Algorithm: Topological Sort (**Tological Ordering. This can be done DFS only**)
Input: A simple connected directed graph G = (V,E)
Output: An array with topologically sorted vertices from 1 to n.

```
Initialize a stack S
TopSort[1..n]                  //Arrays are initialize to 0
Mark[1..n]                     //n the number of vertices in our class notes
index ← n
Pick a starting vertex s and Mark[s] ← 1
S.push(s)
while S ≠ ∅ do
    v  ← S.peek()
    if some vertex adjacent to v not yet visited then
        w ← next unvisited vertex adjacent to v
        Mark[w] ← 1
        S.push(w)
    else
        TopSort[index--] ←S.pop()
// DFS completed
return TopSort
```
