

Depth first search (DFS)

→ Traverse graph in depthward motion.

→ Uses stack to remember to get the next vertex.

steps:

- ① Visit adjacent unvisited vertex, mark it as visited, display it, push it in a stack.
- ② If no adjacent vertex found, pop up a vertex from stack.
- ③ Repeat step ① & step ②, until stack is empty.

Routine

```
void DFS (vertex v)
```

```
{  
    visited [v] = True;
```

```
    for each w adjacent to v
```

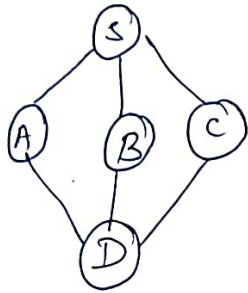
```
        if (! visited [w])
```

```
            DFS (w);
```

```
}
```

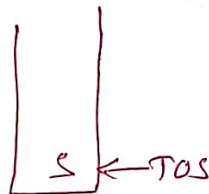
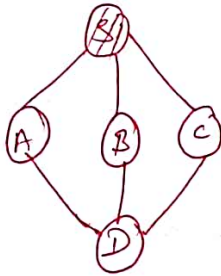
Example:

① Initialize the stack



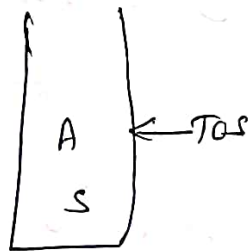
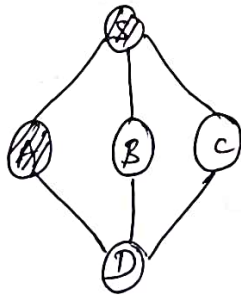
stack

② Mark 'S' as visited & put it onto the stack, explore any unvisited adjacent node from 'S'.



stack

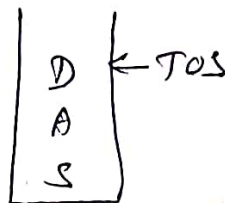
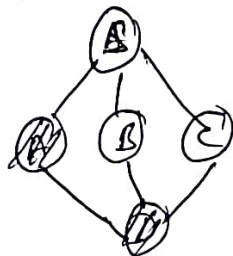
③ Mark 'A' as visited and put it onto the stack, explore any unvisited adjacent node from 'A'.



stack

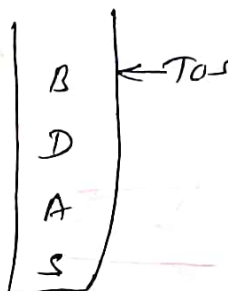
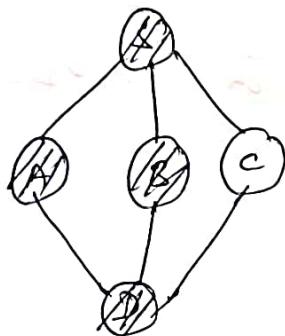
④ Mark 'D' as visited & put it onto the stack.

Here we have 'B' and 'C' nodes which are adjacent to D both are unvisited.



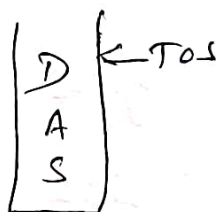
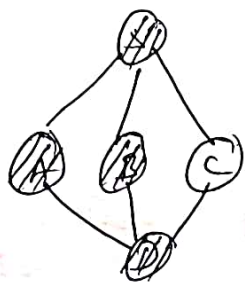
stack

5. Chosen 'B' marked as visited & put it onto the stack. 'B' does not have any unvisited node. So we pop 'B' from stack.

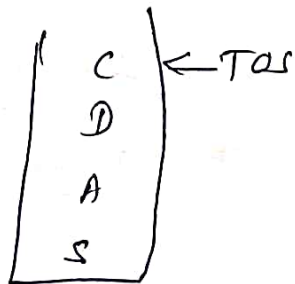
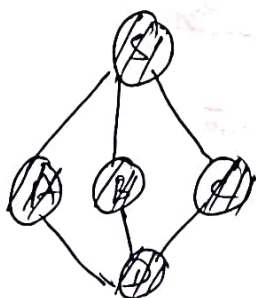


stack

6. We check stack top for return to previous node and check if it has any unvisited node, Here find 'D' to be on top of stack.



7. Only unvisited adjacent node is 'C' from D. Visit and place it on stack.



* 'C' does not have unvisited nodes. So we keep popping the stack until we find a node stack is empty.