CS-208:Artificial Intelligence

Topic-05: Uninformed Searching Methods

Depth First Searching Technique(DFS)

Principle: Pursue a single branch of the tree until it yields to a solution (reaching the goal node) or until some *pre-specified depth* has been reached, only then go back and explore the other branches.

In this search, the other alternatives at the same level are ignored completely as long as there is a hope of reaching the goal node using the original choice.

In some problem the DFS through a tree would slip past the level at which goal node appears and waste incredible energy in exhaustively exploring parts of the lower down. For this type of problems the DFS is the worst possible approach.

DFS Algorithm

Step-1: Form single element Stack that initially consist of the root node representing the starting state.

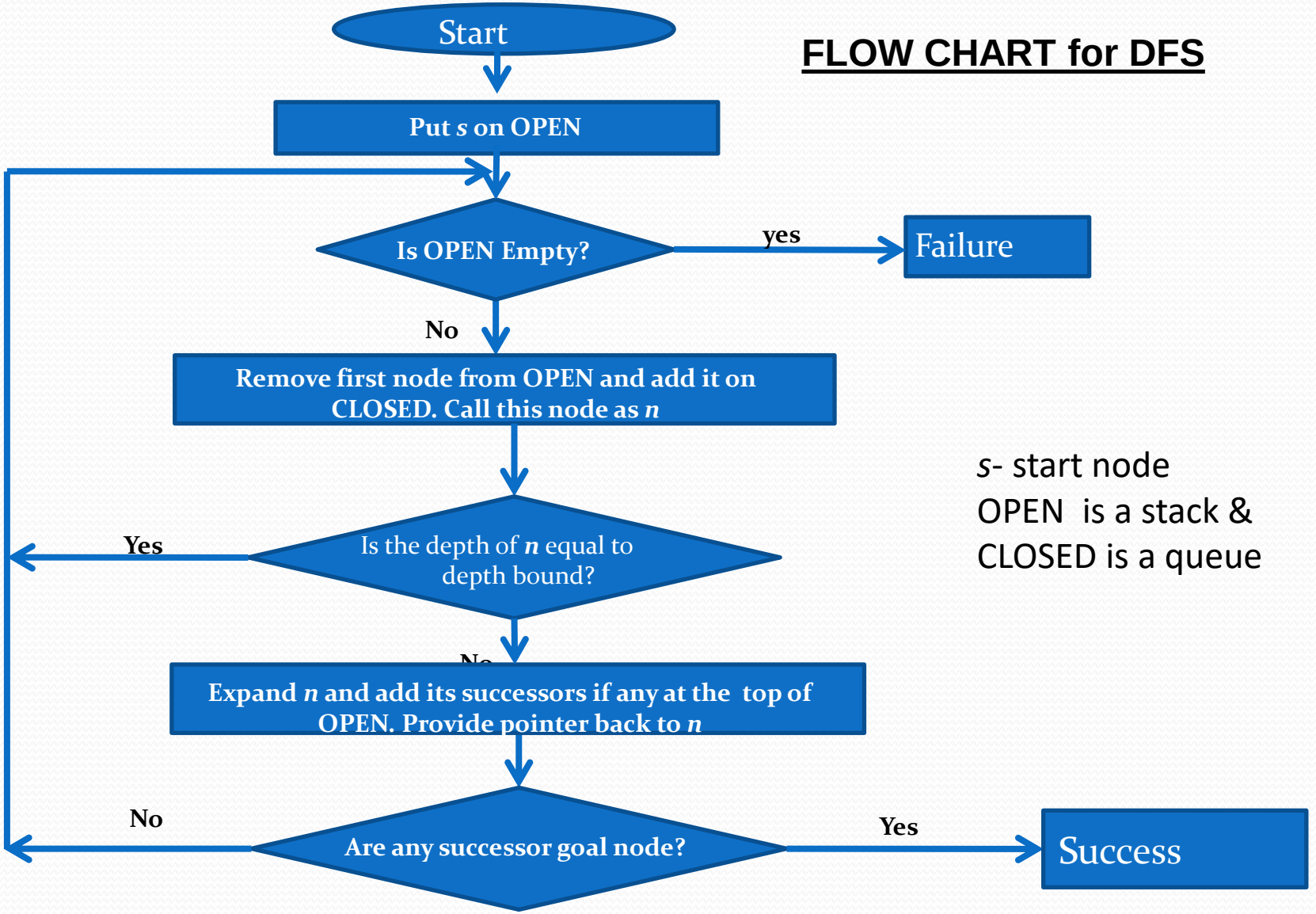
Step-2: Repeat the following until the stack becomes empty or the goal node has been reached:

a. If the element at the top of the stack is the goal node then “do nothing”

b. If the top element is not the goal node, then remove the element at the top of the stack and add its successors if any to the stack

Step-3: If the goal node has been found then announce **SUCCES**S, otherwise **FAILURE**

FLOW CHART for DFS



s- start node
OPEN is a stack &
CLOSED is a queue

Breadth First Searching Technique(BFS)

Principle: All the nodes on one level of the tree are examined before examining any other node in the next level.

Two major merits of BFS:

1. BFS is guaranteed to find the goal node if one exist, provided there are finite number of branches in the tree
2. BFS will also find the goal node with shortest path from the starting node.

Three major demerits of BFS:

1. It requires lot of memory to store the nodes because the number of nodes at each level increases exponentially with level number and all of them must be stored.
2. It may require a lot of work , when the shortest path is too long and the number of nodes increases exponentially with the length of the path.
3. Irrelevant or redundant operators(production rules) will greatly increases the number nodes that must be explored.

BFS Algorithm

Step-1: Form single element queue that initially consist of the root node representing the starting state.

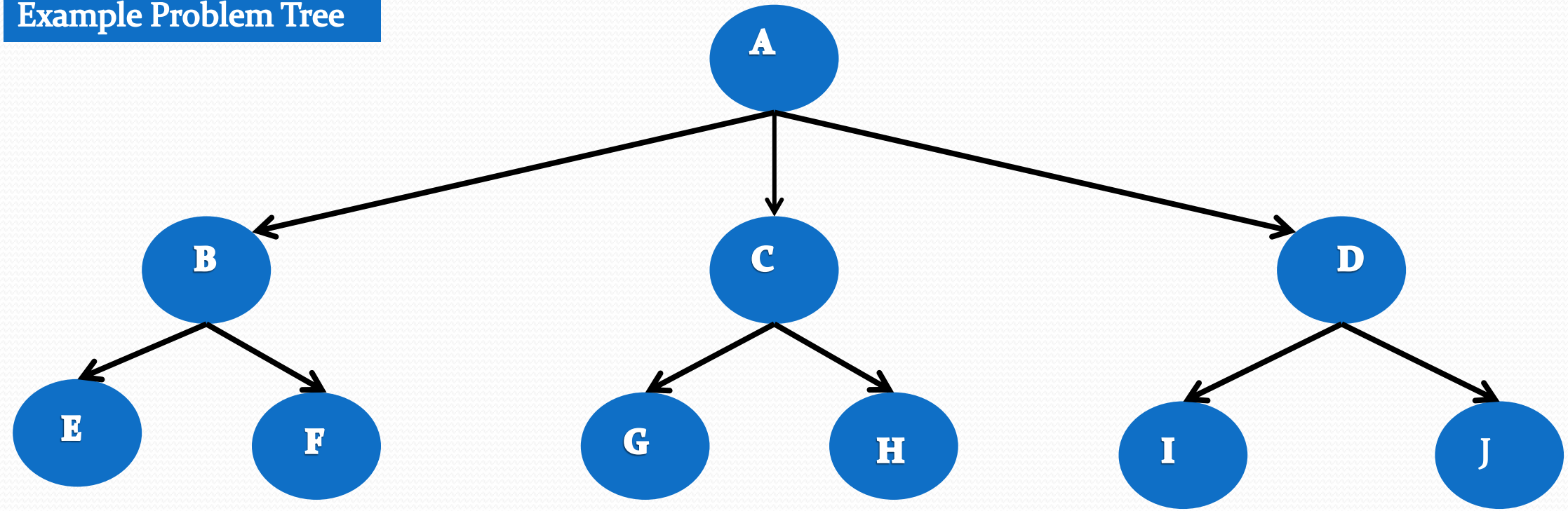
Step-2: Repeat the following until the queue becomes empty or the goal node has been reached:

a. If first element (the element at the front of the queue) is the goal node then “do nothing”

b. If the first element is not the goal node, then remove the first element from the queue and add its successors if any to the rear(end) of the queue

Step-3: If the goal node has been found then announce **SUCCESS**, otherwise **FAILURE**

Example Problem Tree



Path followed by DFS: *A B E F C G H D I J*

Path followed by BFS: *A B C D E F G H I J*

Informed and Uninformed Methods

A searching method is **informed**, if it uses additional information about nodes that have not yet been explored to decide which node to examine next.

If the search method is not informed then it is **uninformed or blind.**

Examples:

Hill Climbing and Best First Search are *Informed search methods.*

Depth First Search and Breadth First Search are *Uninformed search methods*.