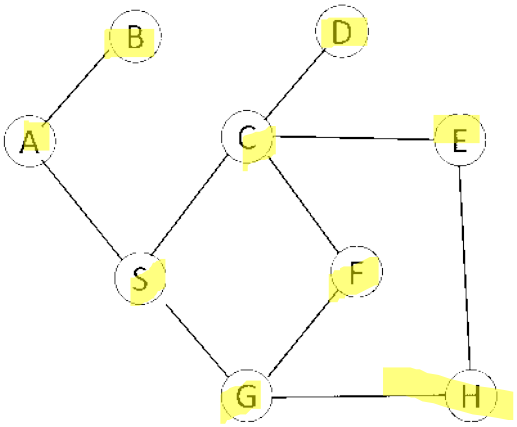
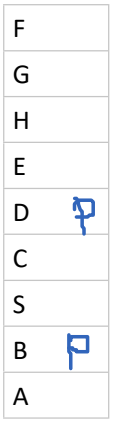
Depth-First Search

Thursday, May 21, 2020 9:24 PM

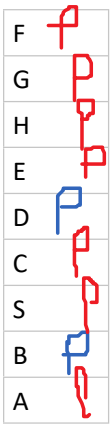
DFS is similar to tree preorder traversal - Goes as far as possible from a vertex before backing up.
DFS uses stack to keep track the status of nodes (Last In First Out)





Last

First



Output (visited nodes)

A B S C D E H G F

Starting step 12

Notes:

- *For the purpose of demonstration, we will follow the alphabet order when deciding which one should be first to select.*
- *To pop, instead of erasing a node, I marked **P** next to it so it would help to follow.*

1. We start at node A, highlight A, push A to stack; mark it as visited.
2. Node A connects B; push B to stack; mark it as visited.
3. Node B only connects to A; pop B out of stack;
4. Back to A, node A has S; push S to stack; mark it as visited.
5. Node S is top of stack; S connects node C; push C to the top of stack; mark it as visited.
6. Node C is top of stack; C connects node D; push D to the top of stack; mark it as visited.
7. Node D has no children; pop D out of stack; backtrack to node C
8. Node C connects to E; push E to stack; mark it as visited.
9. Node E connects to H; push H to stack; mark it as visited.
10. Node H connects to G; push G to stack; mark it as visited.
11. Node G connects to F; push F to stack; mark it as visited.
    Now
12. Node F is at the top of stack; all connected nodes were visited; pop F; backtrack to node G.
13. Node G is at the top of stack; all connected nodes were visited; pop G; backtrack to node H
14. Node H is at the top of stack; all connected nodes were visited; pop H; backtrack to node E
15. Node E is at the top of stack; all connected nodes were visited; pop E; backtrack to node C.
16. Node C is at the top of stack; all connected nodes were visited; pop C; backtrack to node S.
17. Node S is at the top of stack; all connected nodes were visited; pop S; backtrack to node A
18. Node A is at the top of stack; all connected nodes were visited; pop A; Stack is empty.

The search ends when the stack status is empty.