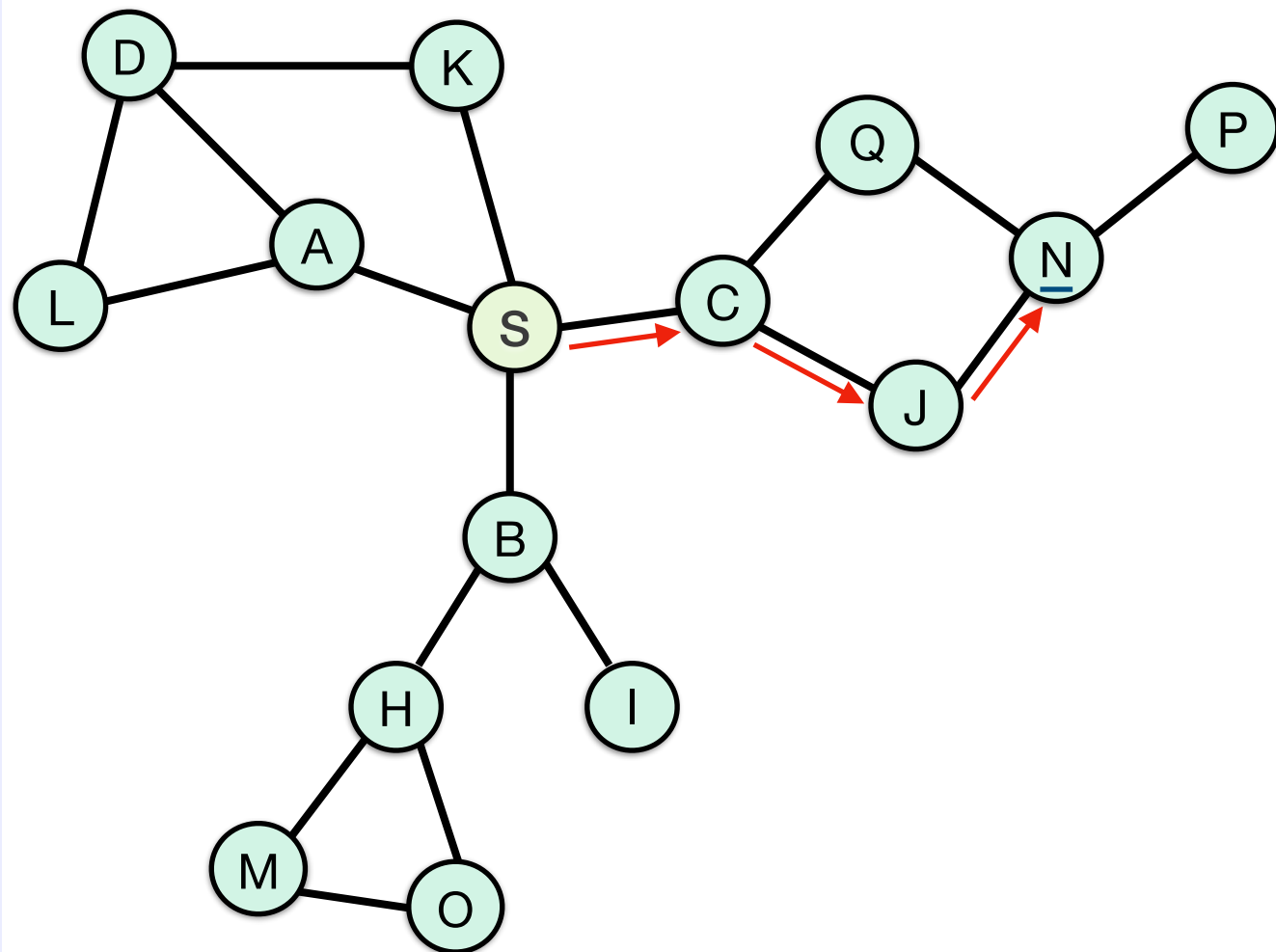
DFS tree

DFS Traversal and DFS Tree



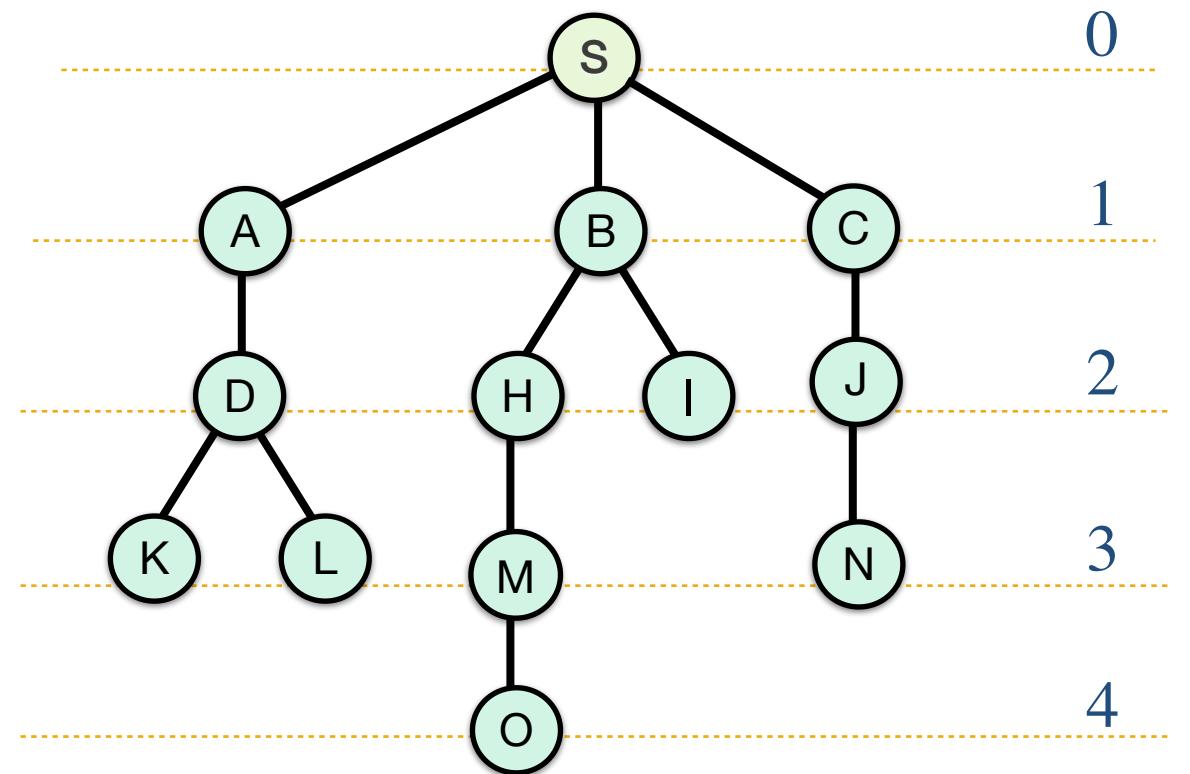
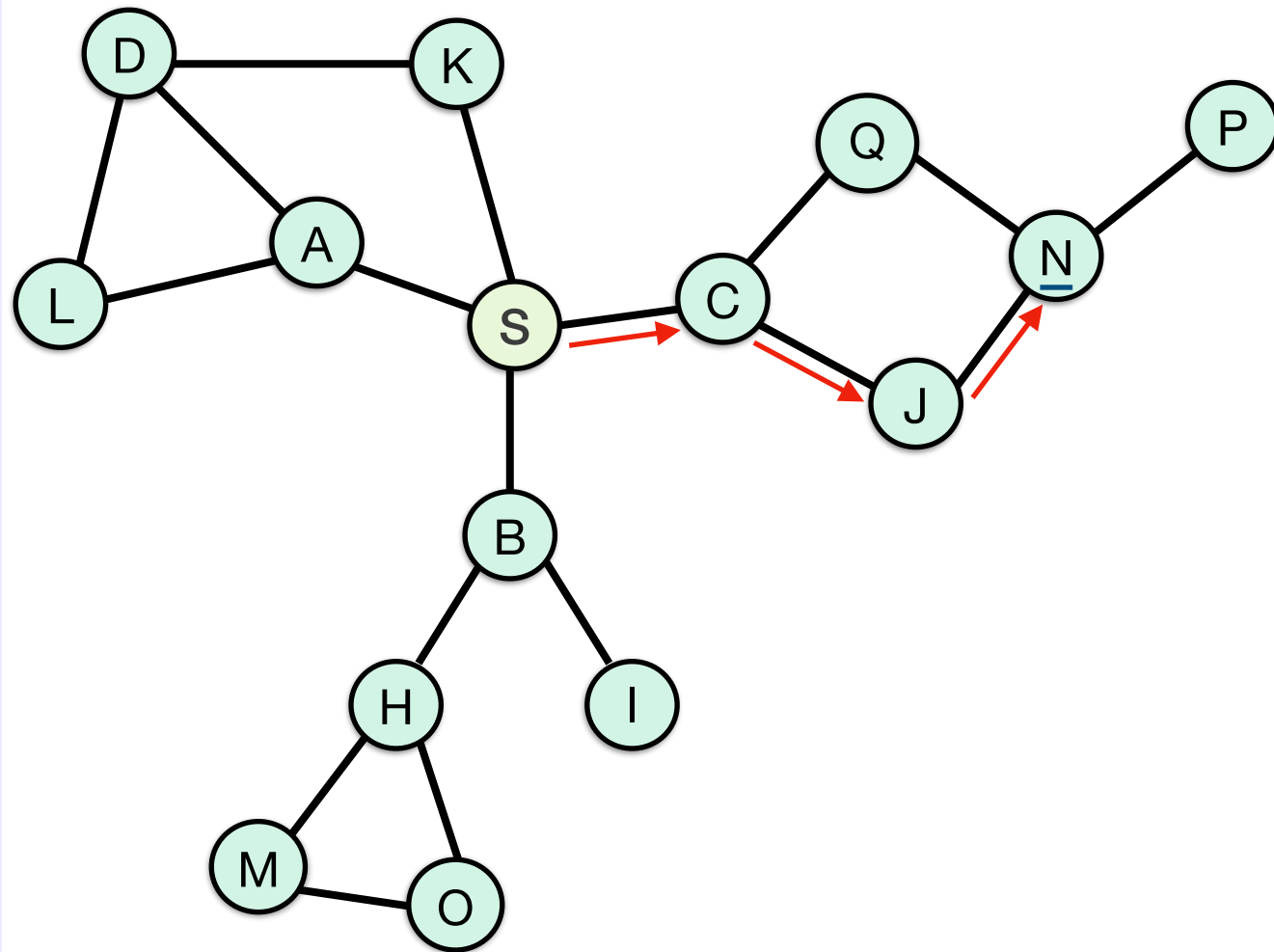
DFS tree

DFS Traversal and DFS Tree



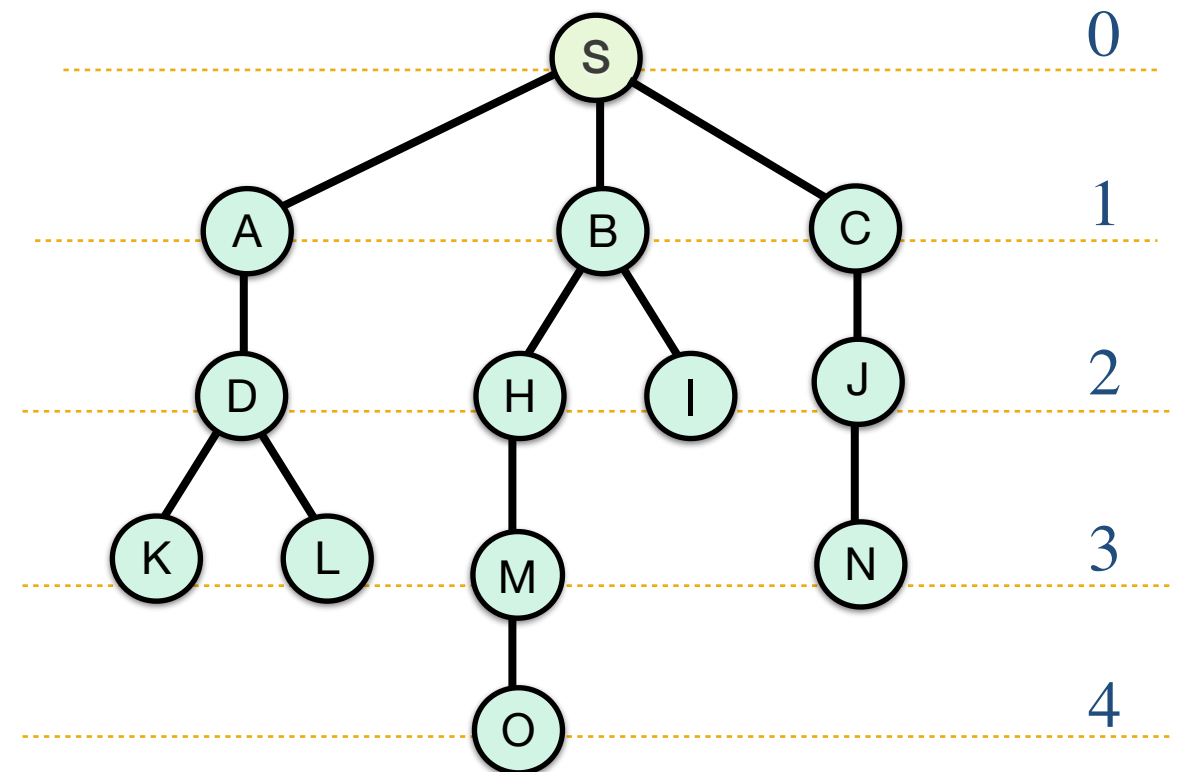
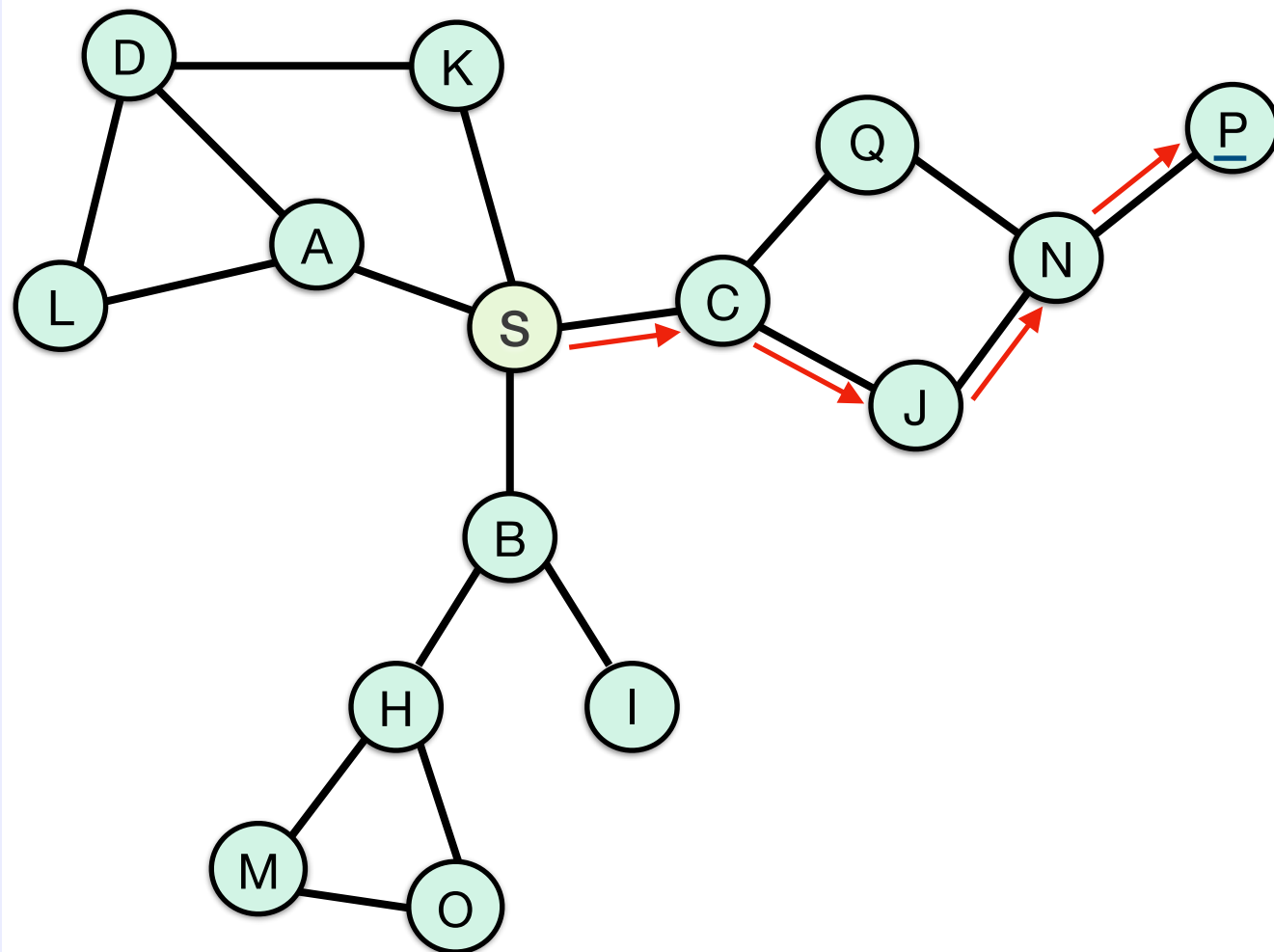
DFS tree

DFS Traversal and DFS Tree



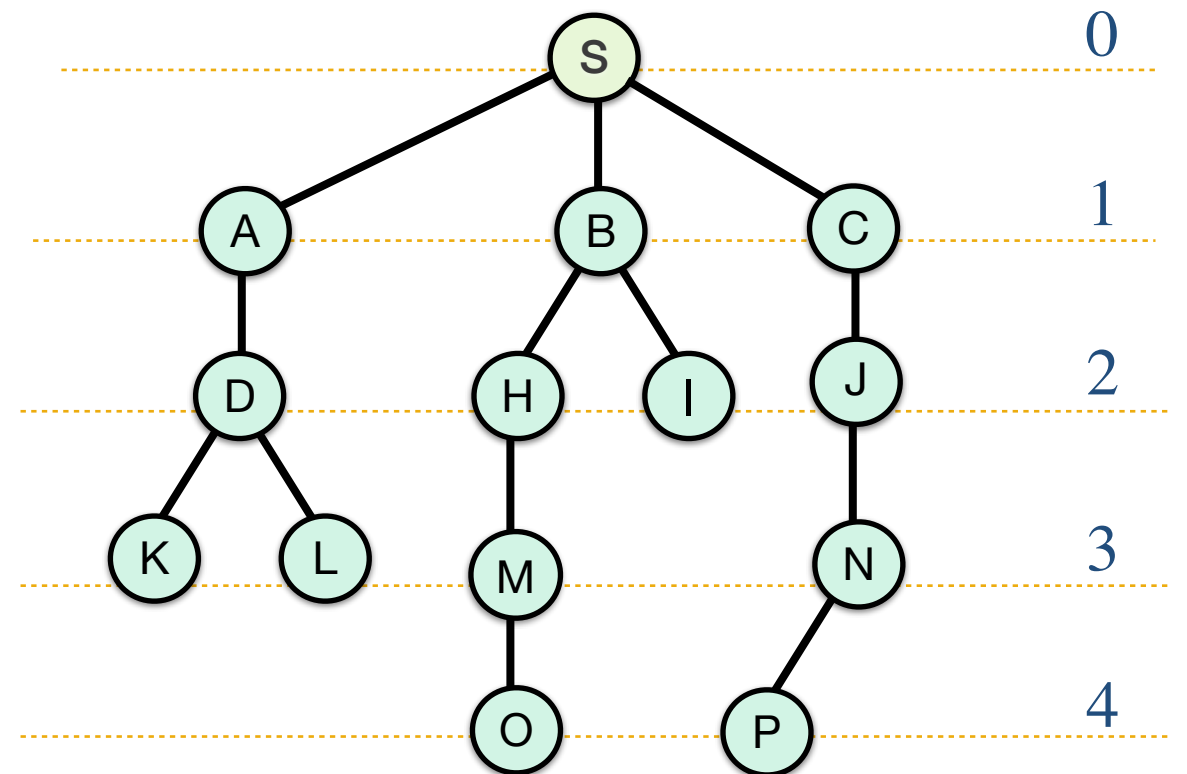
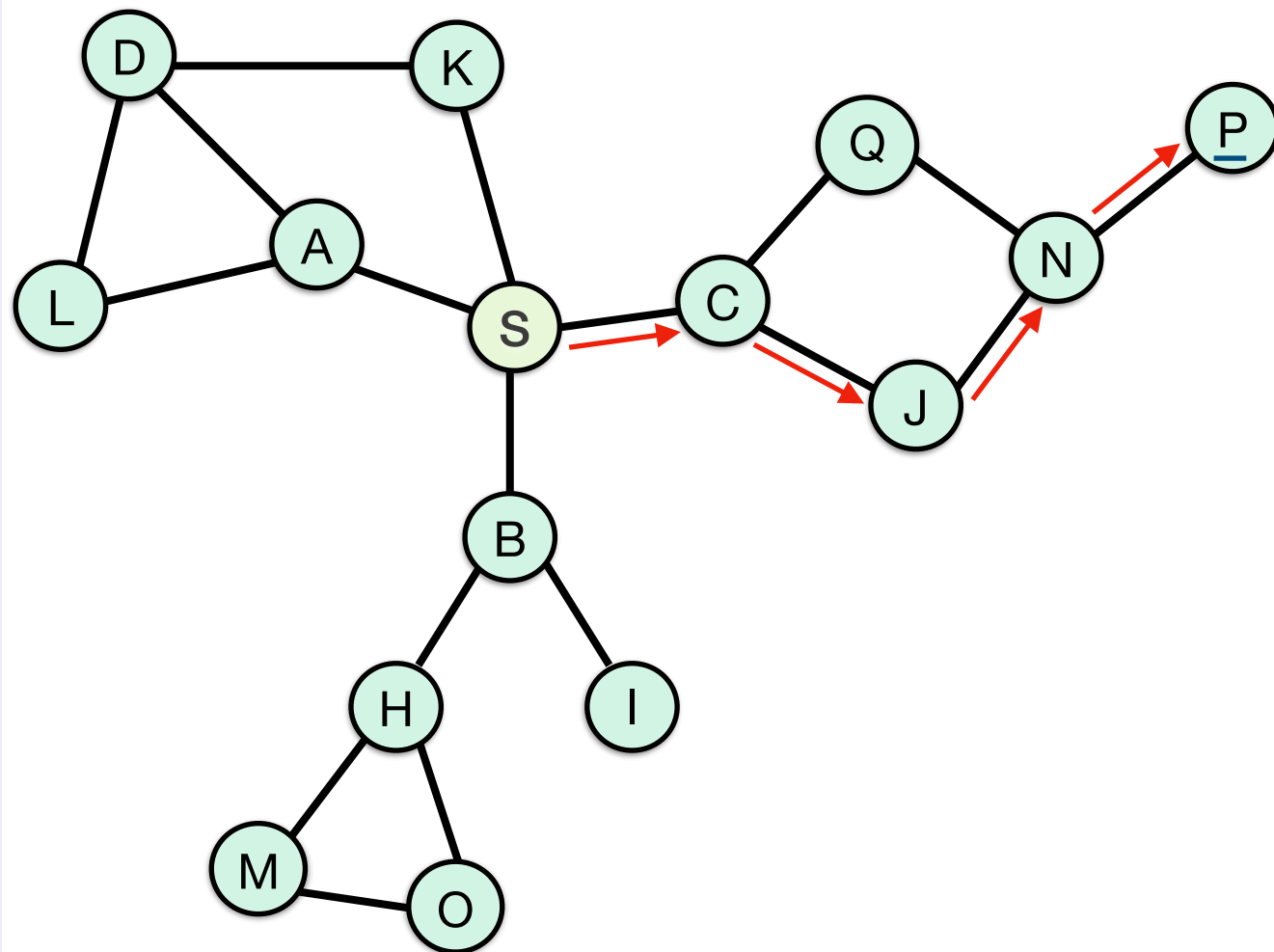
DFS tree

DFS Traversal and DFS Tree



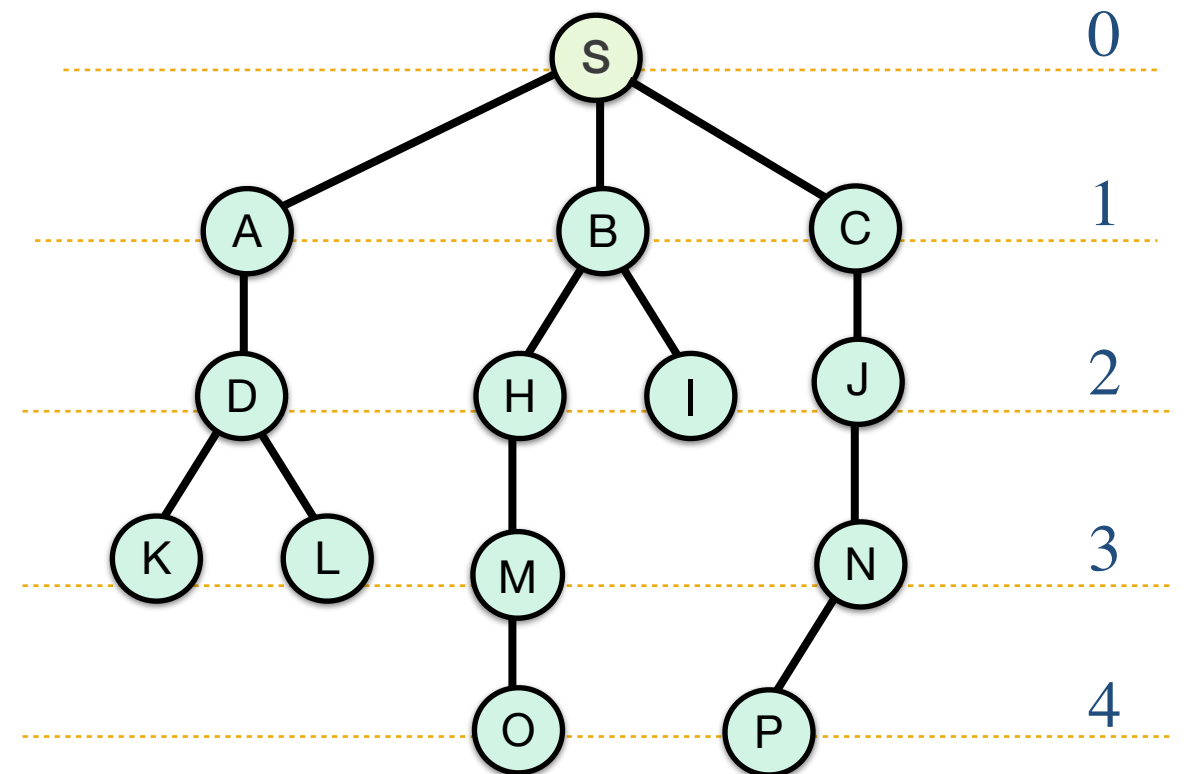
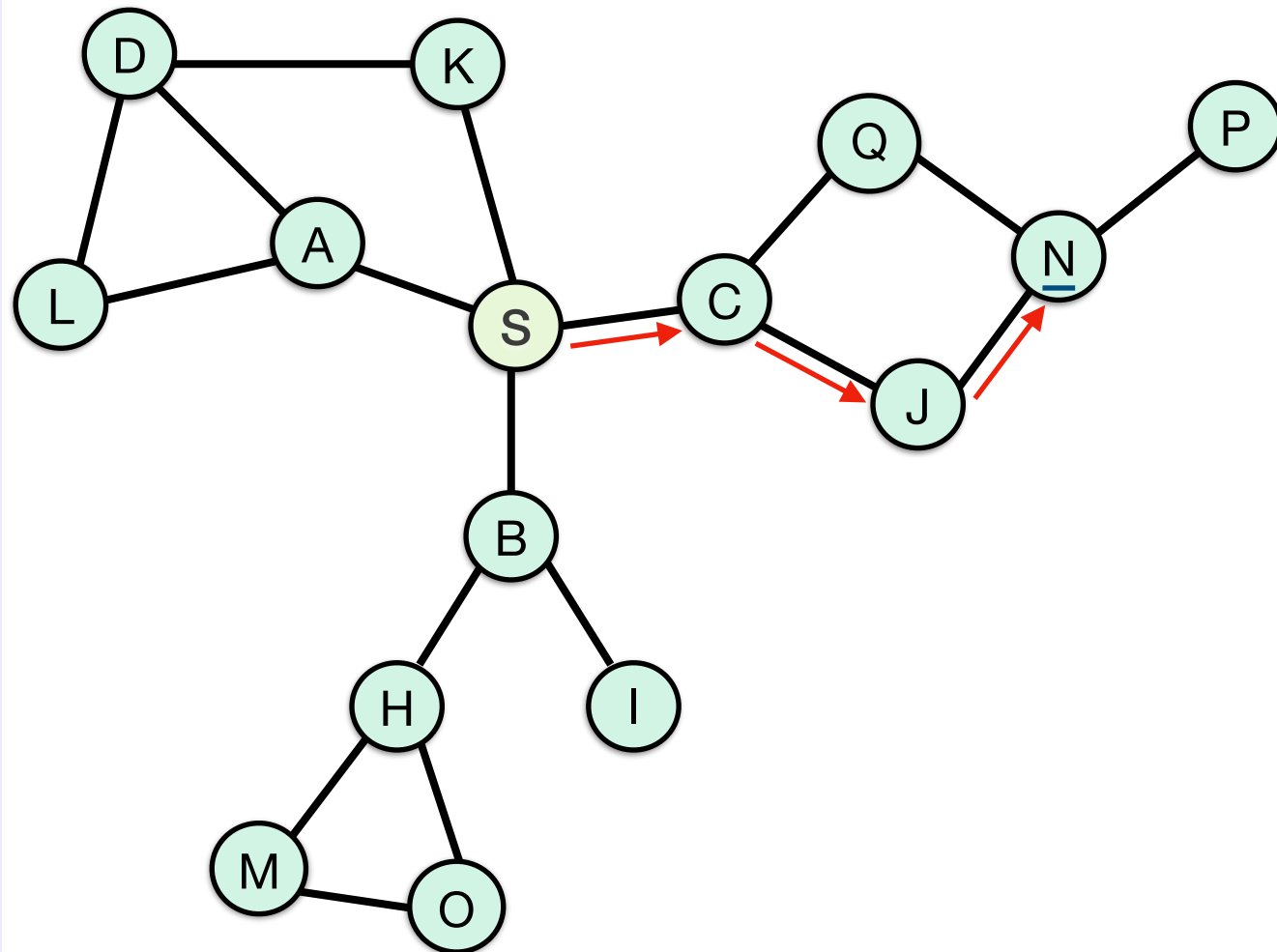
DFS tree

DFS Traversal and DFS Tree

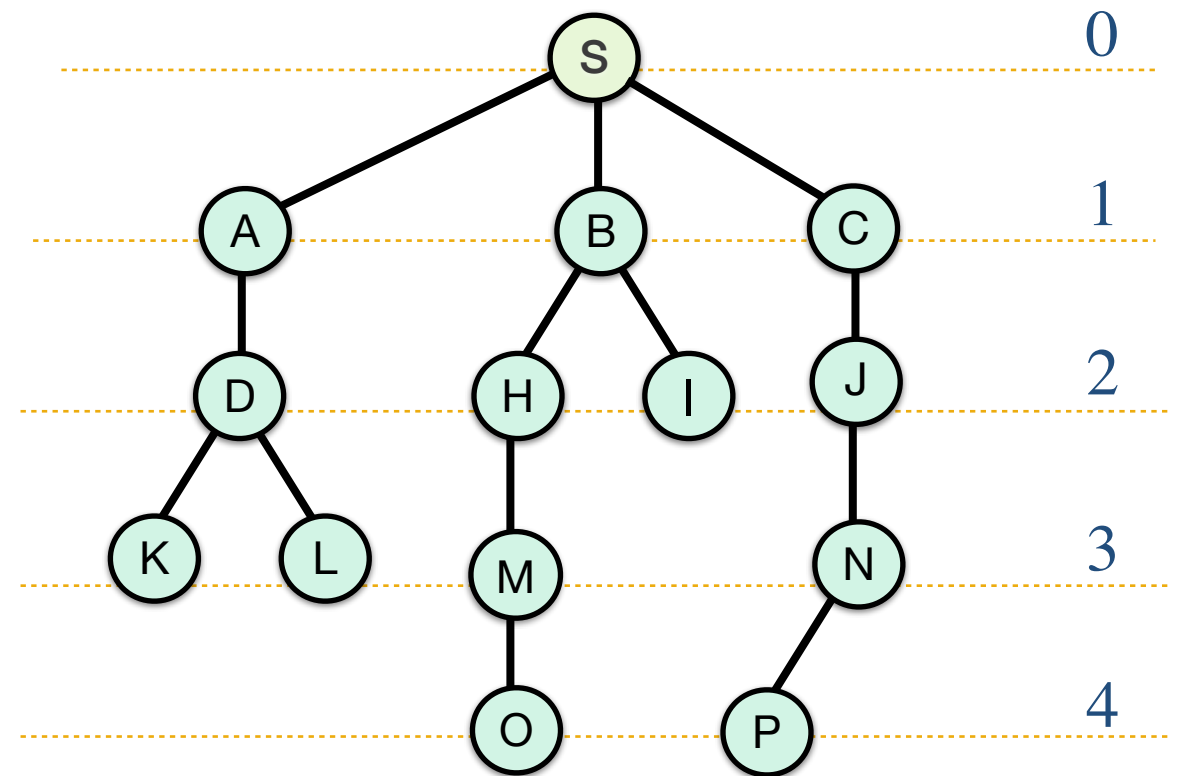
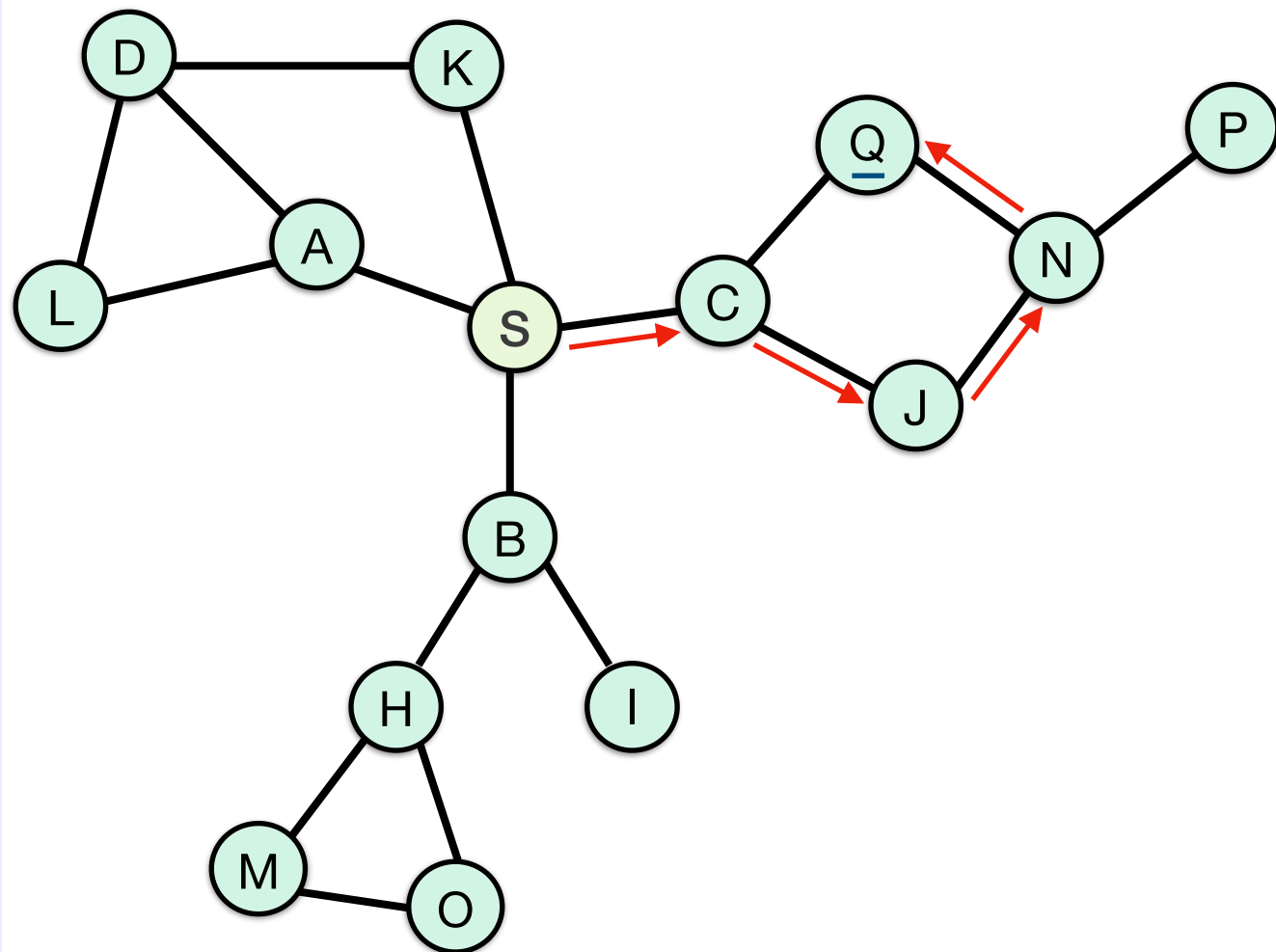


DFS tree

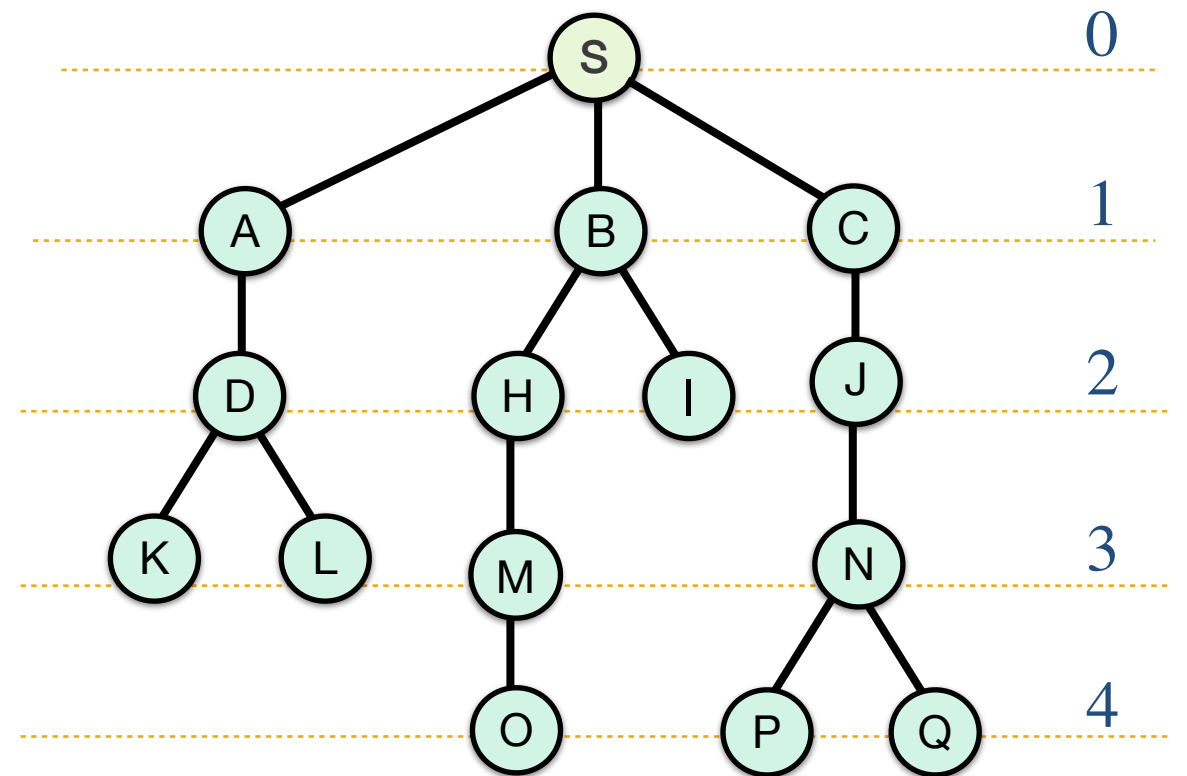
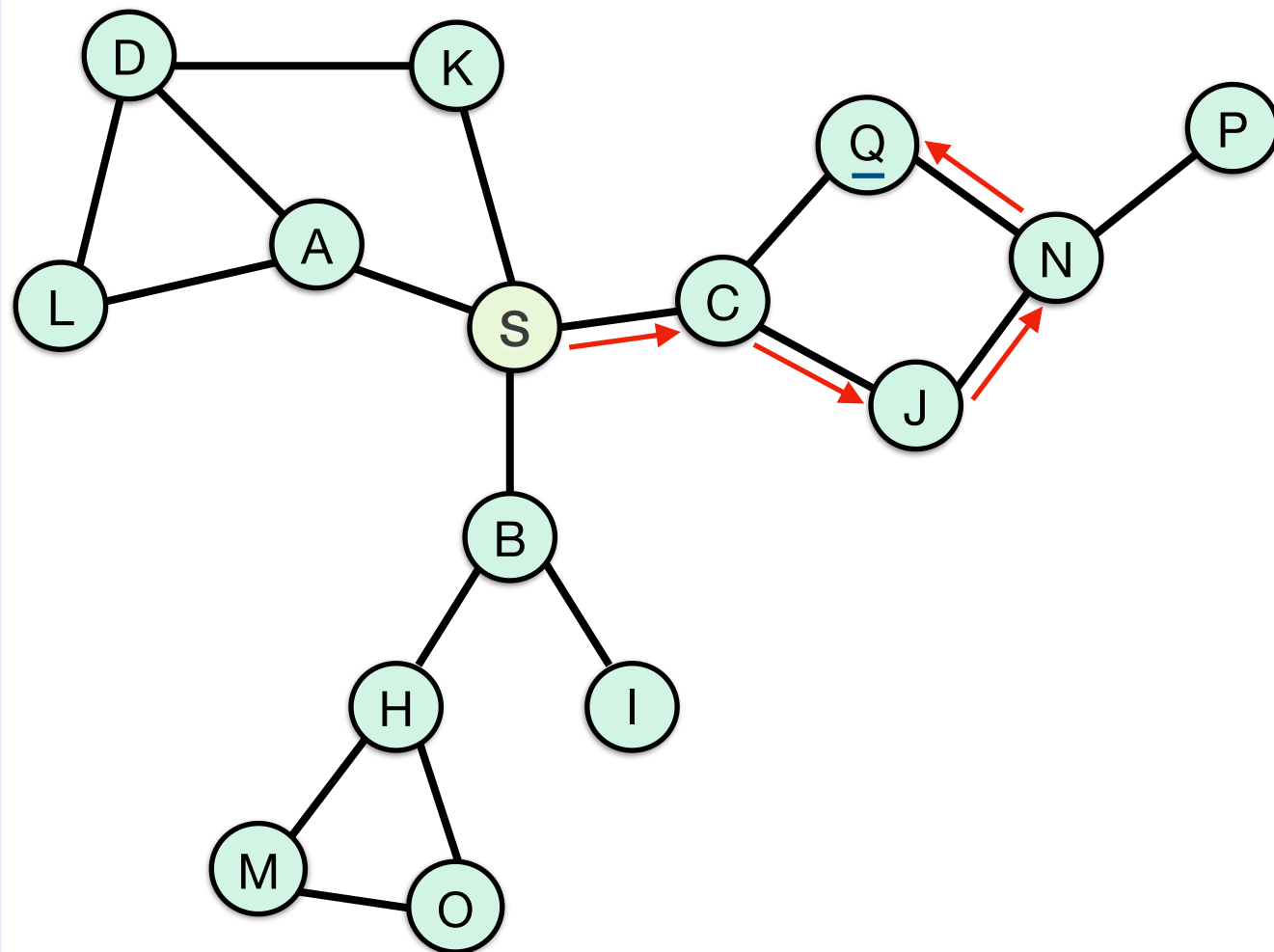
DFS Traversal and DFS Tree



DFS Traversal and DFS Tree

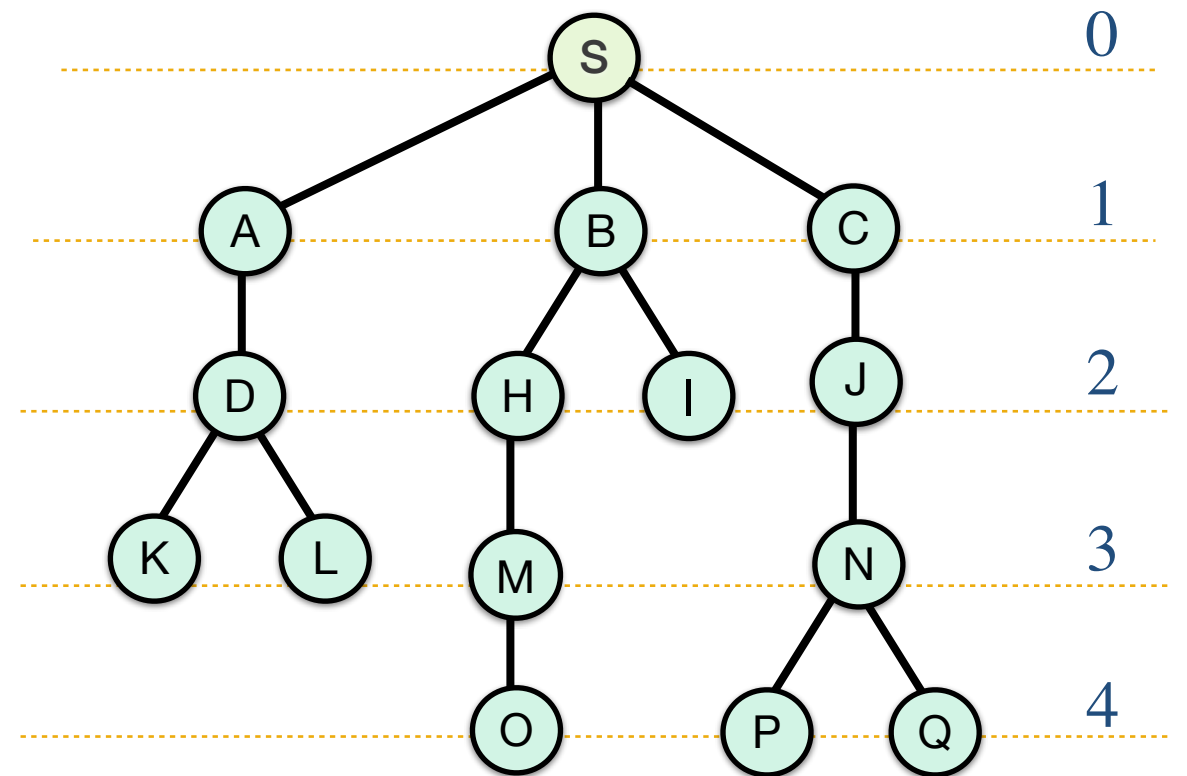
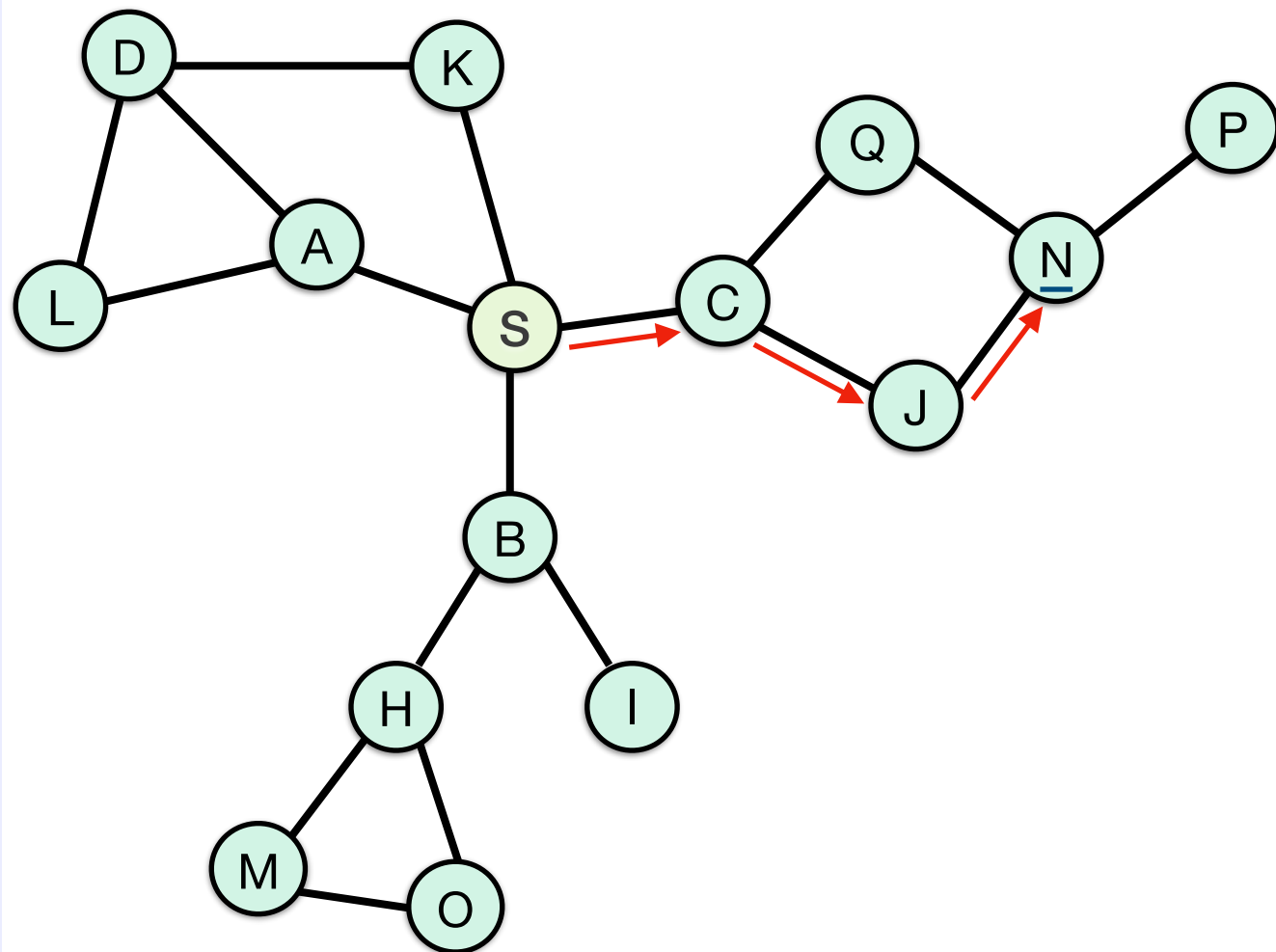


DFS Traversal and DFS Tree



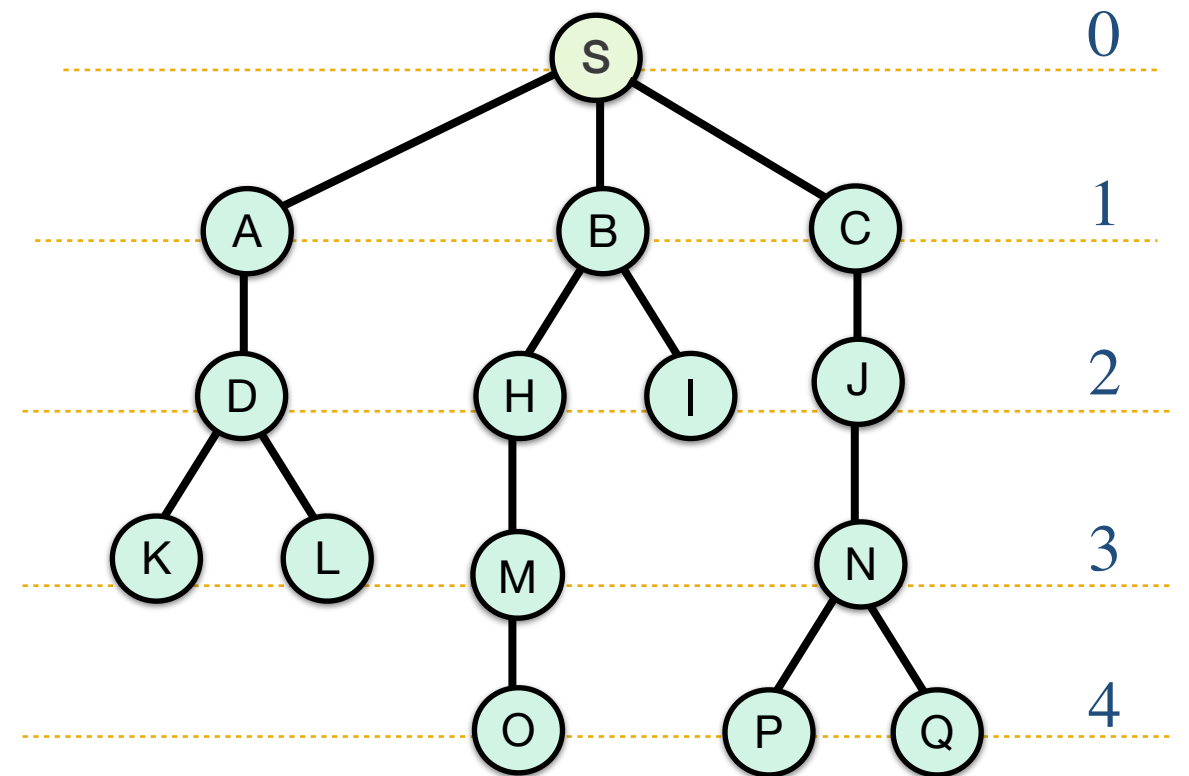
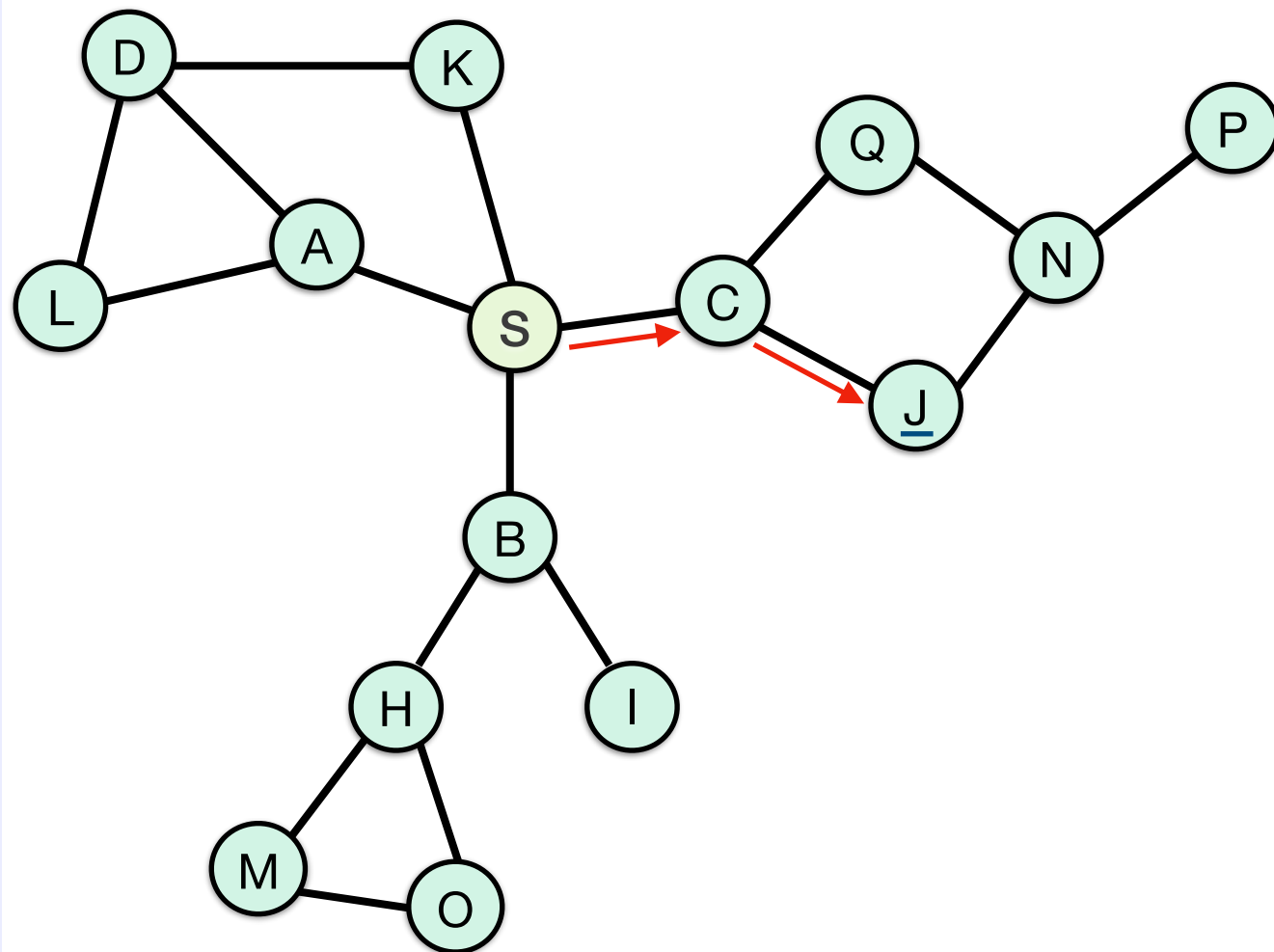
DFS tree

DFS Traversal and DFS Tree



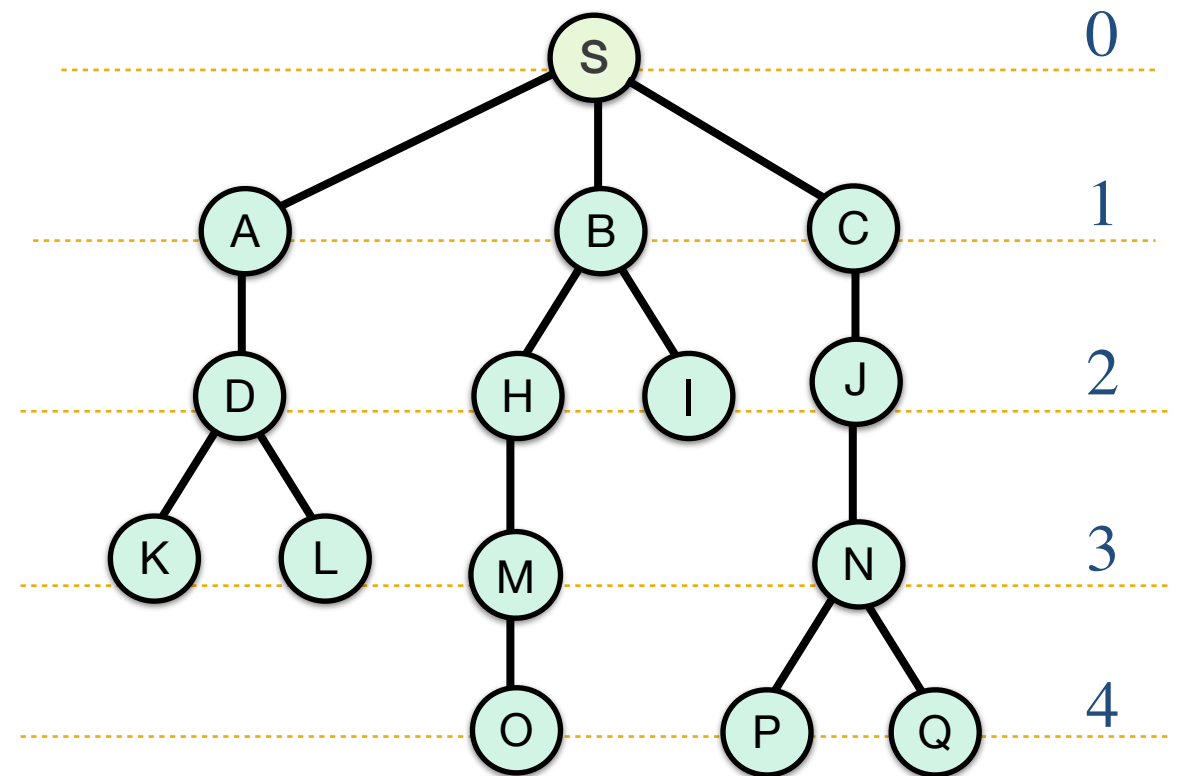
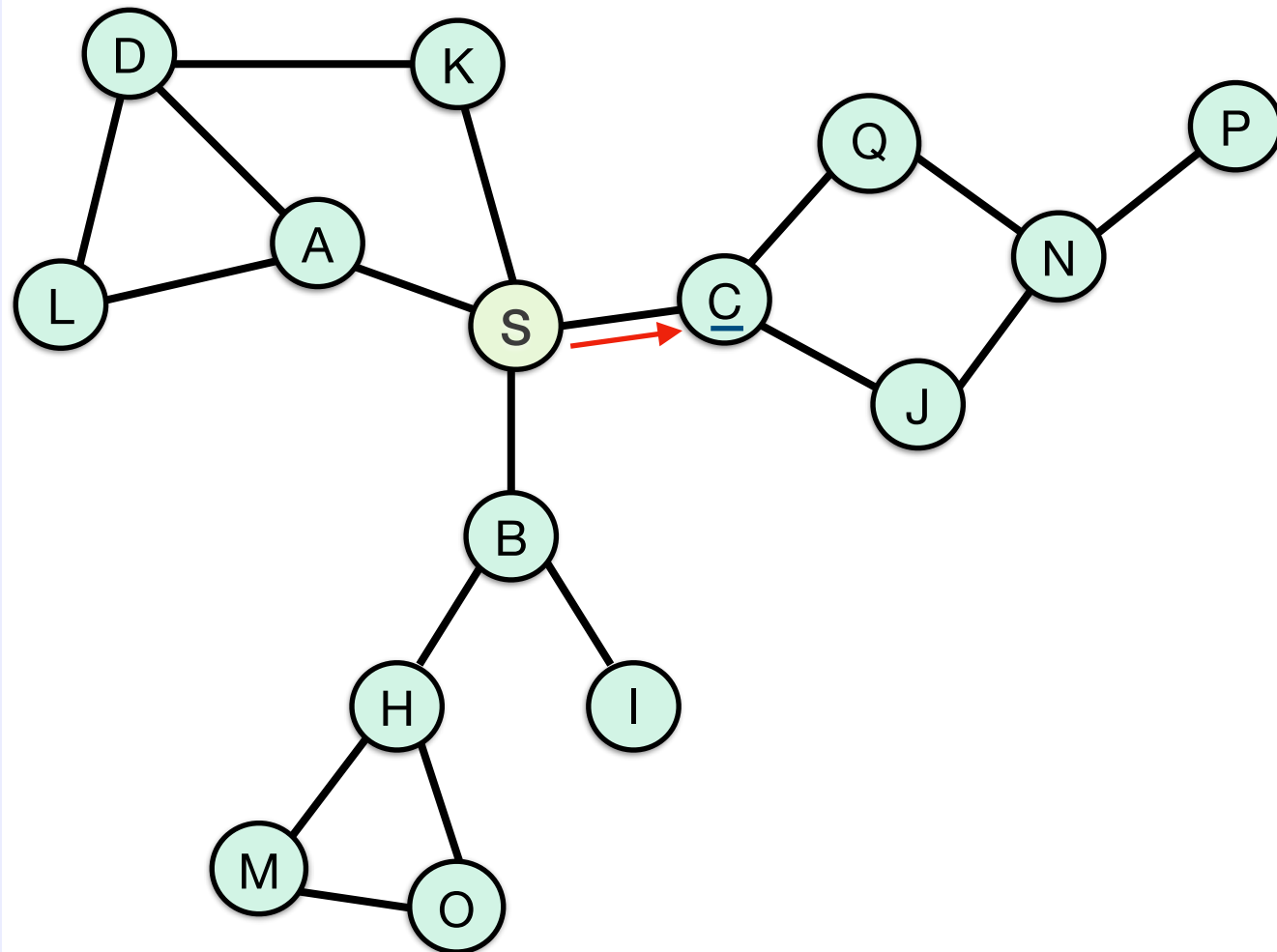
DFS tree

DFS Traversal and DFS Tree



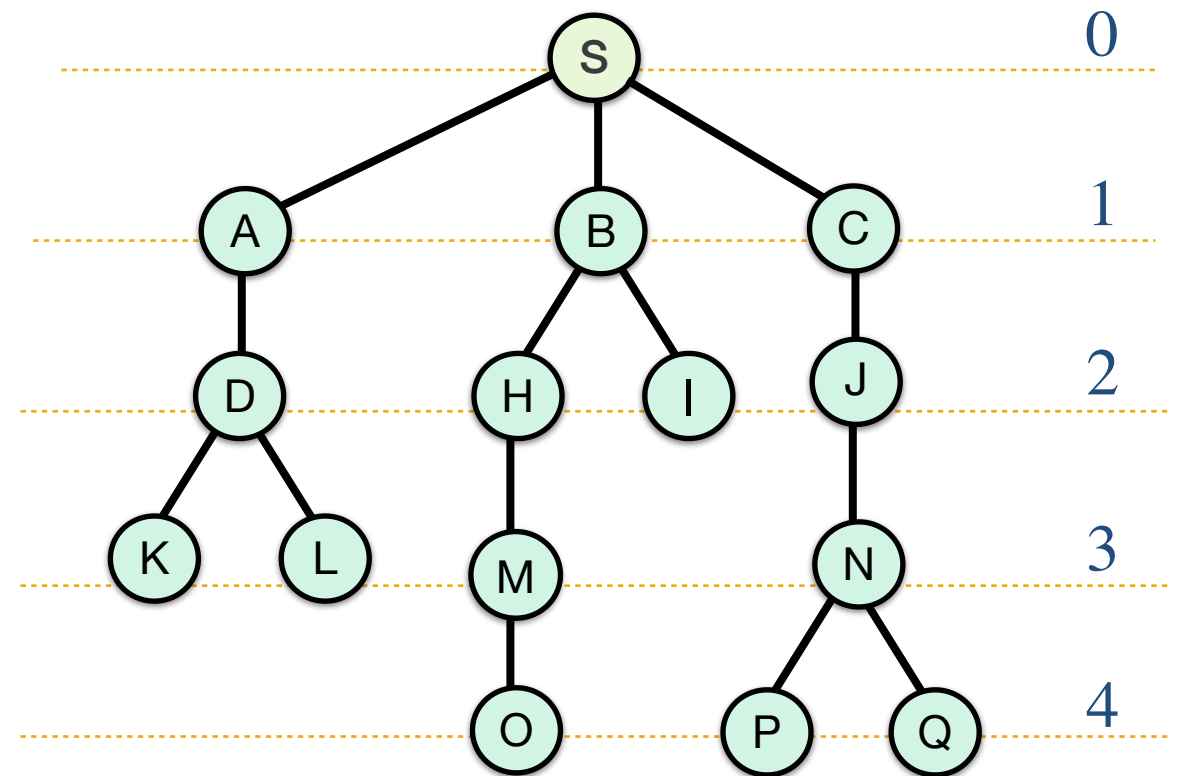
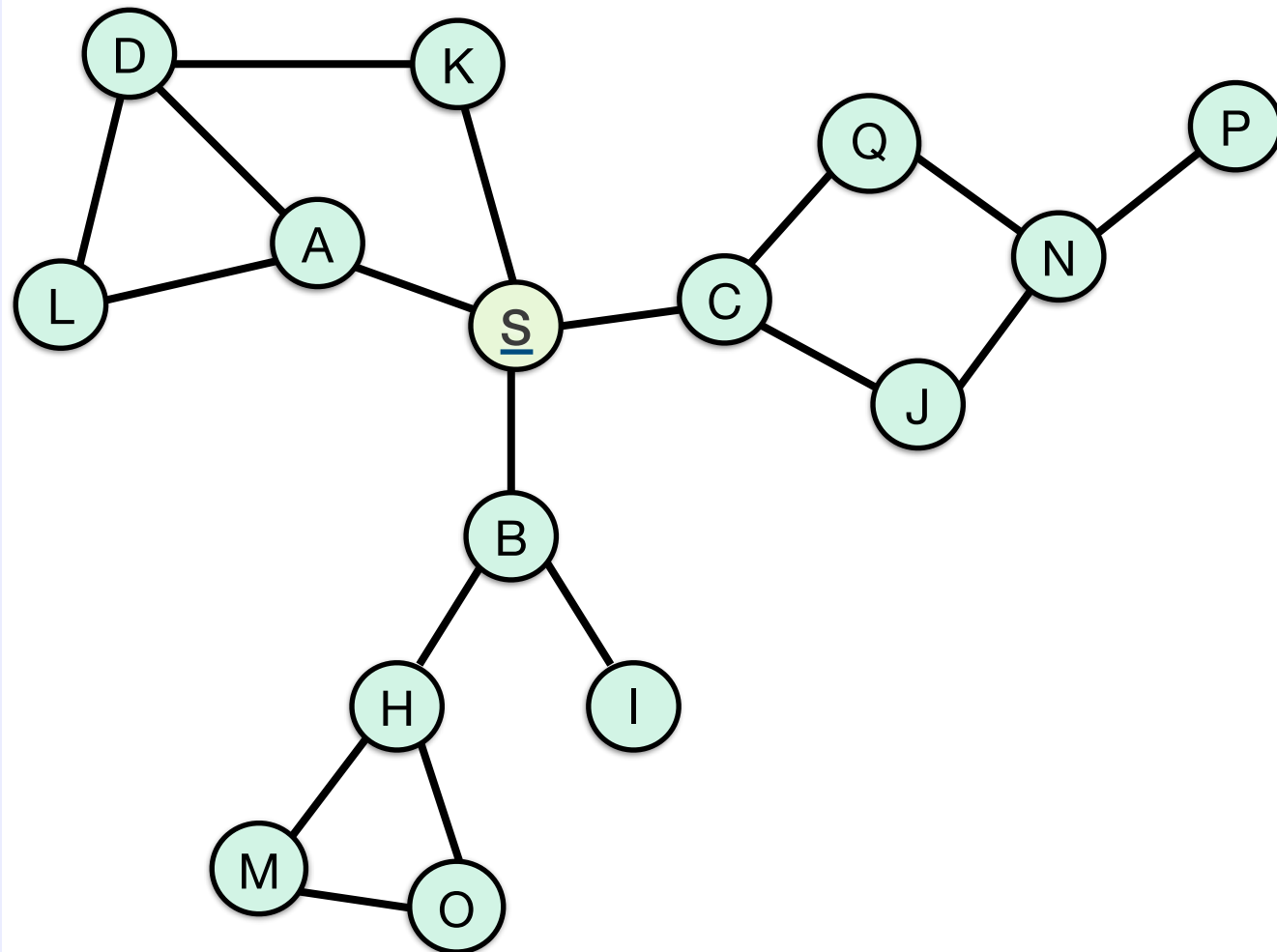
DFS tree

DFS Traversal and DFS Tree



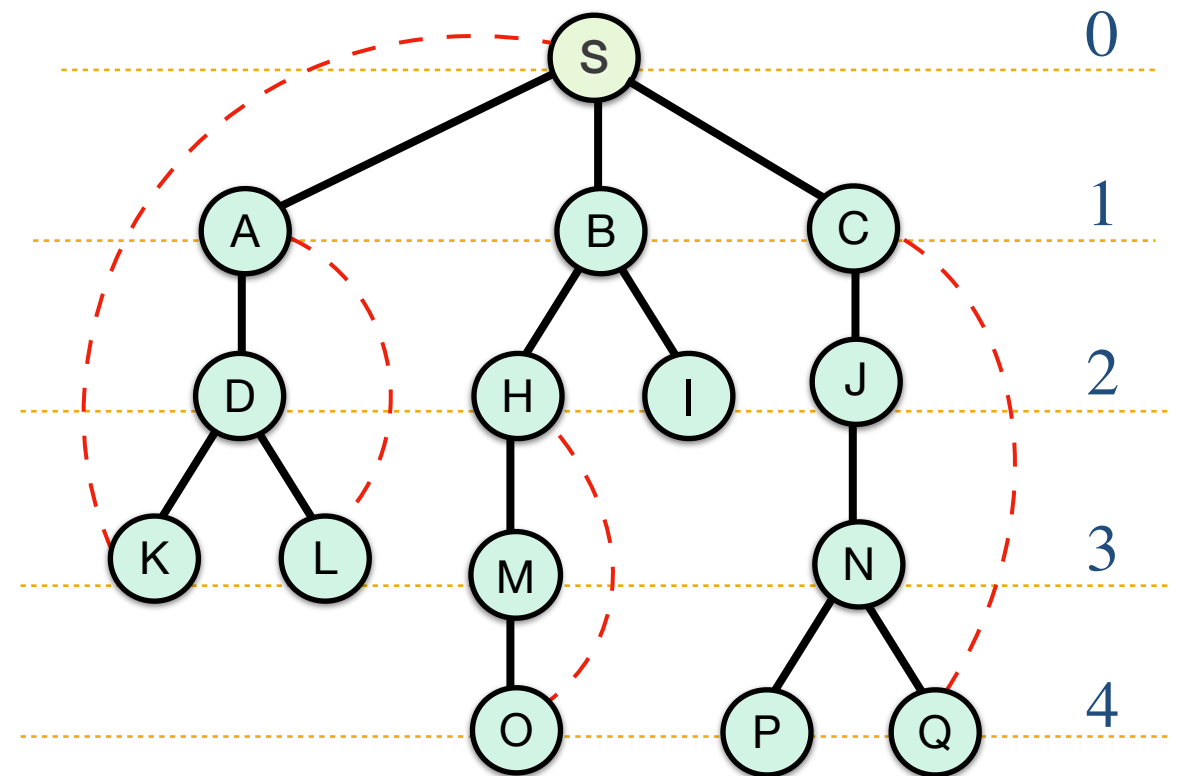
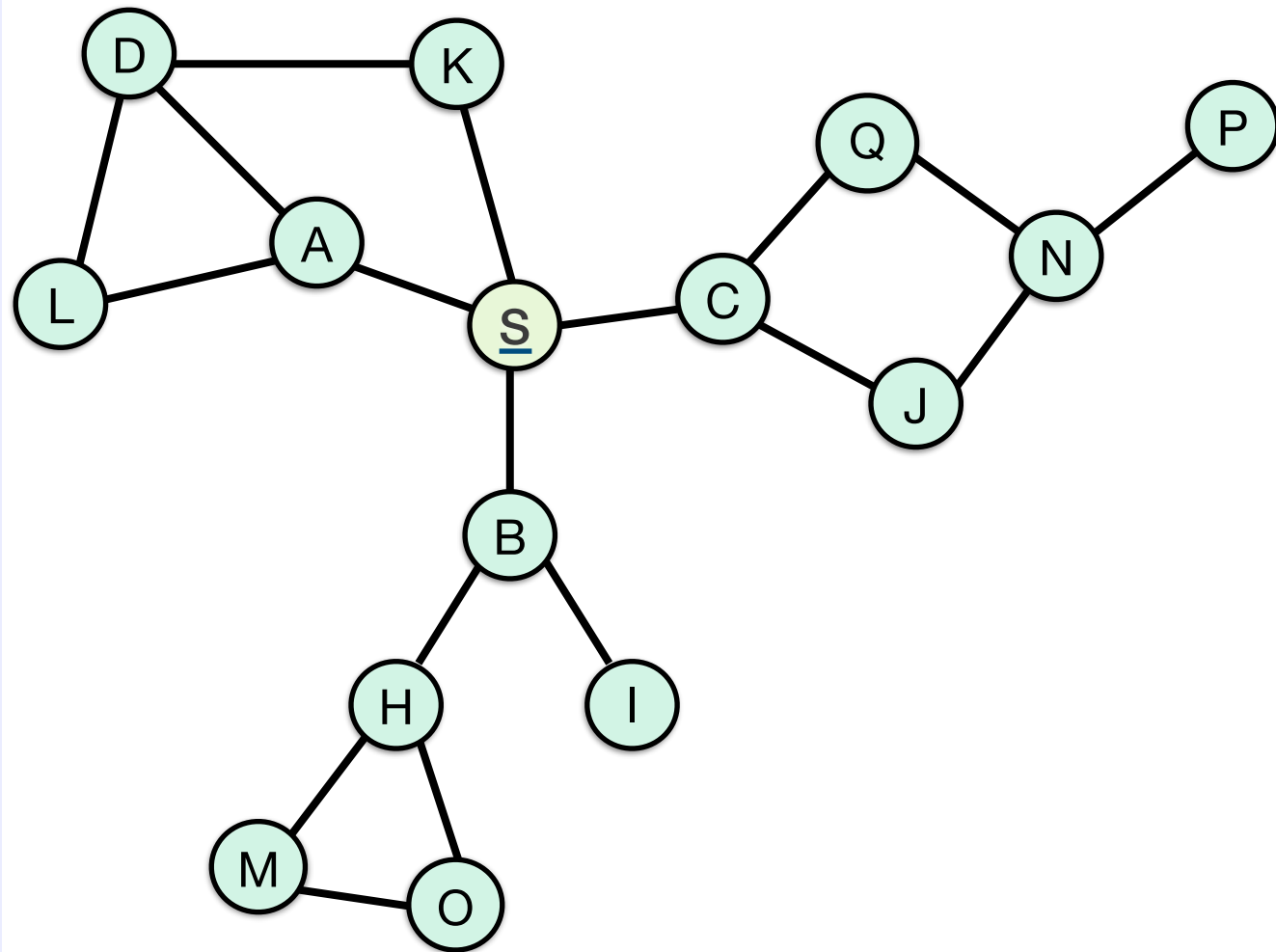
DFS tree

DFS Traversal and DFS Tree



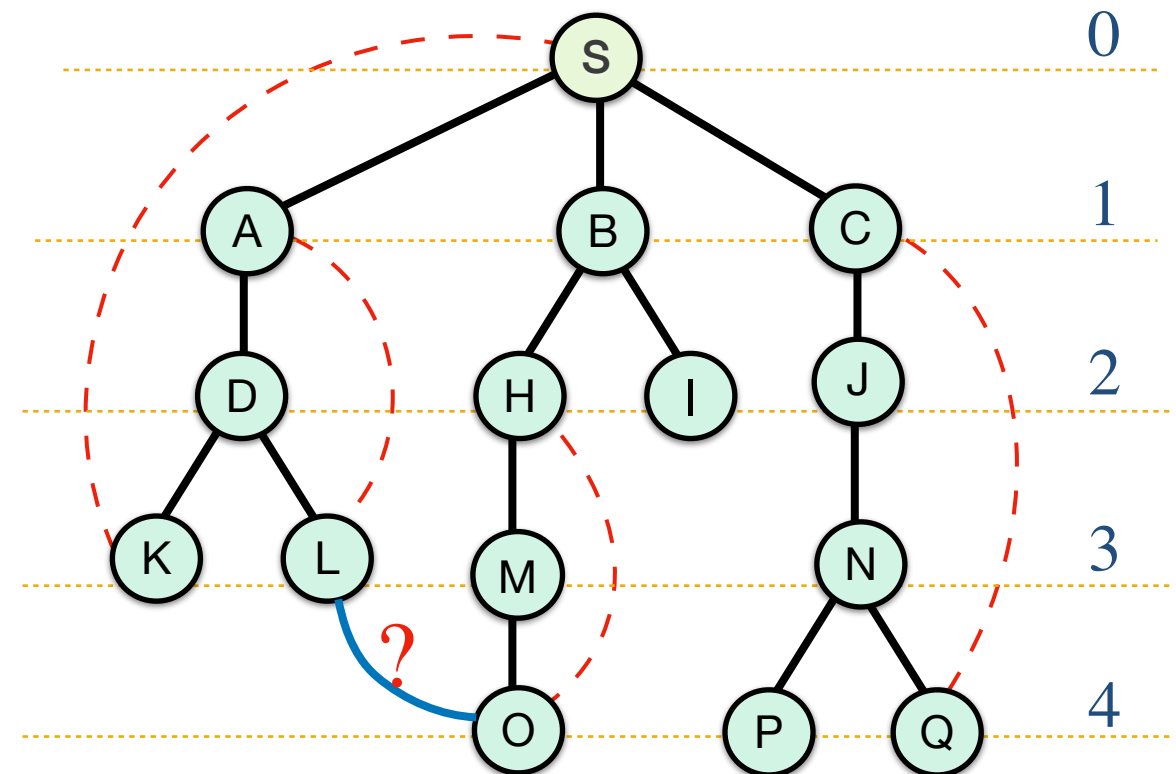
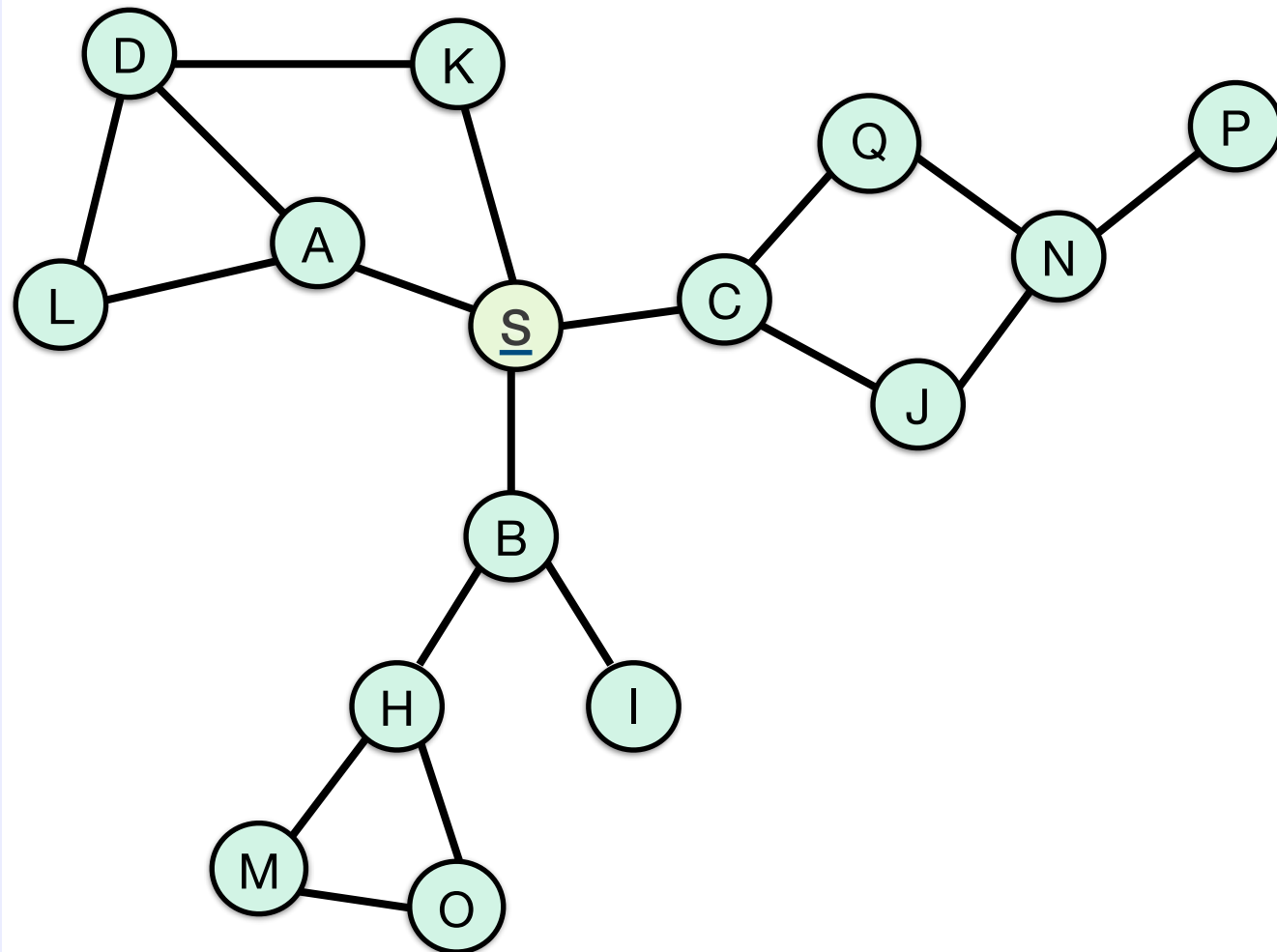
DFS tree

DFS Traversal and DFS Tree



DFS tree

DFS Traversal and DFS Tree



DFS tree

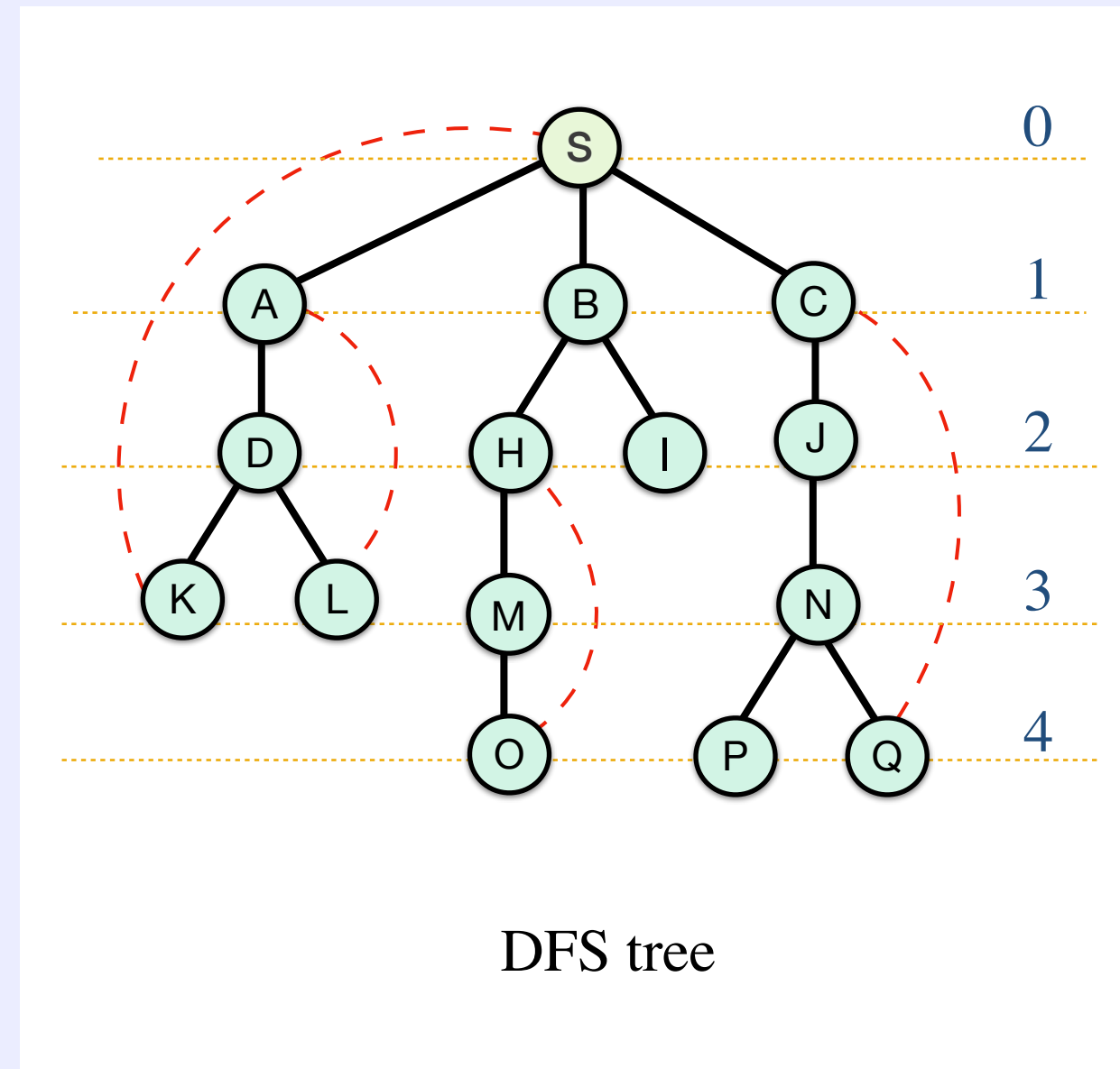
Question: Can there be a non-tree edge (x, y) in DFS tree such that x and y do not have ancestor-descendant relation?

No

DFS Traversal and DFS Tree

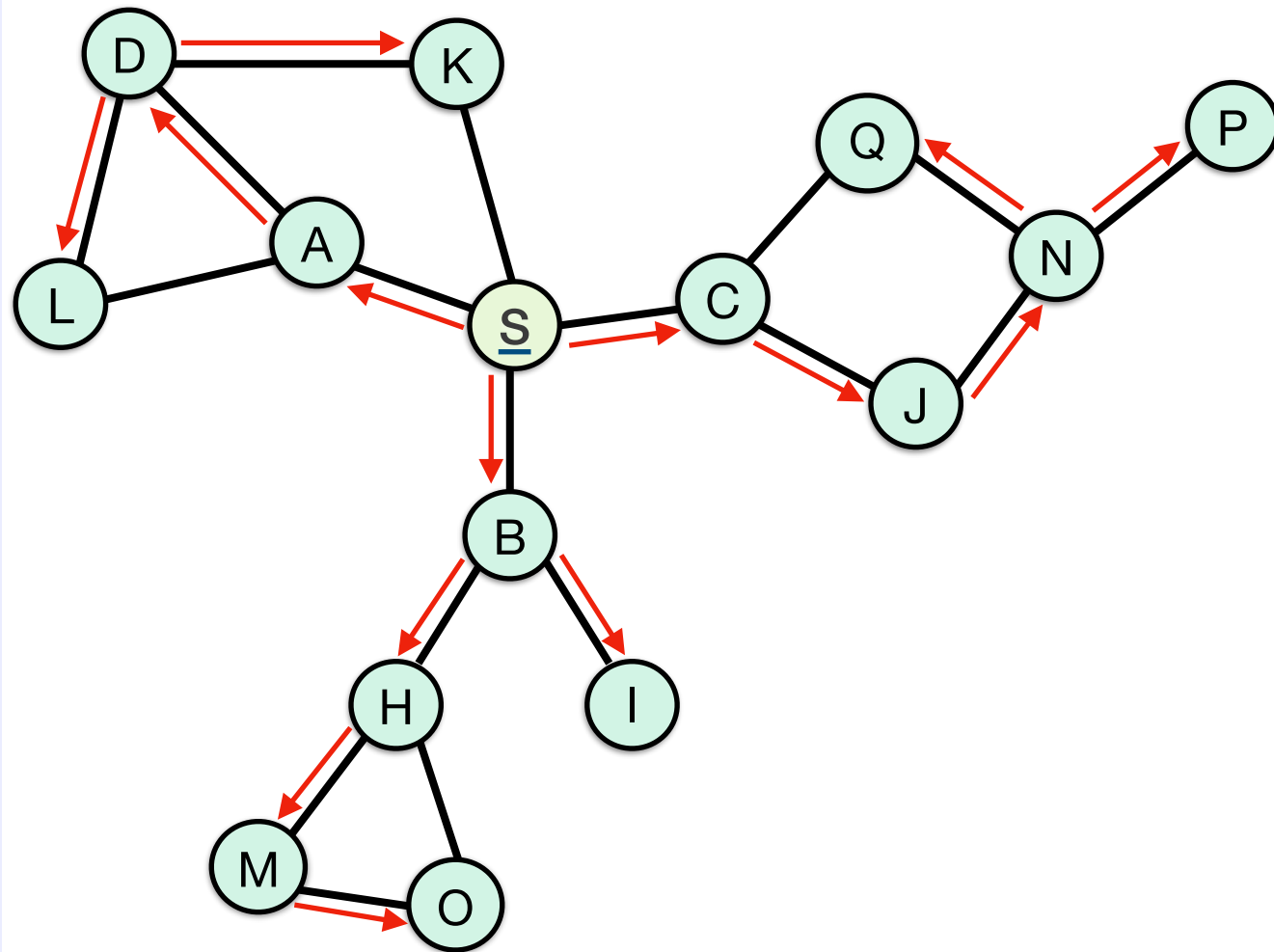
Definition of DFS Tree

Rooted tree
obtained by
DFS traversal.



What should be
the *algorithm*?

DFS Traversal Algorithm



Algorithm:

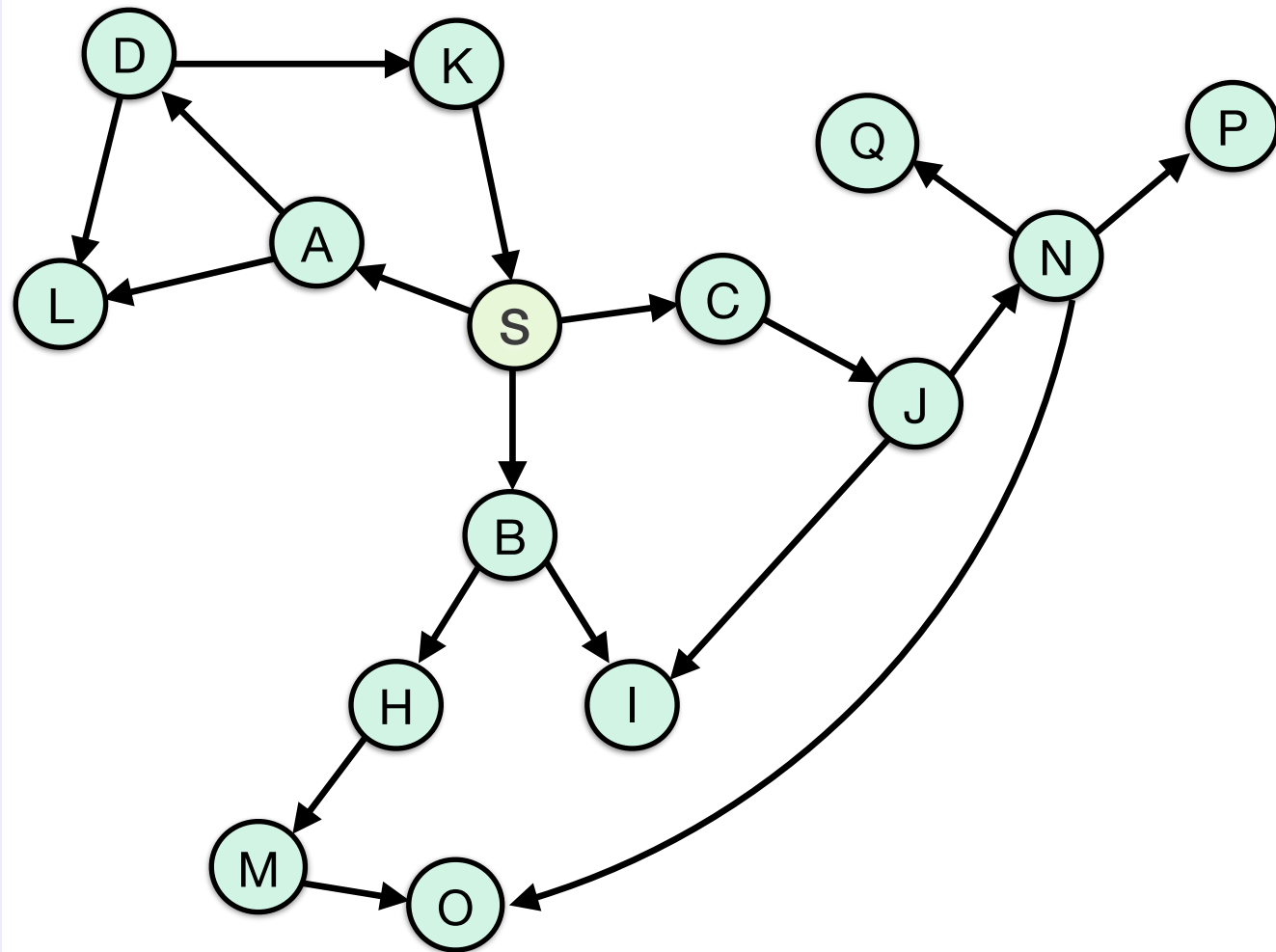
1. For each $v \in V$:
 $visited(v) = false$;
2. Invoke DFS(s);

DFS(x):

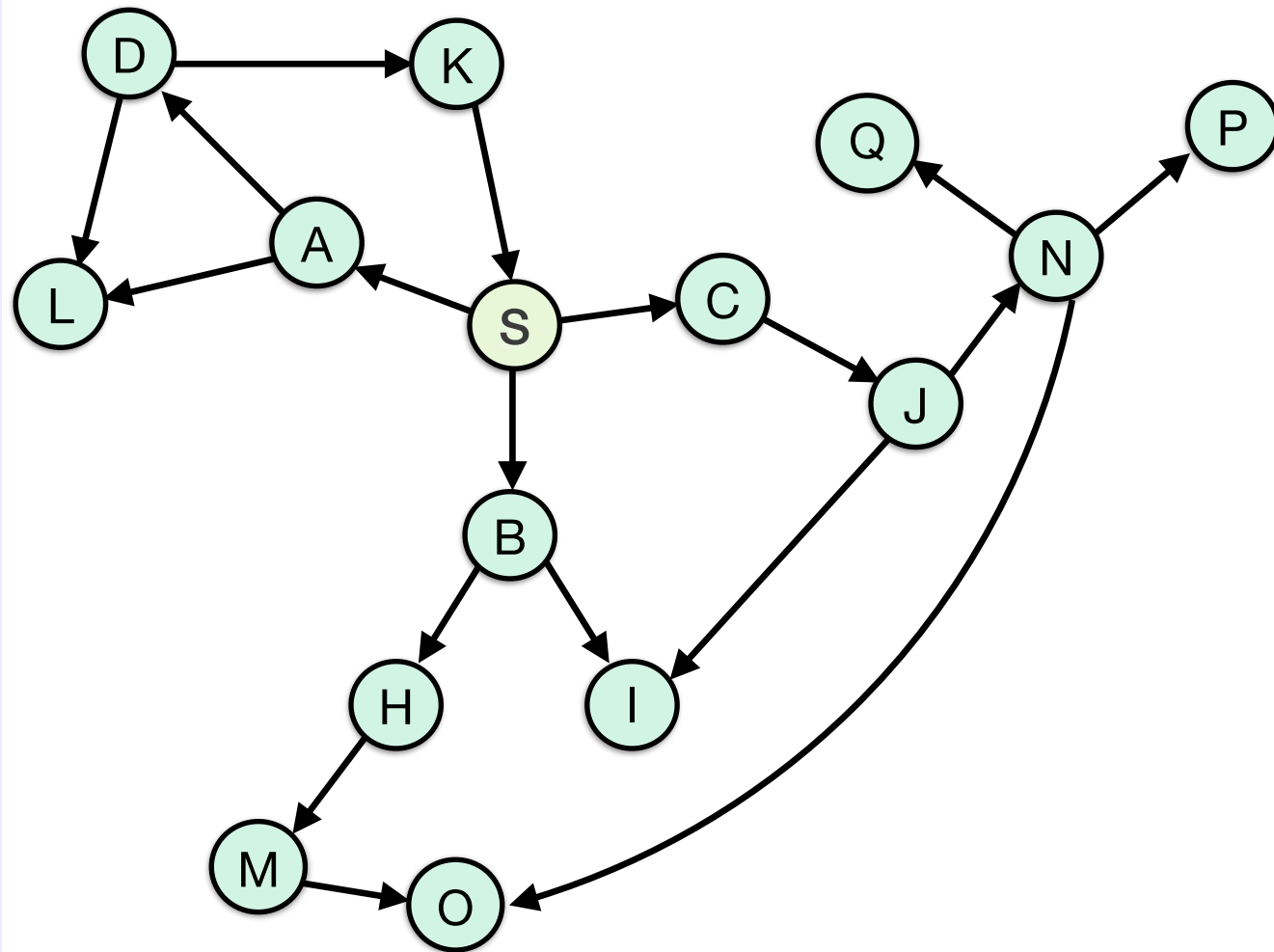
1. Set $visited(x) = true$;
2. For each $y \in N(x)$:
If ($visited(y) = false$):
Invoke DFS(y)

Time = $O(m + n)$

What if G is directed?



What if G is directed?



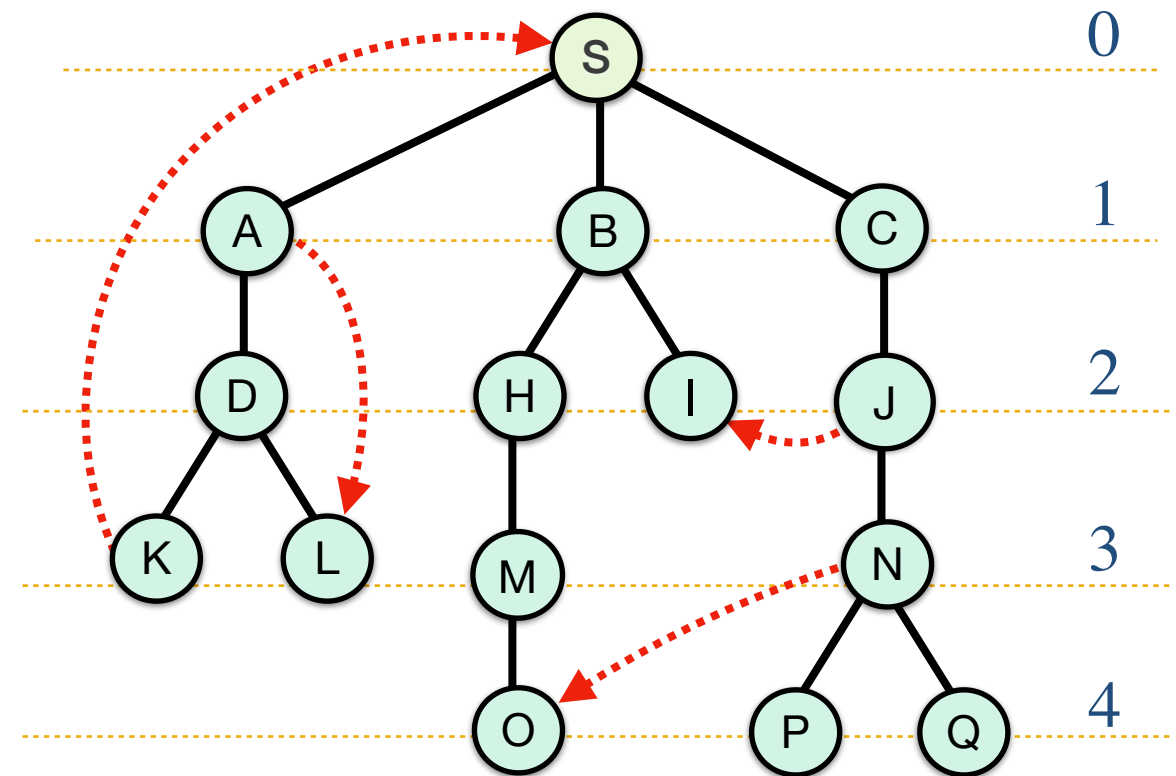
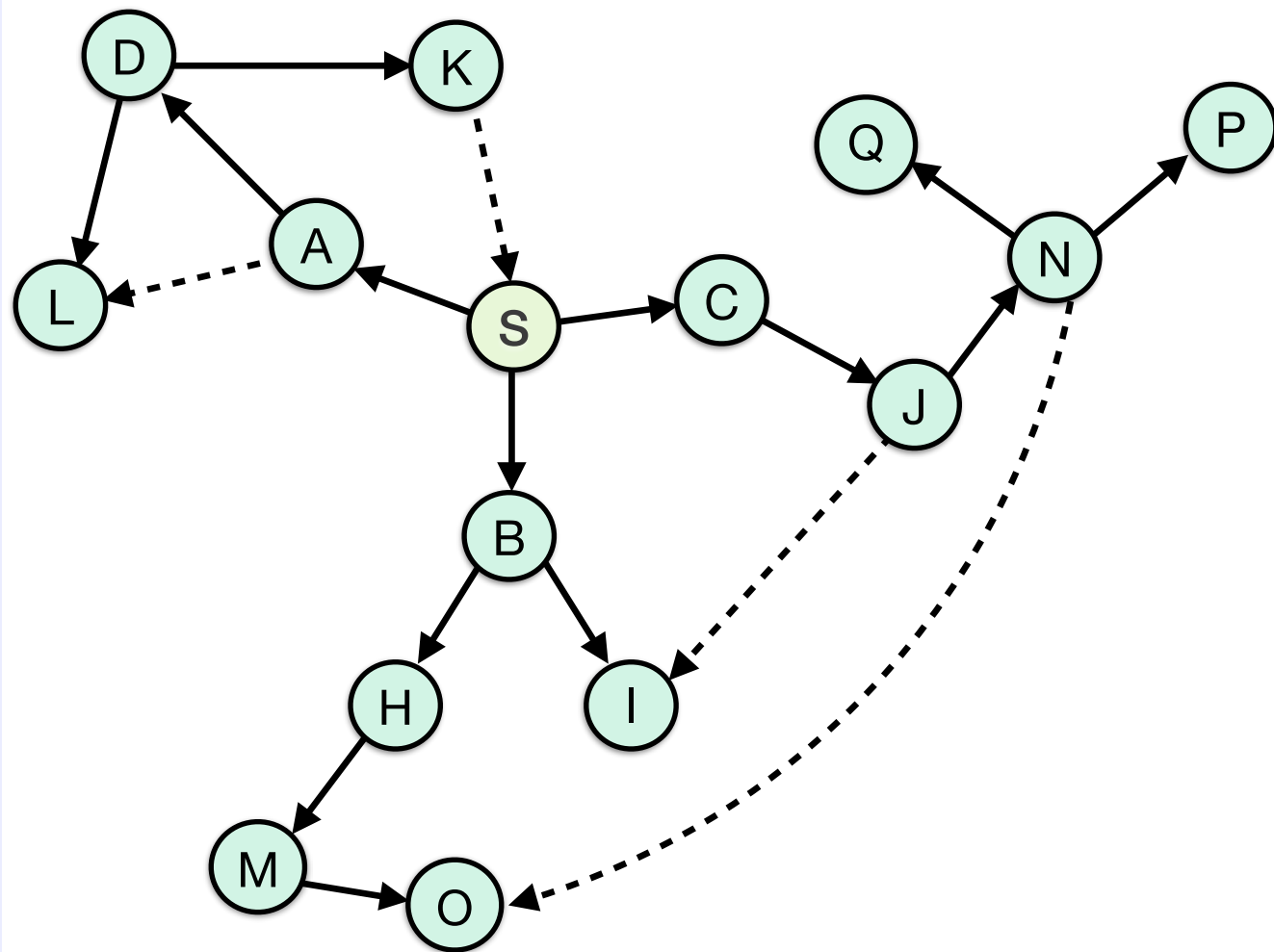
Algorithm:

1. For each $v \in V$:
 $visited(v) = false$;
2. Invoke DFS(s);

DFS(x):

1. Set $visited(x) = true$;
2. For each $y \in N(x)$:
If ($visited(y) = false$):
Invoke DFS(y)

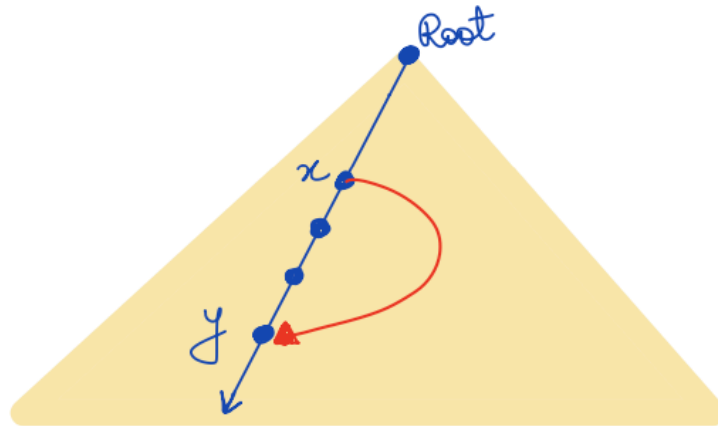
What if G is directed?



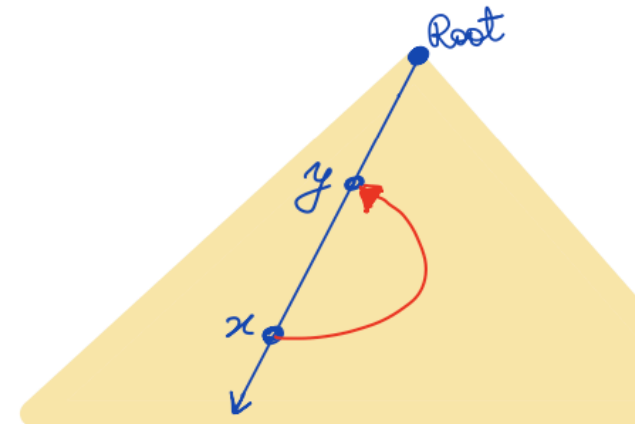
DFS tree
of a directed graph (on left)

Classification of non-tree edges

Forward Edges

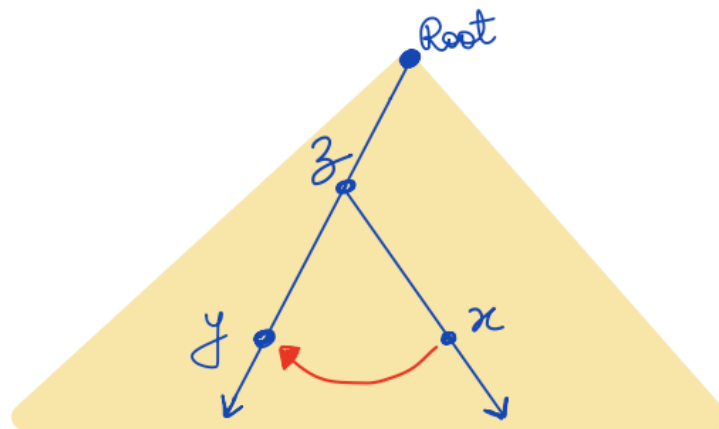


Back Edges

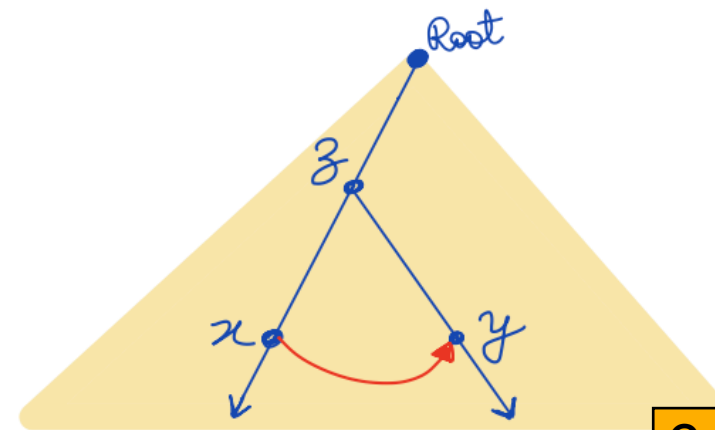


Cannot lie in
DFS tree
of
directed
acyclic
graph

Cross Edges



Anti-Cross Edges



Cannot lie in
DFS tree

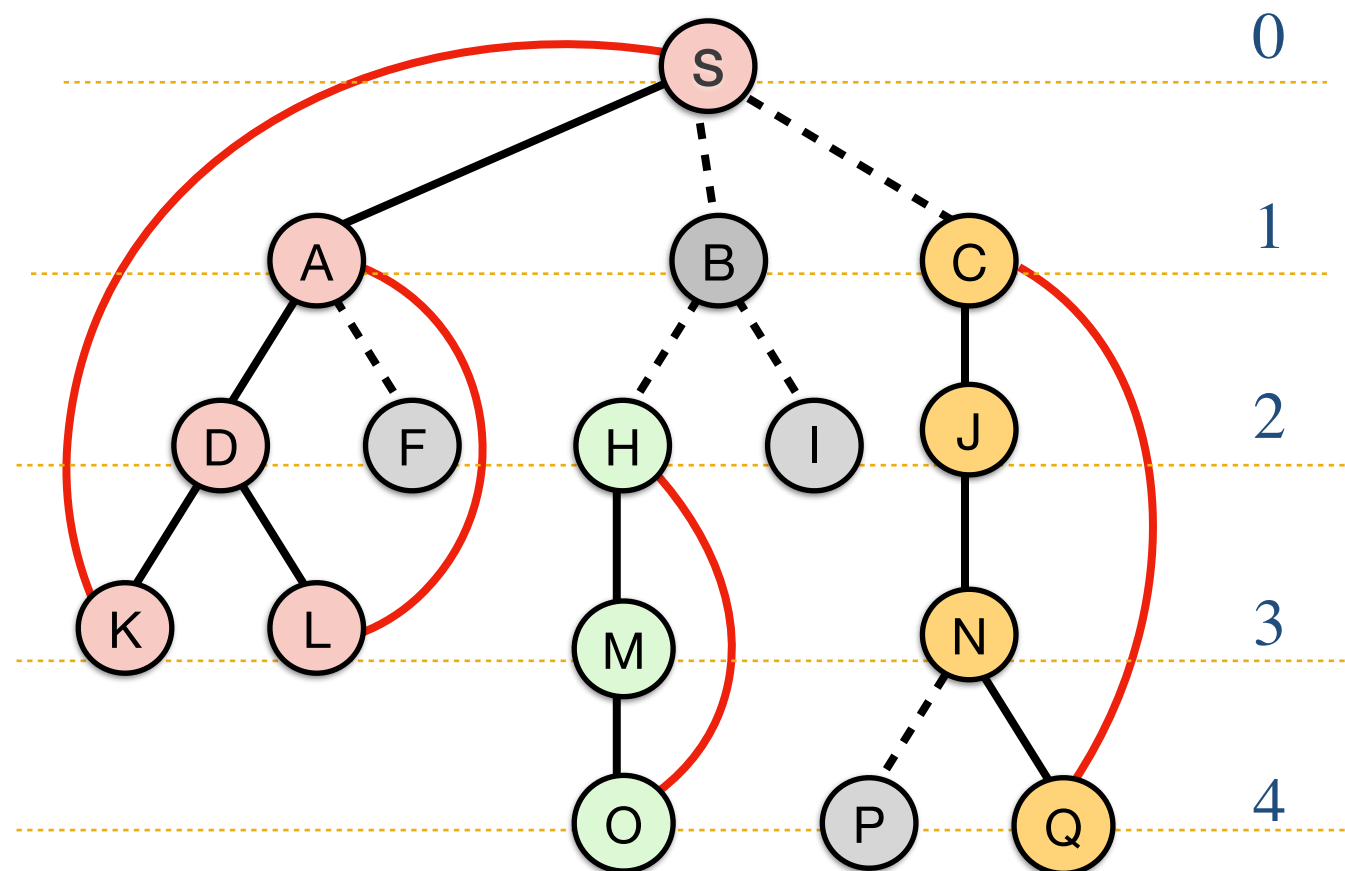
Homework Question

Two vertices x, y in an **undirected graph** G are said to be **bi-connected** if there exists **two edge-disjoint-paths** from x to y .

Use DFS to design an algorithm to find all vertex-pairs that are bi-connected.

Hint:

Two vertices x, y are **bi-connected** if and only if the path from x to y in the DFS tree does not contain any Bridge edges. (See Example)



Homework Question

Let G be a **directed acyclic graph**.

Use DFS to compute an **ordering**, $v_1, \dots, v_n$ of $V(G)$ satisfying the following:

for any edge $(v_i, v_j) \in E(G)$, we have $i < j$

