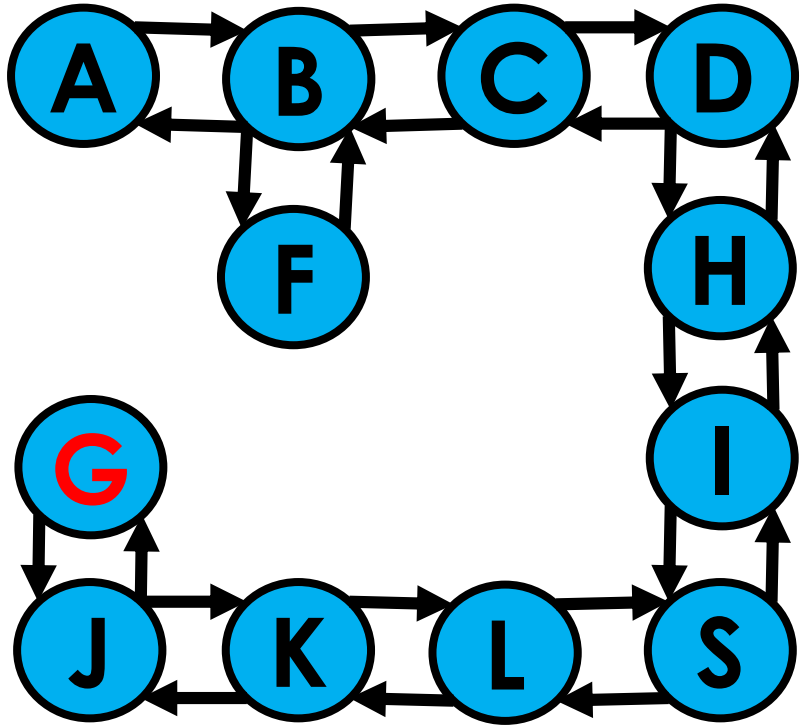
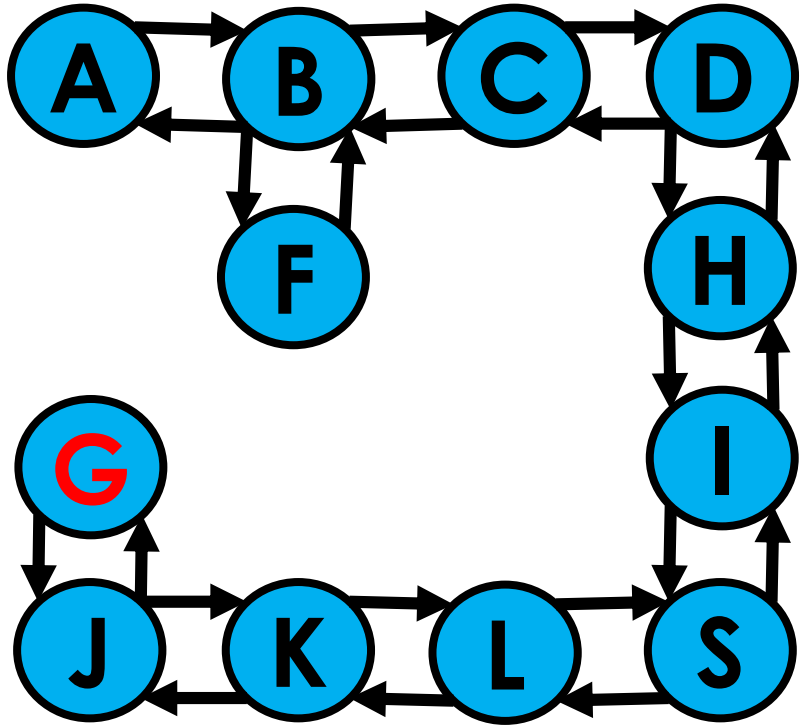
Graph Search

Part 1: Depth-first Search



Depth-first Search (DFS)

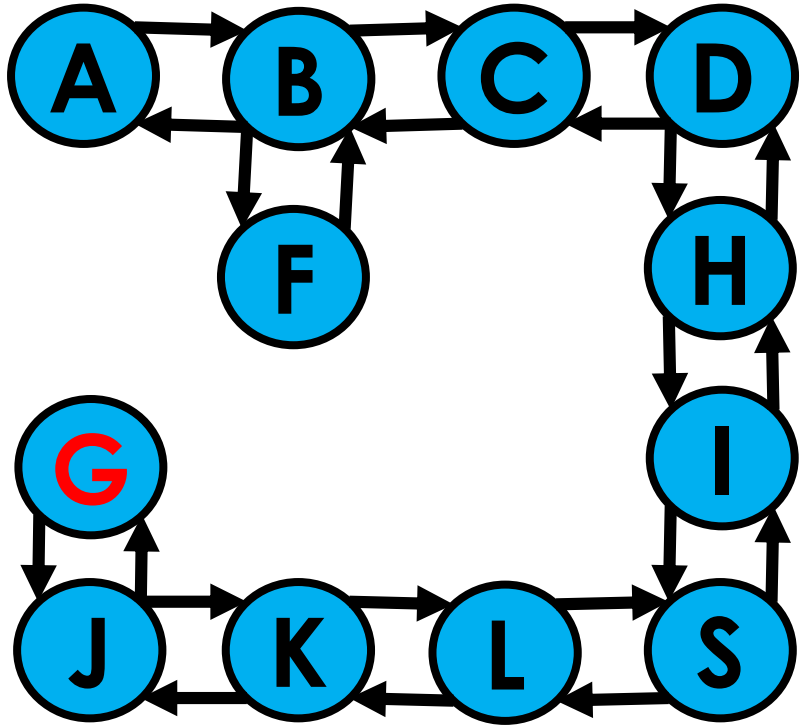
How to keep track of where to search next?
How to keep track of what's been visited?
How to keep track of the path from start to goal?



Depth-first Search (DFS)

How to keep track of where to search next?

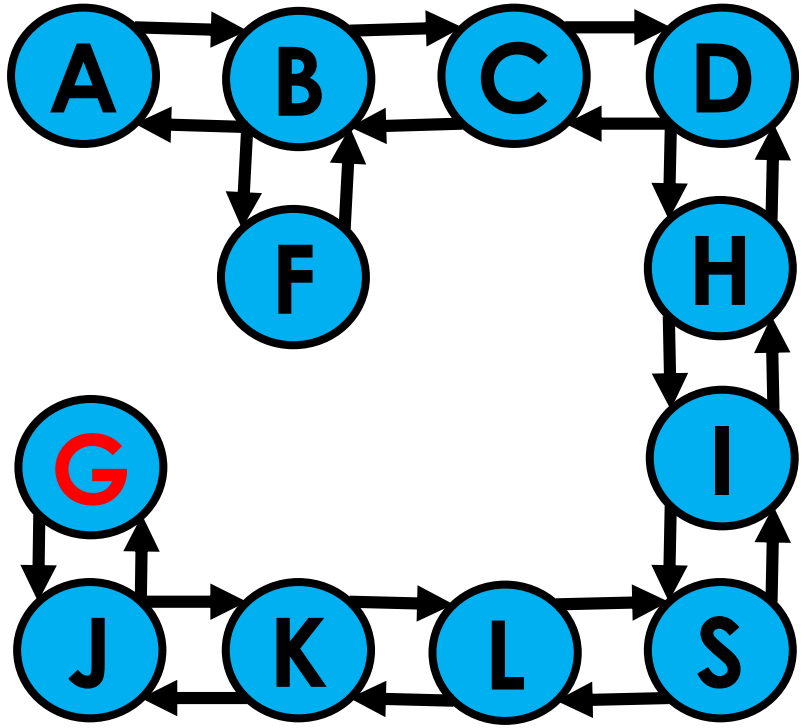
Stack: List where you add and remove from one end only:
push → add an element
pop → remove an element



Depth-first Search (DFS)

How to keep track of what's been visited?

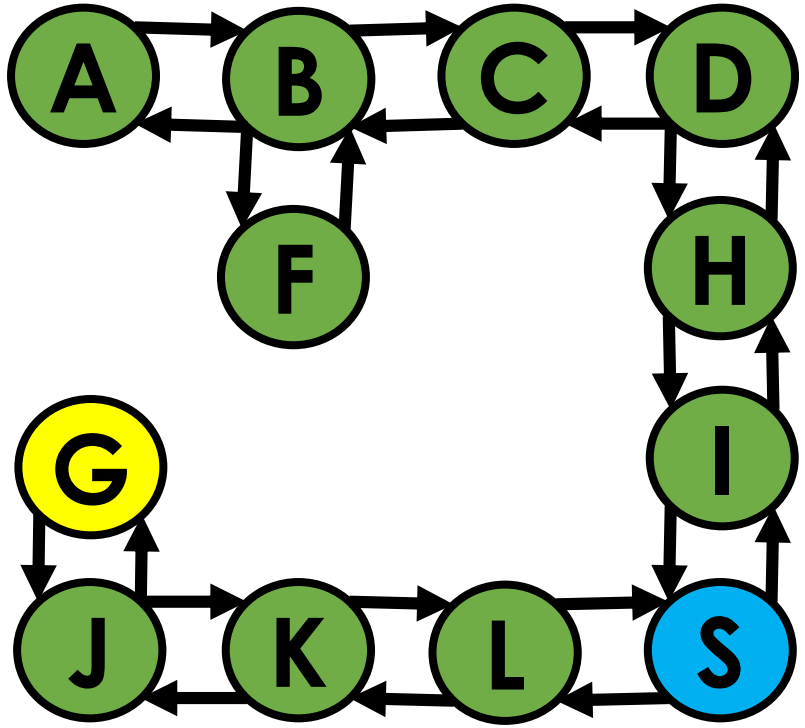
HashSet: Constant time add, remove, and find



Depth-first Search (DFS)

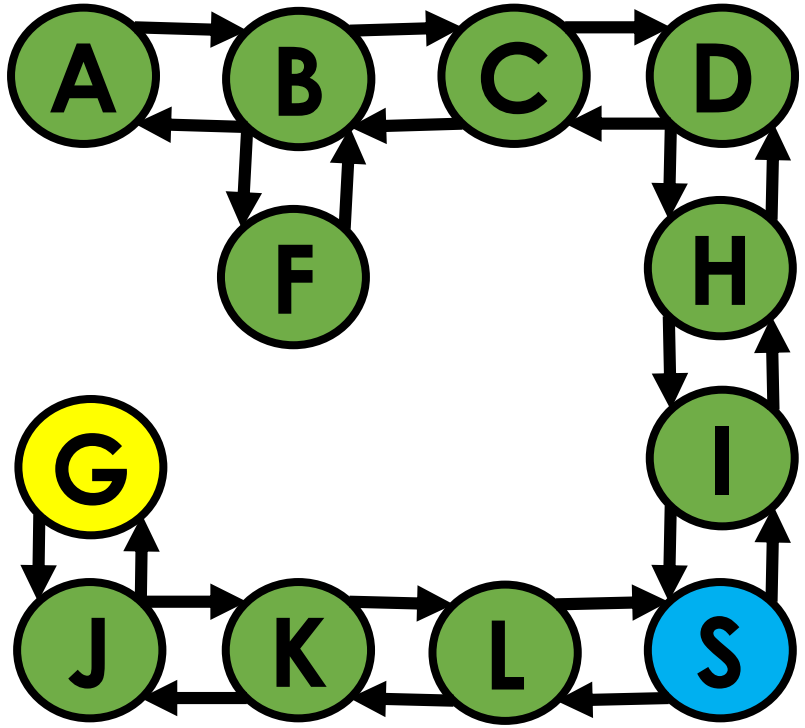
How to keep track of the path from start to goal?

HashMap: Link each node to the node from which it was discovered



DFS: Algorithm

```
DFS(S, G):
    Initialize: stack, visited HashSet and parent HashMap
    Push S onto the stack and add to visited
    while stack is not empty:
        pop node curr from top of stack
        if curr == G return parent map
        for each of curr's neighbors, n, not in visited set:
            add n to visited set
            add curr as n's parent in parent map
            push n onto the stack
    // If we get here then there's no path
```



DFS: Algorithm (recursive)

```
DFS(S, G, visited, parents):
    if S == G return;
    for each of S's neighbors, n, not in visited set:
       add n to visited set
       add S as n's parent in parents map
       DFS(n, G, visited, parents)
```