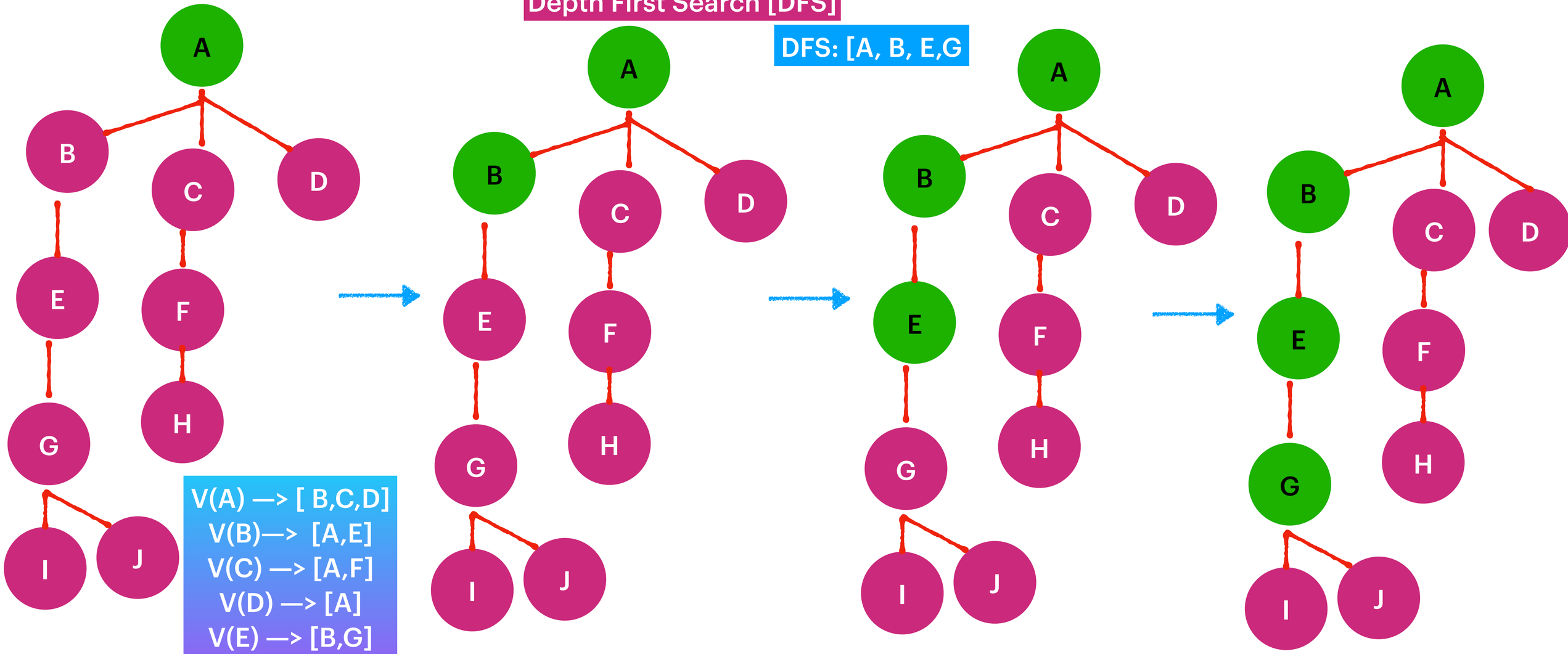
Depth First Search [DFS]

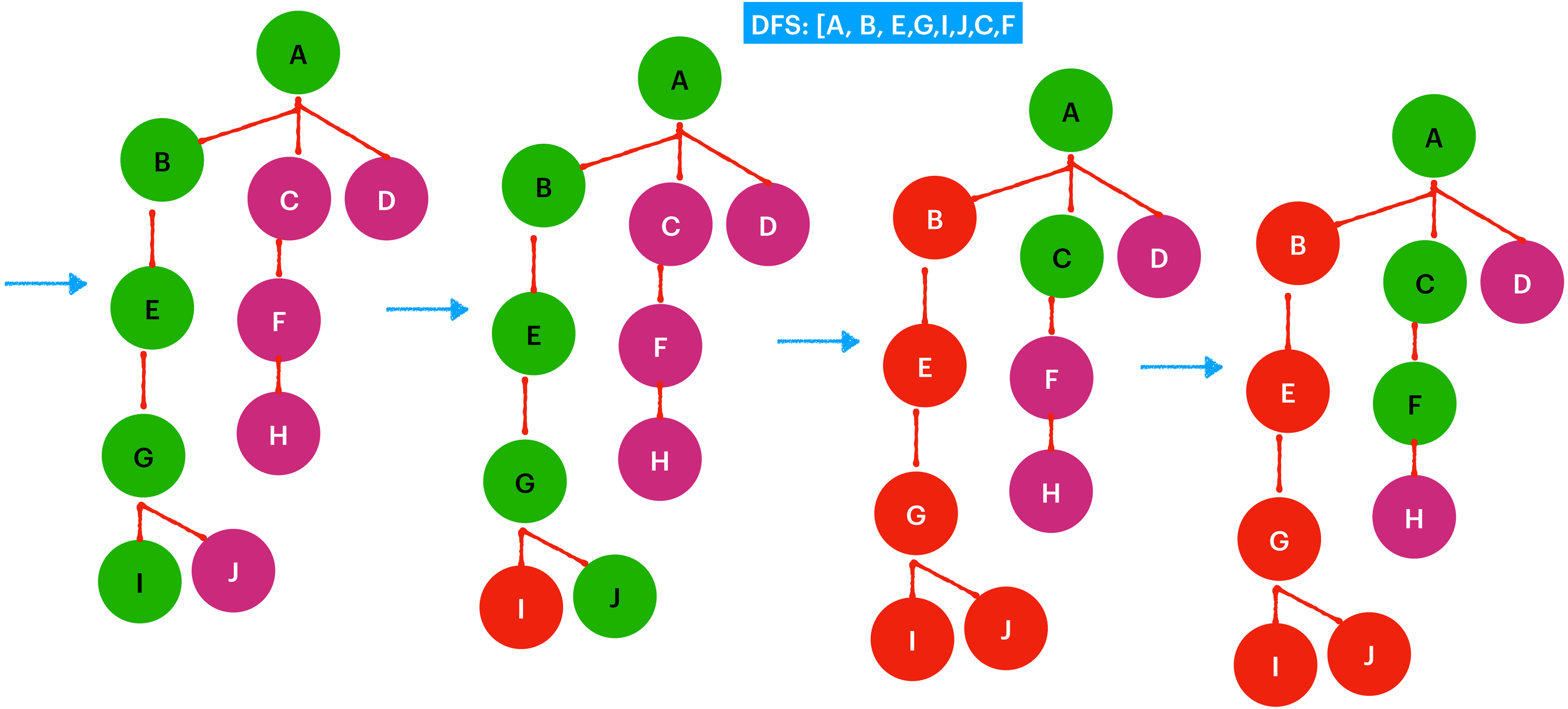
DFS: [A, B, E,G



V(A) —> [B,C,D]
V(B)—> [A,E]
V(C) —> [A,F]
V(D) —> [A]
V(E) —> [B,G]
V[F] —> [C,H]
V(G) —> [E,I]
V(H) —> [F]
V(I) —> [G]
V(J) —> [G]

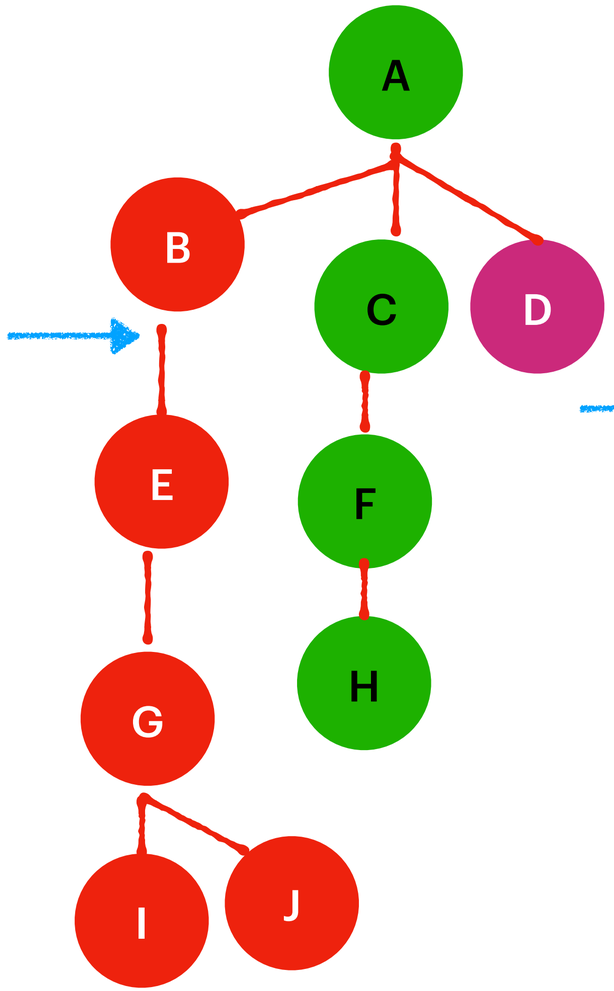
Depth First Search [DFS]

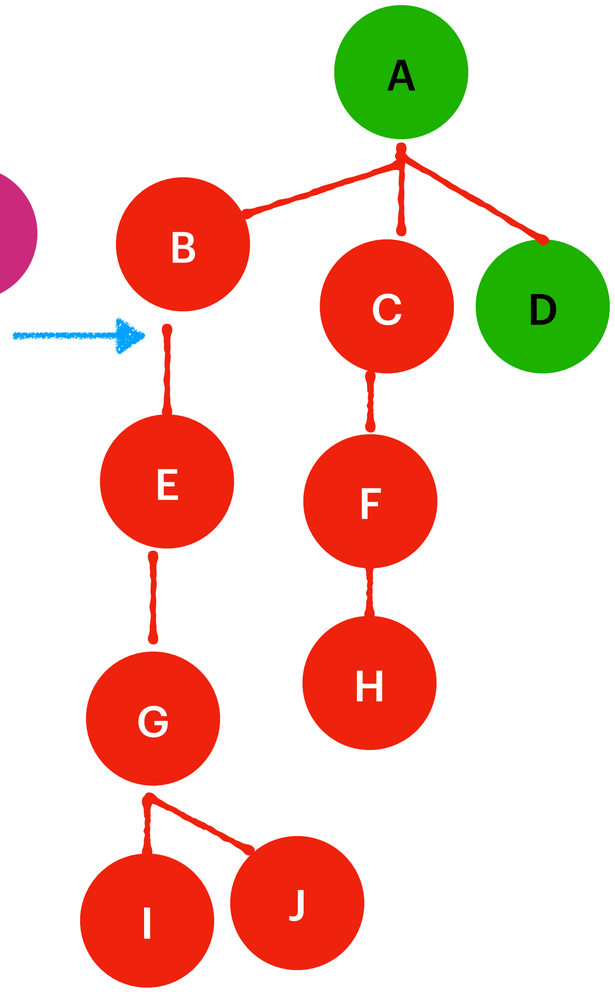
DFS: [A, B, E,G,I,J,C,F

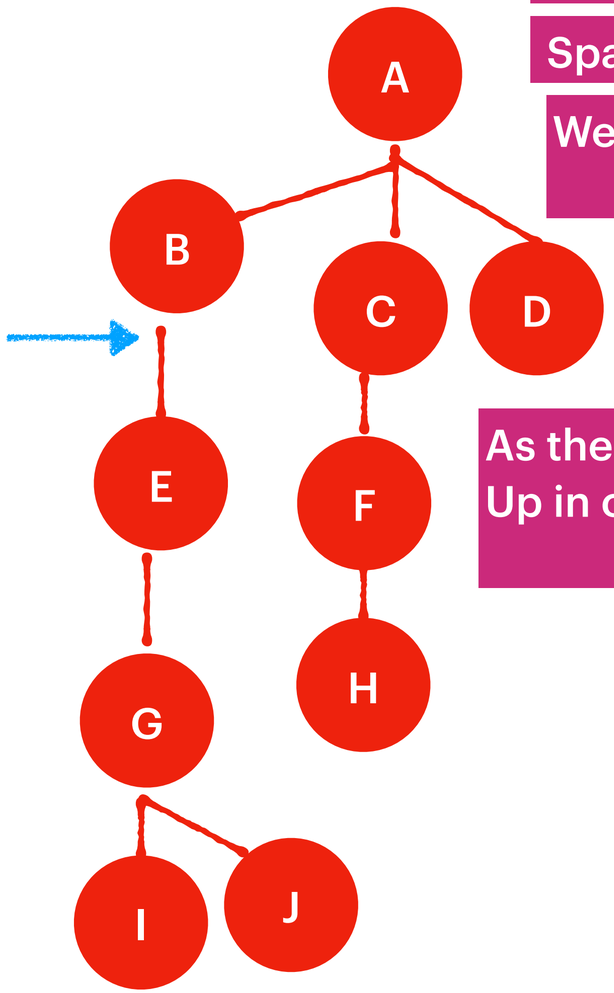


Depth First Search [DFS]

DFS: [A, B, E,G,I,J,C,F,H,D]







Time Complexity : O(V + E)

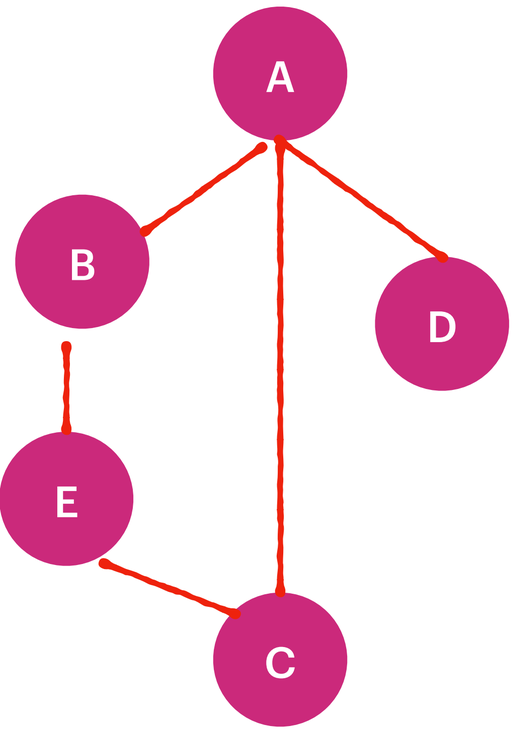
Space Complexity : O(V)

We maintain, HashSet to track The visiting vertices.

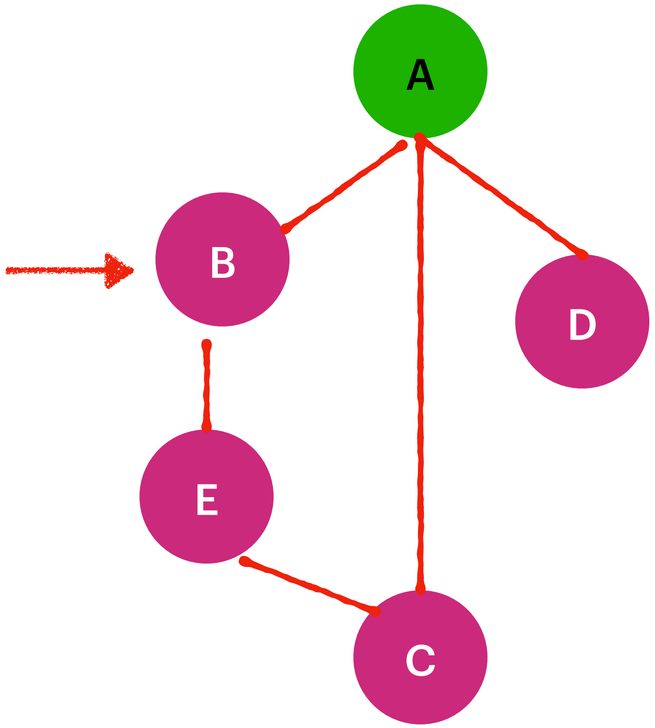
As the Graph is Bidirectional, we end Up in cyclic loop if we don't maintian Visited notes in the Set.

Depth First Search [DFS]

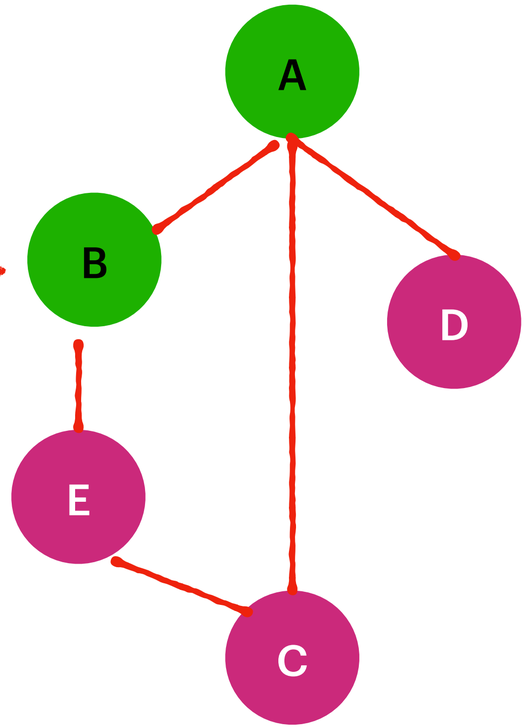
DFS : []
visitedSet: []
Stack: [A]



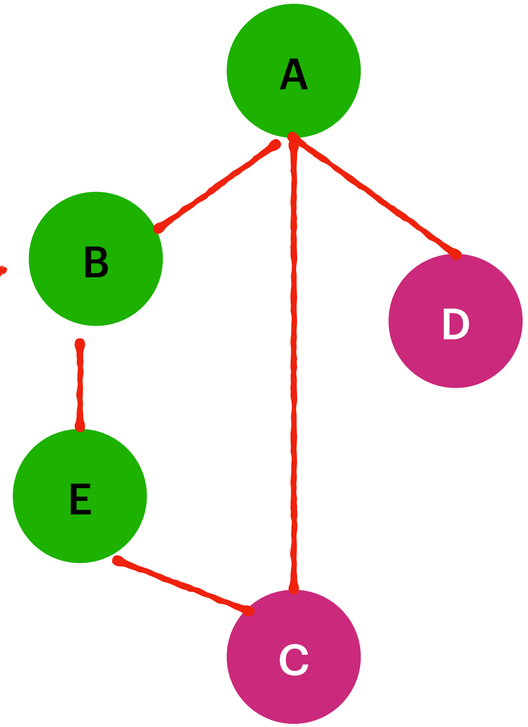
DFS : [A]
visitedSet: [A]
Stack: [D <— C <— B(Top)]



DFS : [A,B]
visitedSet: [A,B]
Stack: [D <— C <— E(top)]



DFS : [A,B,E]
visitedSet: [A,B,E]
Stack: [D <— C <— C(top)]



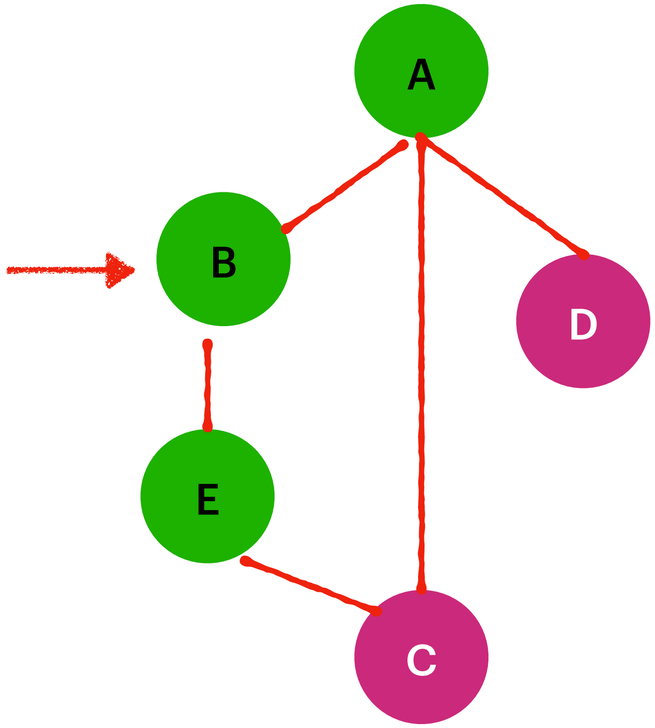
Algorithm

If the Stack is not Empty.

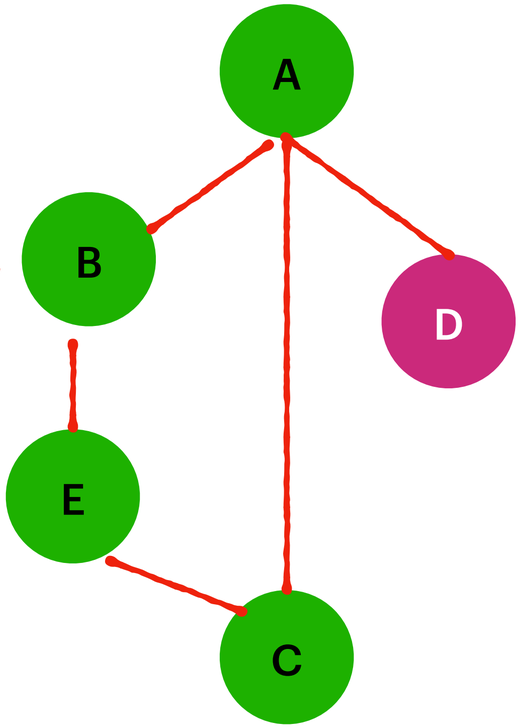
1. Pop the vertex from Stack.
2. Add vertex to the DFSList if its not Visited.
3. Mark the vertex as Visited.
4. Add the vertex connections to the Stack If they are not visited.

Depth First Search [DFS]

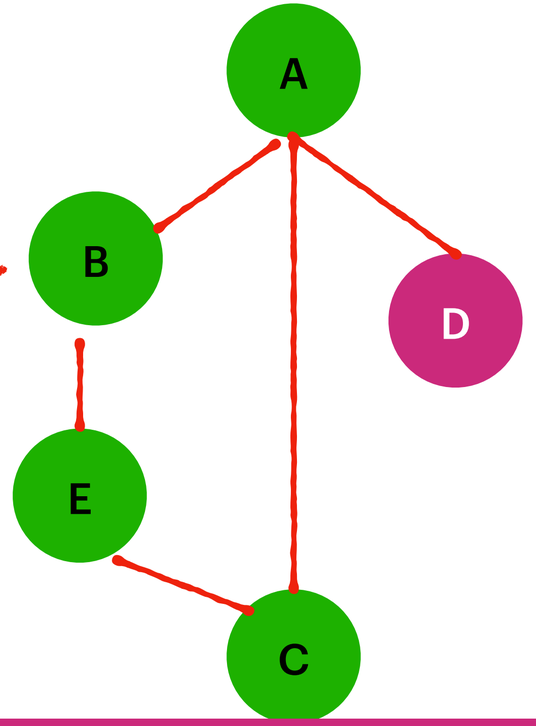
DFS : [A,B,E]
visitedSet: [A,B,E]
Stack: [D <— C <— C(top)]



DFS : [A,B,E,C]
visitedSet: [A,B,E,C]
Stack: [D <— C(top)]



DFS : [A,B,E,C]
visitedSet: [A,B,E,C]
Stack: [D]

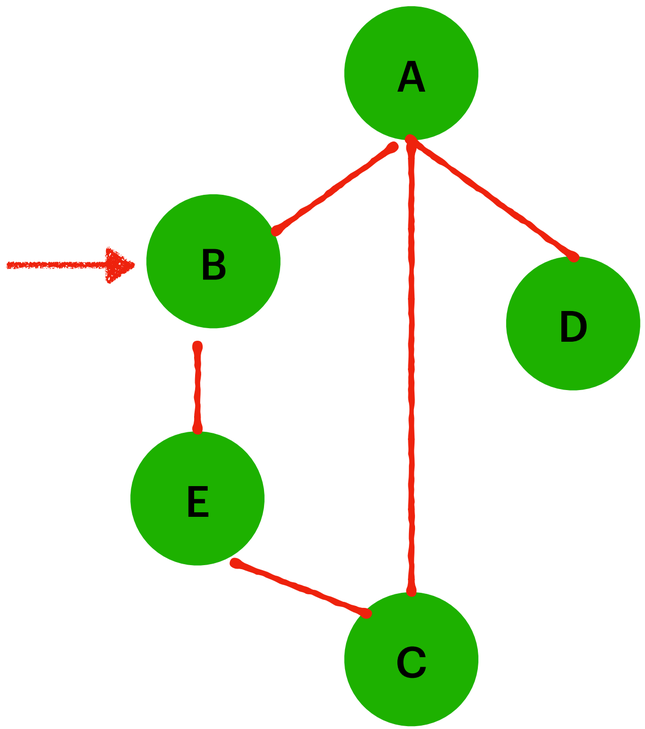


If the Stack is not Empty.

1. Pop the top vertex from Stack.
2. Add popped Vertex connections to the Stack If they are not visited.
3. Mark the popped vertex as Visited.

Current Stack top is V(c),
C is already visited in the previous Step,
so don’t add to DFSList

DFS : [A,B,E,C,D]
visitedSet: [A,B,E,C,D]
Stack: []



Stack is Empty
So DFS is —>
DFS : [A,B,E,C,D]

Time Complexity : O(V + E)

Space Complexity : O(V)

We maintain, HashSet to track
The visiting vertices.

As the Graph is Bidirectional, we end
Up in cyclic loop if we don't maintian
Visited vertices in the Set.

If the Stack is not Empty.

1. Pop the top vertex from Stack.
2. Add popped Vertex connections to the Stack If they are not visited.
3. Mark the popped vertex as Visited.