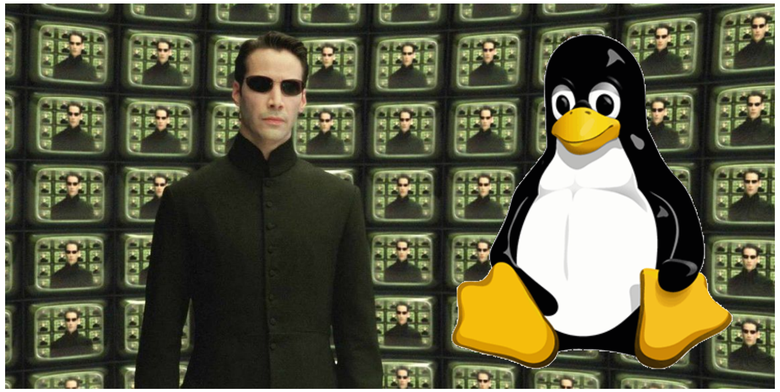
30.07.20 17:33

CSCI 125
Introduction to Computer Science and Programming II
Lecture 7: Data Structure IV



Jetic Gū

2020 Summer Semester (S2)

Some changes

- Assignment 4 and Lab 4 due 9 Aug, covering Lecture 7
- No class next Monday (BC day)
- Last batch of OJ problem: 6 problems p025-p030

Admin

Overview

- Focus: Data Structures
- Architecture: Linux/Unix OS
- Core Ideas:
    1. Breath-First Search VS Depth-First Search

P1
Searching

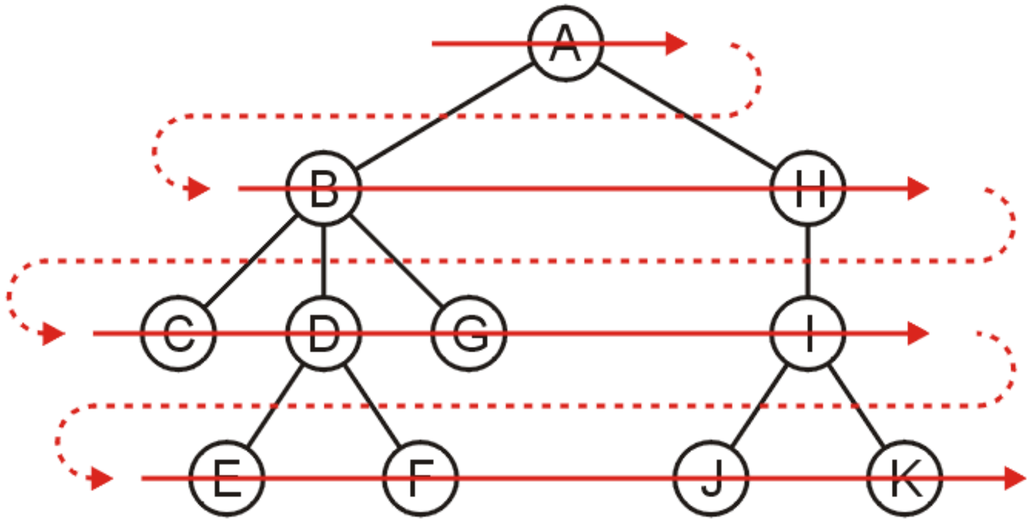
Search Algorithms

Breath-First Search VS Depth-First Search

Summary

Breath-First Search

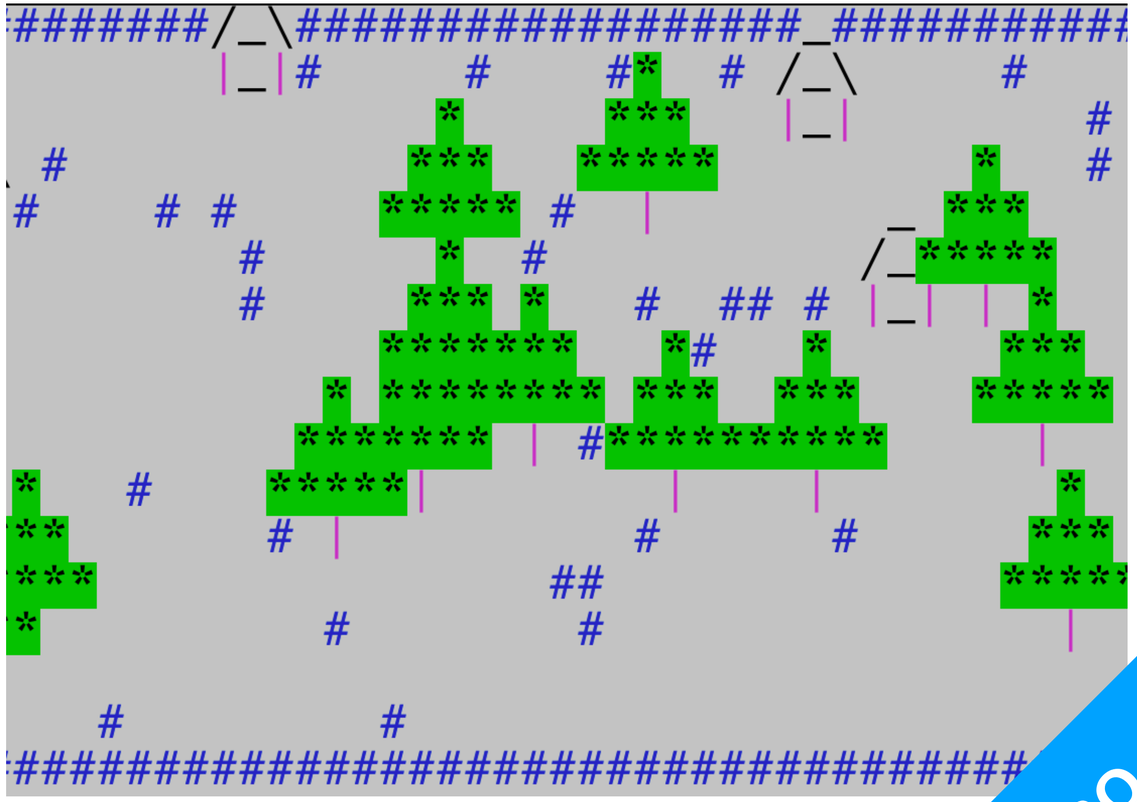
- Algorithm for traversing or searching
- Visit all nodes on the same depth, before moving on
  - Implementation using Queue
  - Works on trees and graphs



Review

BFS Example

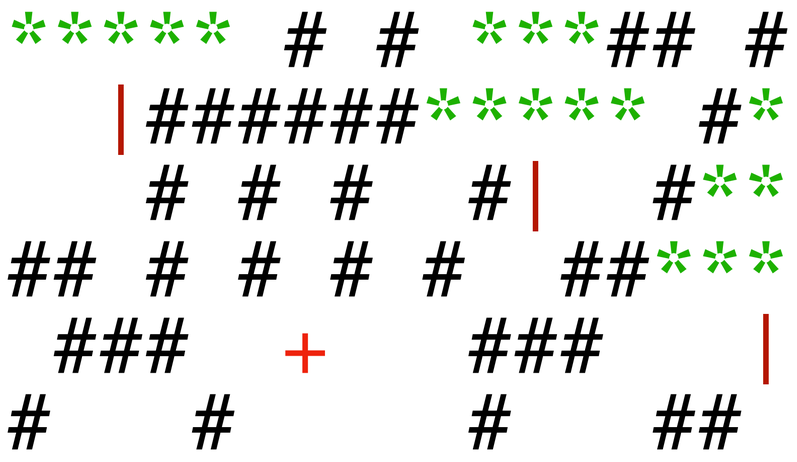
- Village of the Sorcerer
  - Certain areas of the map are not reachable (fences, houses)
  - Certain areas might be entirely blocked off by fences



Demo

BFS Example

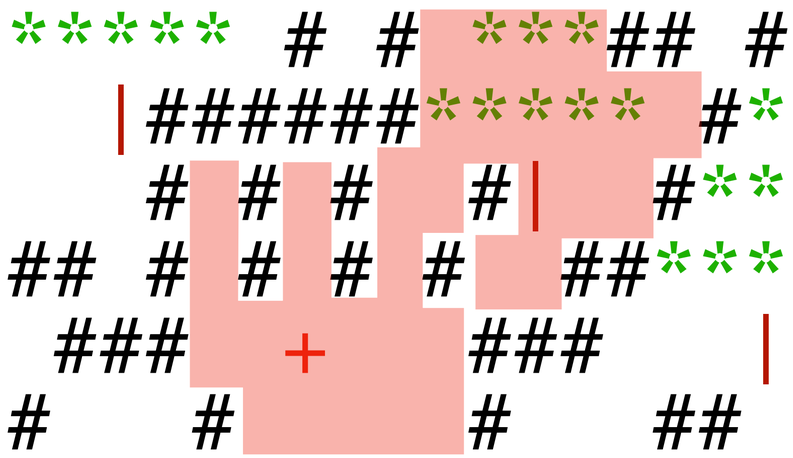
- Village of the Sorcerer
  - Certain areas of the map are not reachable (fences, houses)
  - Certain areas might be entirely blocked off by fences
  - +: player



Demo

BFS Example

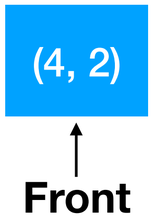
- Village of the Sorcerer
  - Certain areas of the map are not reachable (fences, houses)
  - Certain areas might be entirely blocked off by fences
  - +: player
  - Light Red: player accessible regions



Demo

BFS Example

- We start by initialising a Queue to store coordinates
- push(playerCoordinate), which is 4,2
  - `push((4,2));`
  - In C++, you can simulate this using 2 `int`-based queues



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

(4, 2)

↑

Front

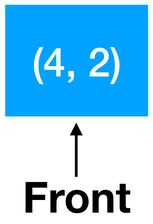
Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

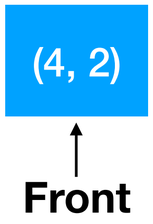


Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,2);`



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,2);`
  - `push(4,1); push(4,3); push(5,2)`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,2);`
  - `push(4,1); push(4,3); push(5,2)`
  - `pop();`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,1);`

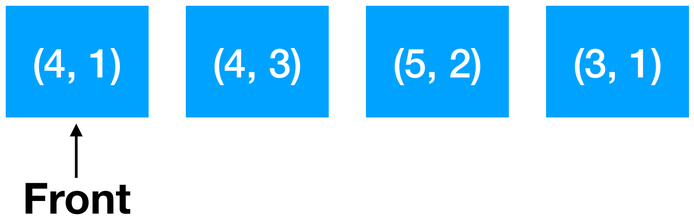
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,1);`
  - `push(3,1);`



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,1);`
  - `push(3,1);`
  - `pop();`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

P1
Searching

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,3);`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,3);`
  - `push(3,3);`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (4,3);`
  - `push(3,3);`
  - `pop();`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (5,2);`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (5,2);`
  - `no push here`

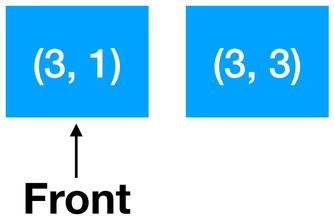
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (5,2);`
  - `no push here`
  - `pop();`



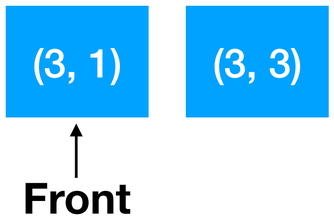
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	



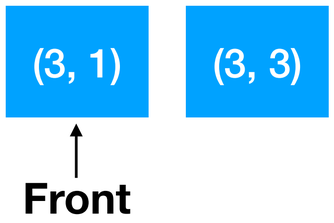
Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

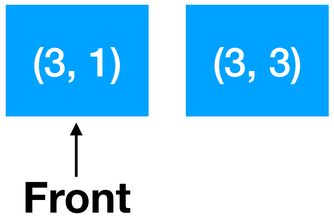


Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,1);`



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,1);`
  - `push(2,1);`

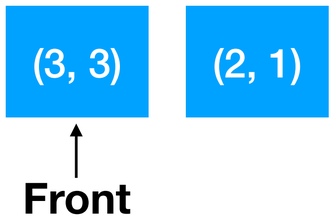
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,1);`
  - `push(2,1);`
  - `pop();`



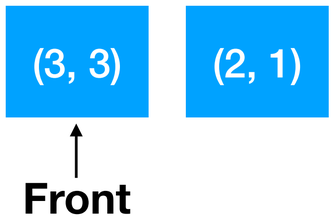
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	



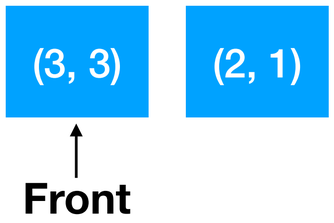
Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

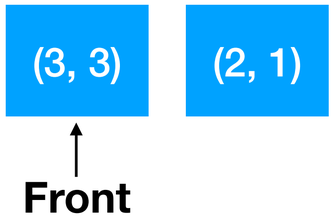


Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,3);`



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,3);`
  - `push(2,3);`

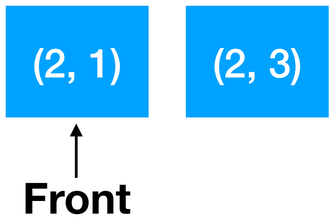
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (3,3);`
  - `push(2,3);`
  - `pop();`



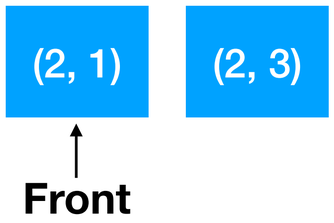
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

Demo

1. P030

BFS Example

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

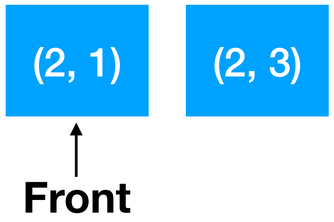


Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited



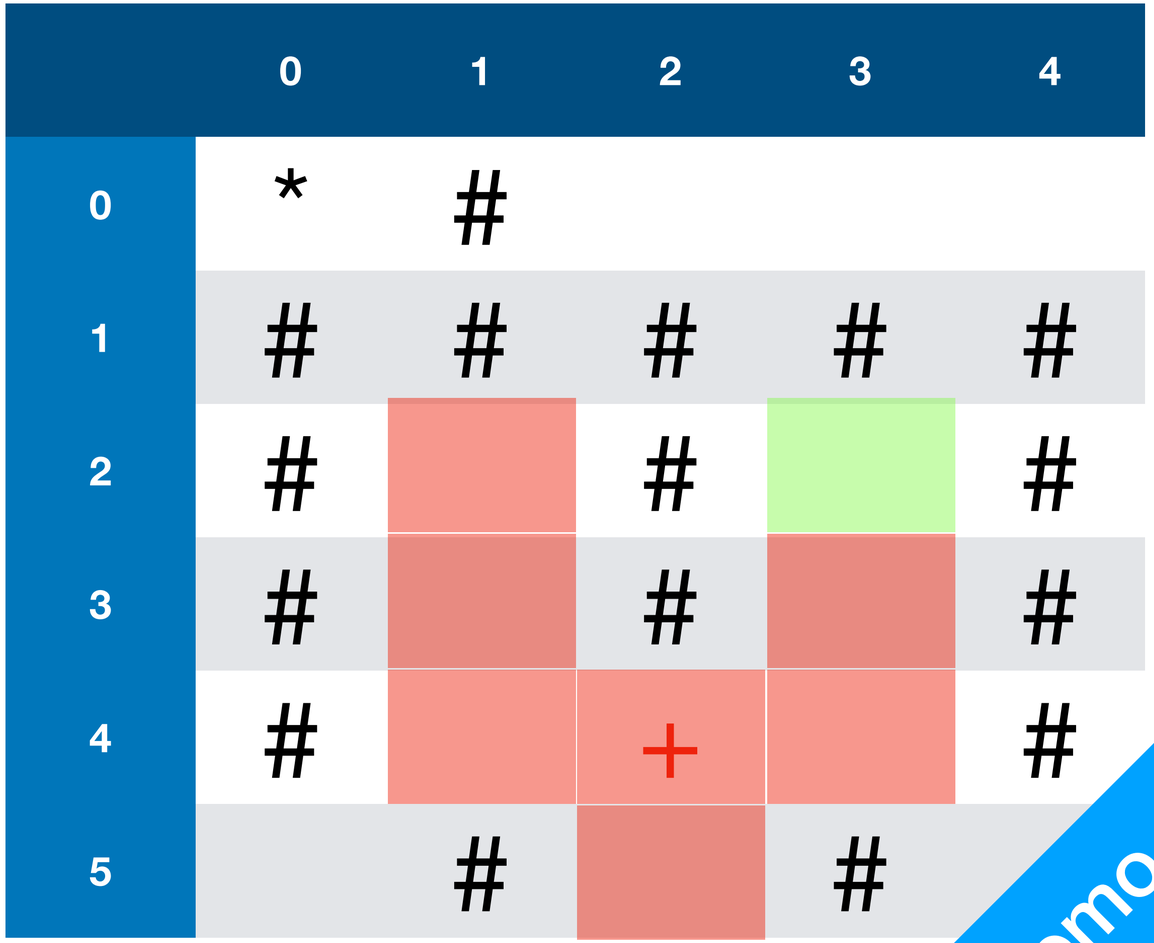
	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#		#		#
4	#		+		#
5		#		#	

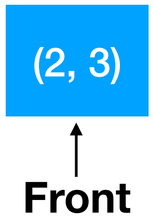
Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (2,1); pop();`



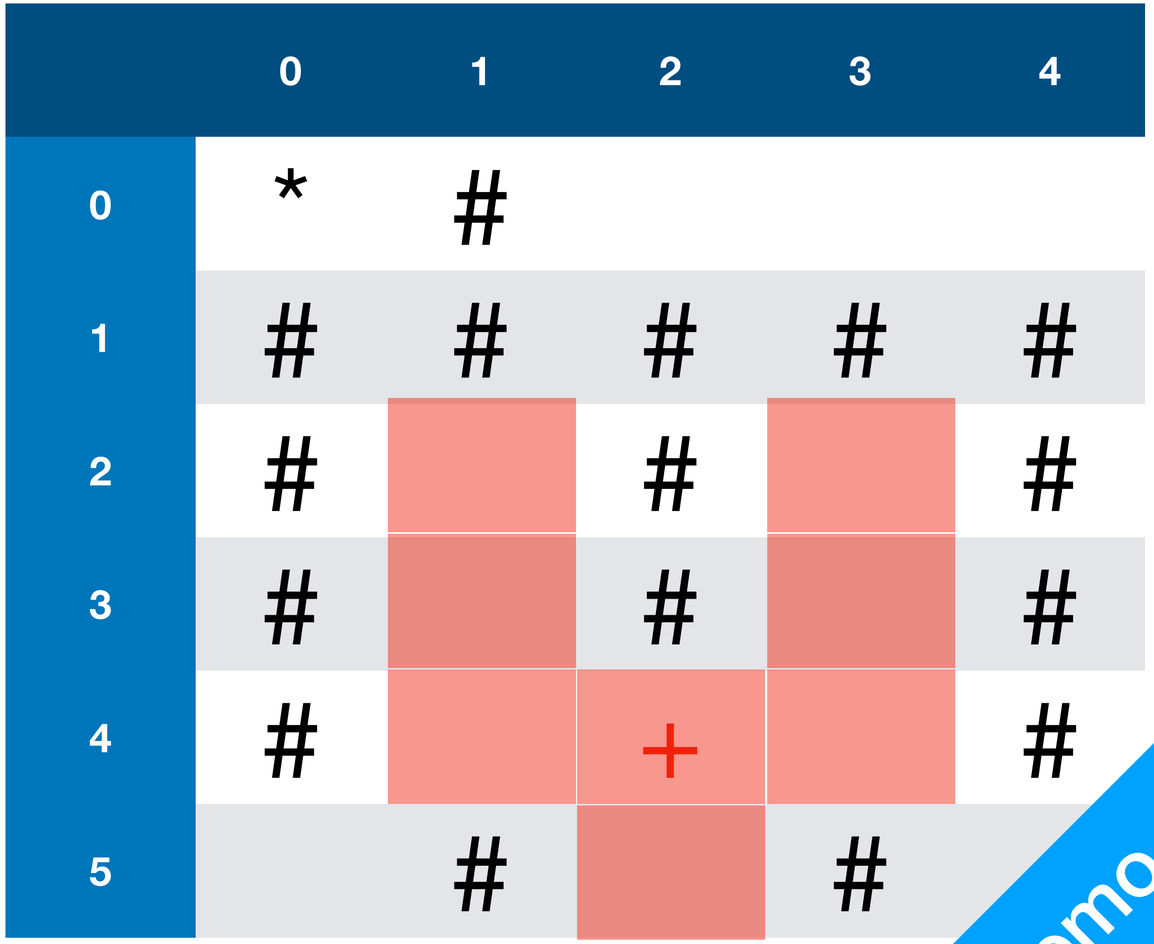


Demo

1. P030

BFS Example

- For every frontal element, push it's neighbouring reachable coordinates, and mark itself as visited
  - `front() = (2,1); pop();`
  - `front() = (2,3); pop();`



1. P030

BFS Example

Demo

BFS Example

- Assuming map size $N \times M$, what is the time and space complexity?

Demo

BFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$

Demo

BFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?

Demo

BFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?
  - Space complexity: $O(N \times M)$

Demo

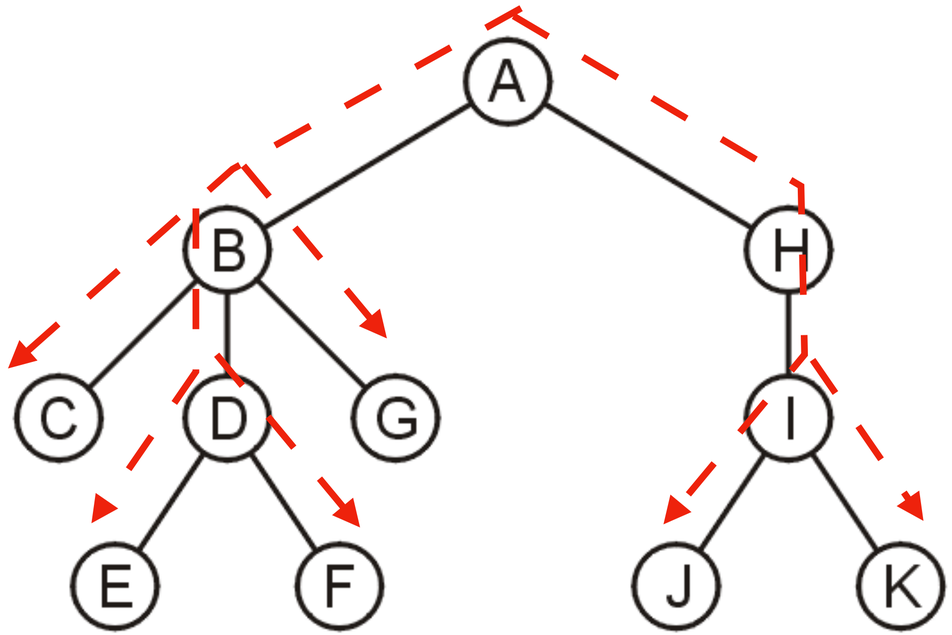
BFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?
  - Space complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?

Demo

Depth-First Search

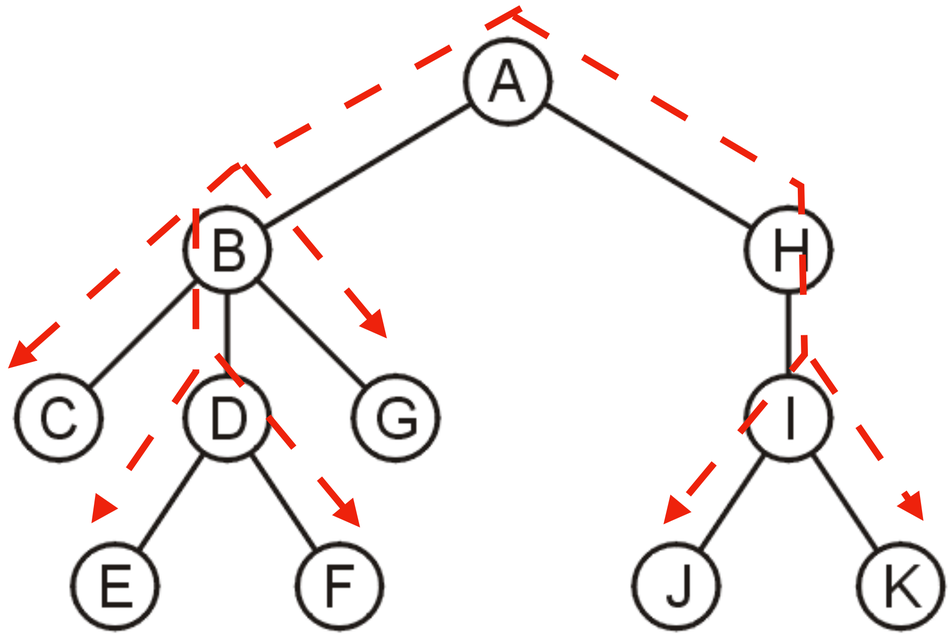
- Algorithm for traversing or searching
- Take a single route till the end, then look for other routes
  - Implementation using Stack
  - Works on trees and graphs



Review

Depth-First Search

- Depth-First Search (BFS)
  - Go all the way till there's no way forward, then go back to find another way
  - Classic searching algorithm



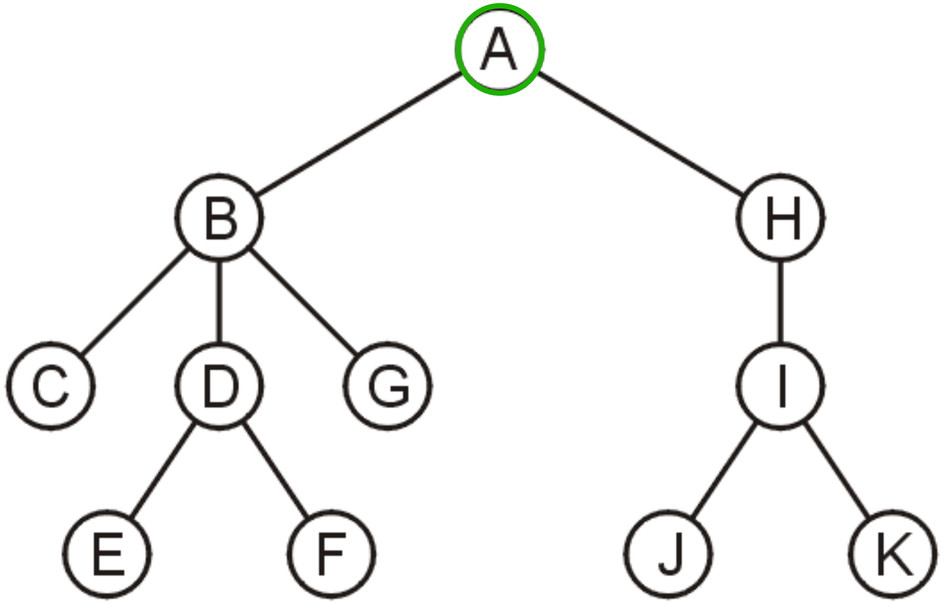
Review

P1
Searching

DFS Tree

- Push the root directory A

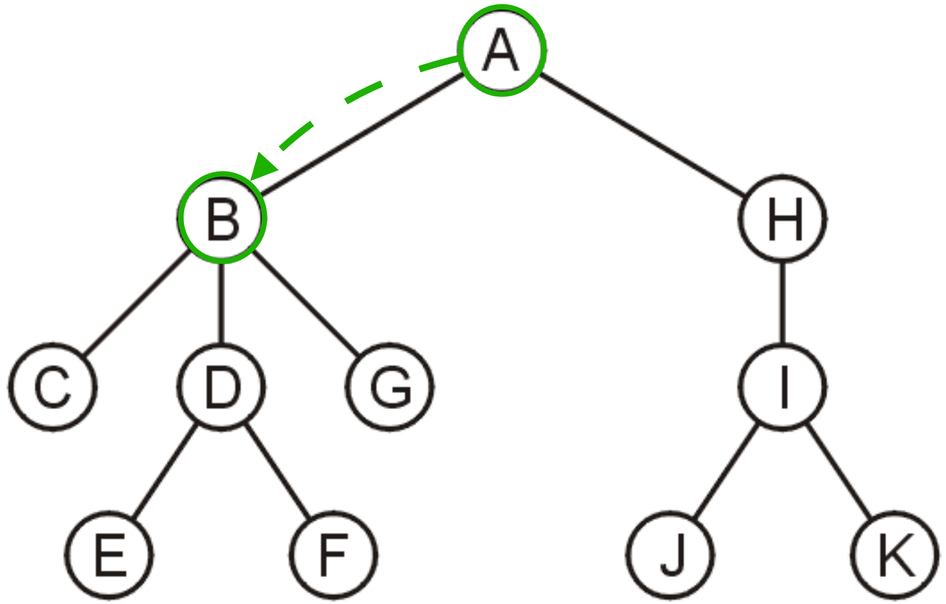
A



Demo

DFS Tree

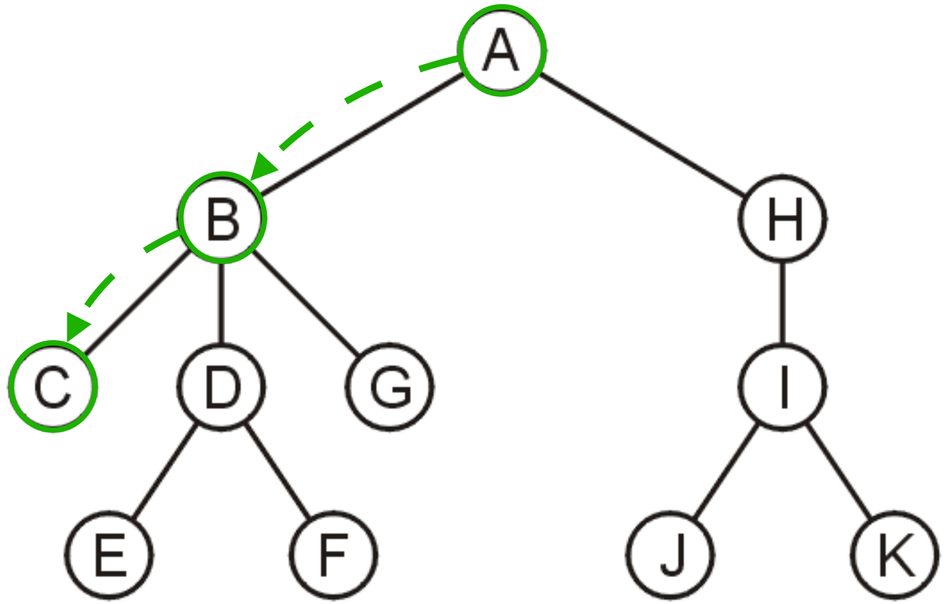
- Push B



Demo

DFS Tree

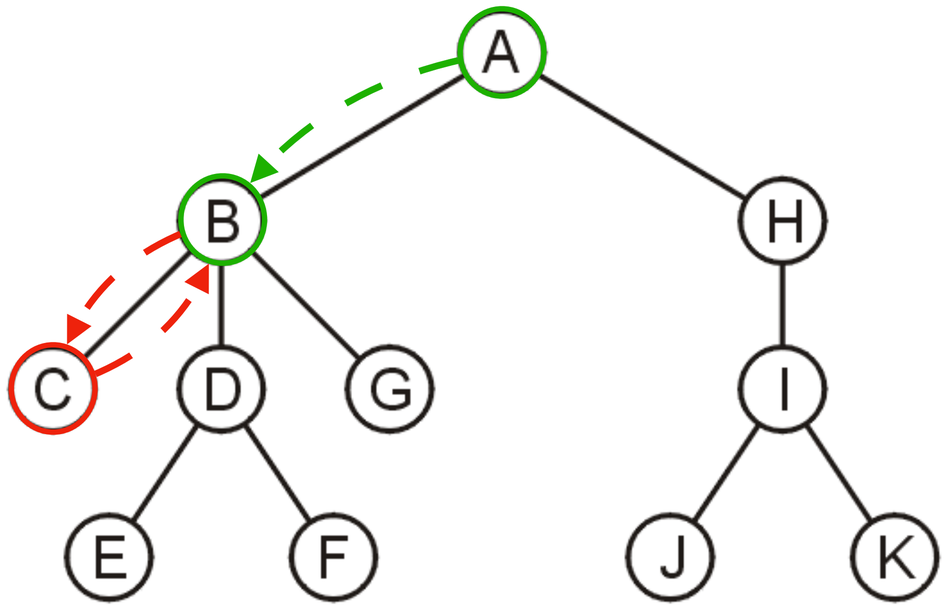
- Push C



Demo

DFS Tree

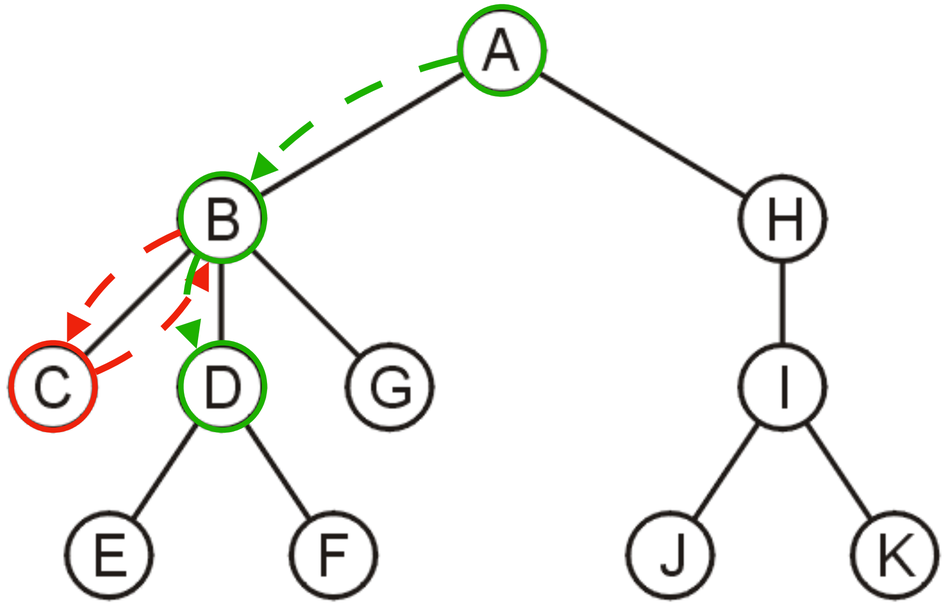
- Cannot go any further, pop C



Demo

DFS Tree

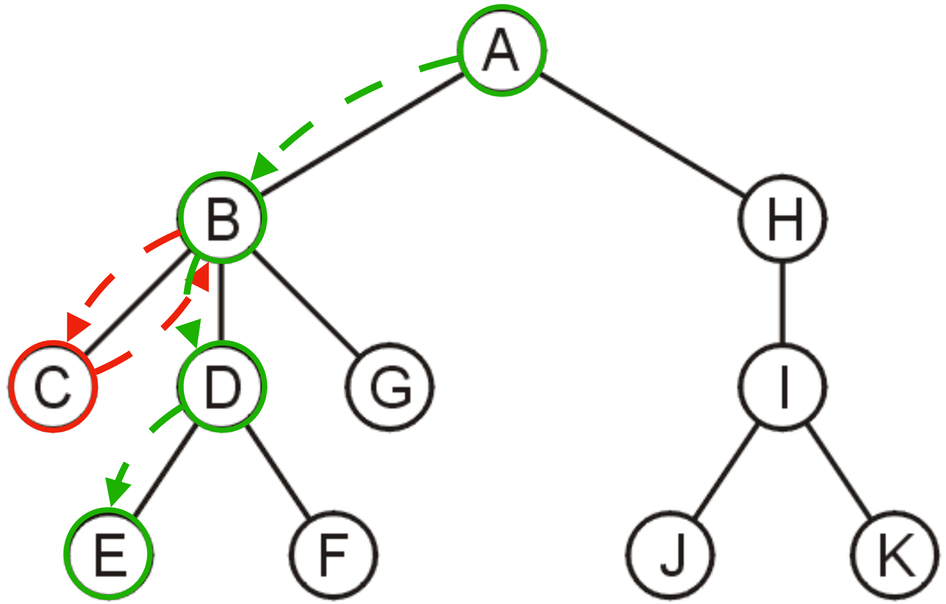
- Push D, the next not yet visited child of B



Demo

DFS Tree

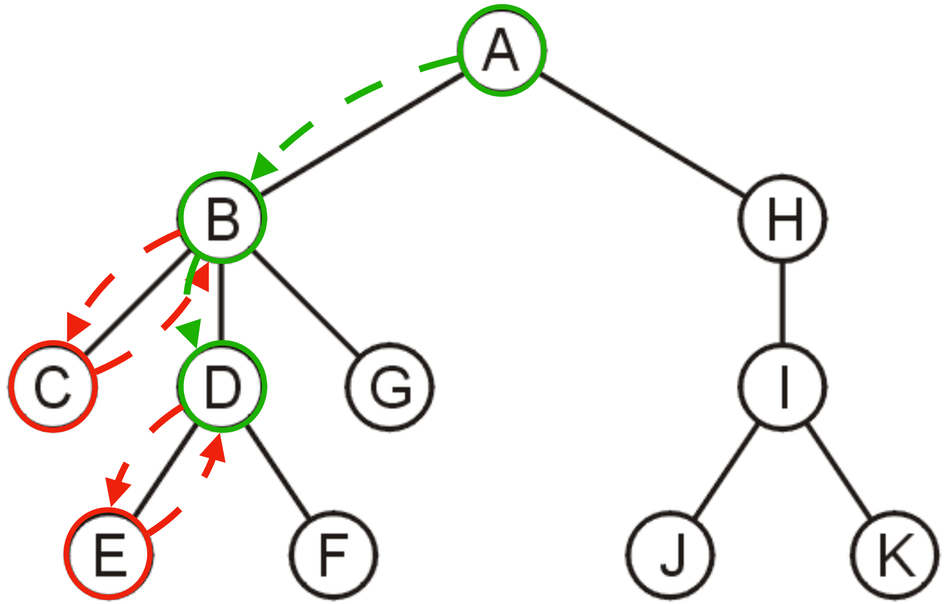
- Push E



Demo

DFS Tree

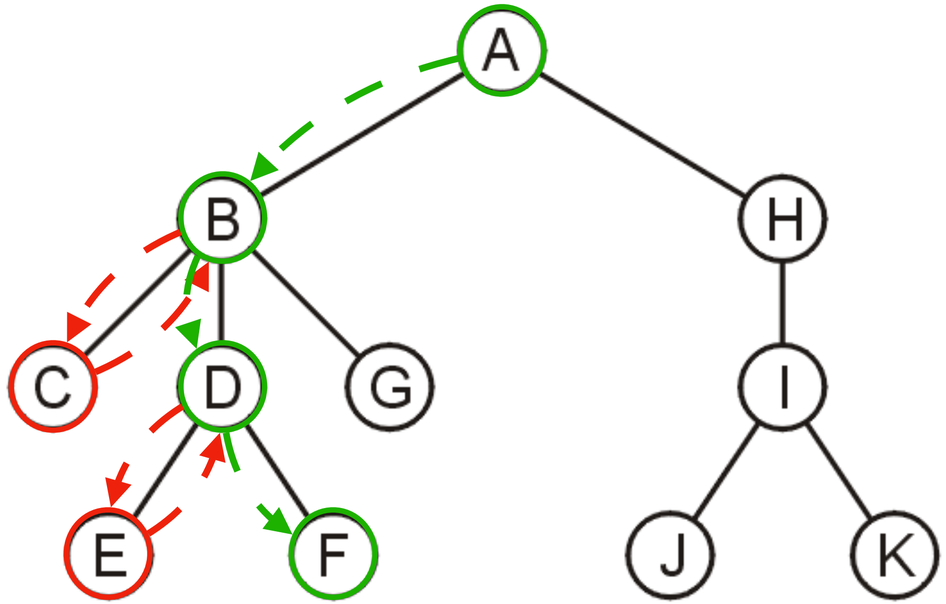
- Pop E, go back to D



Demo

DFS Tree

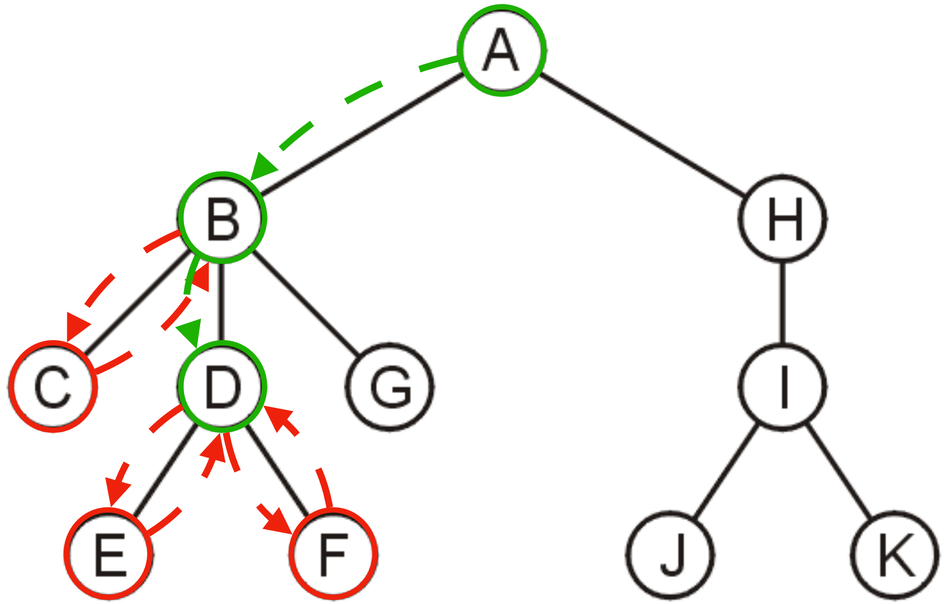
- Push F



Demo

DFS Tree

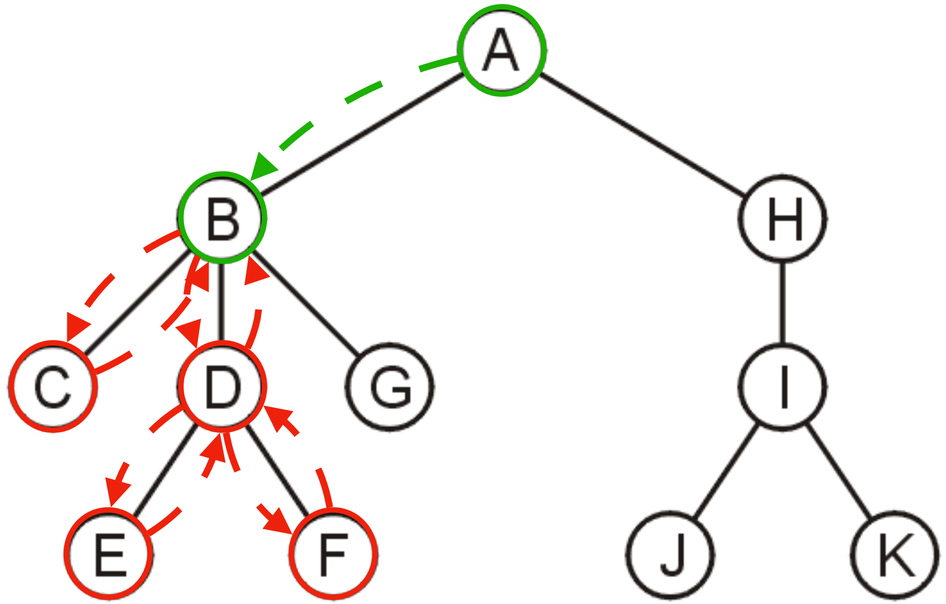
- Pop F, go back to D



Demo

DFS Tree

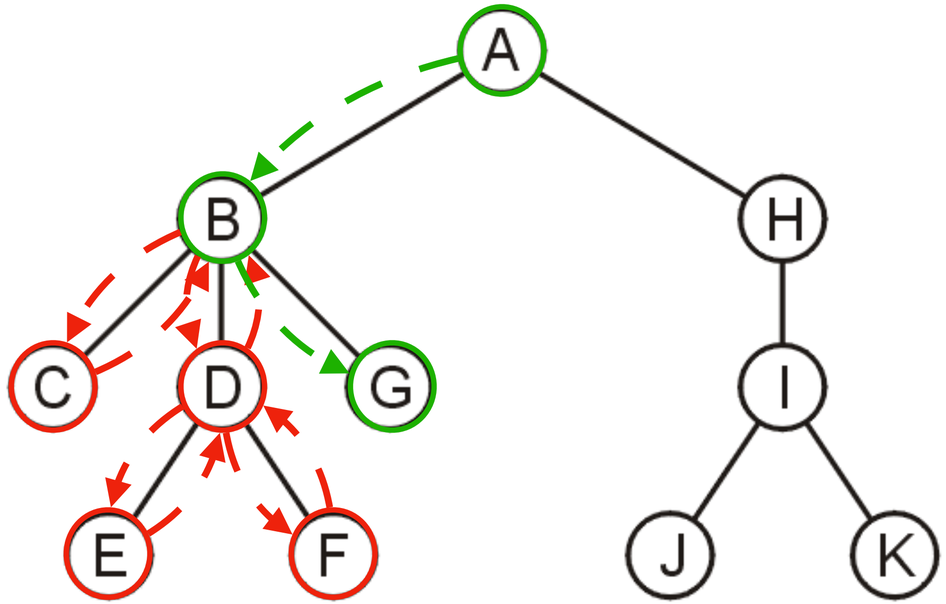
- Pop D, go back to B



Demo

DFS Tree

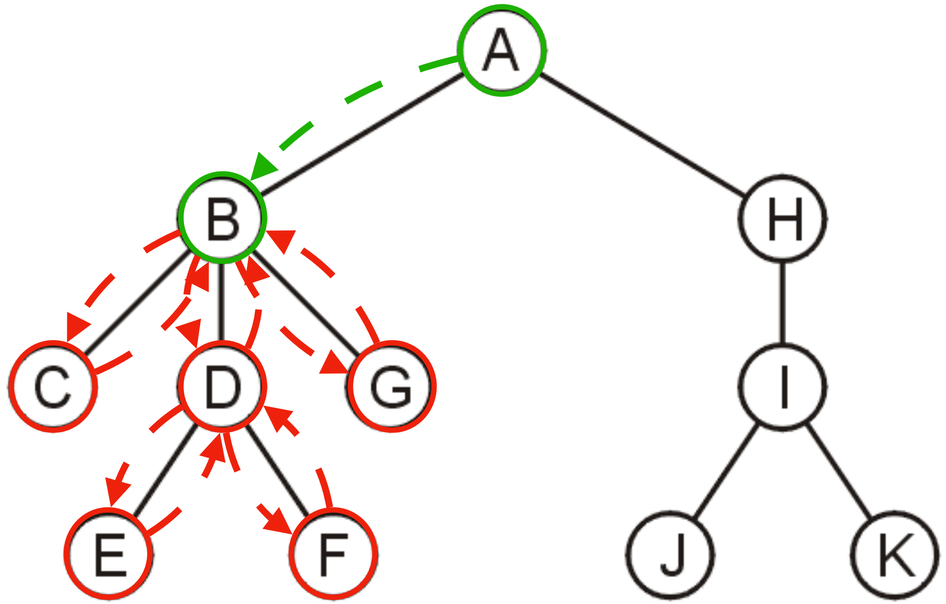
- Push G



Demo

DFS Tree

- Pop G, go back to B

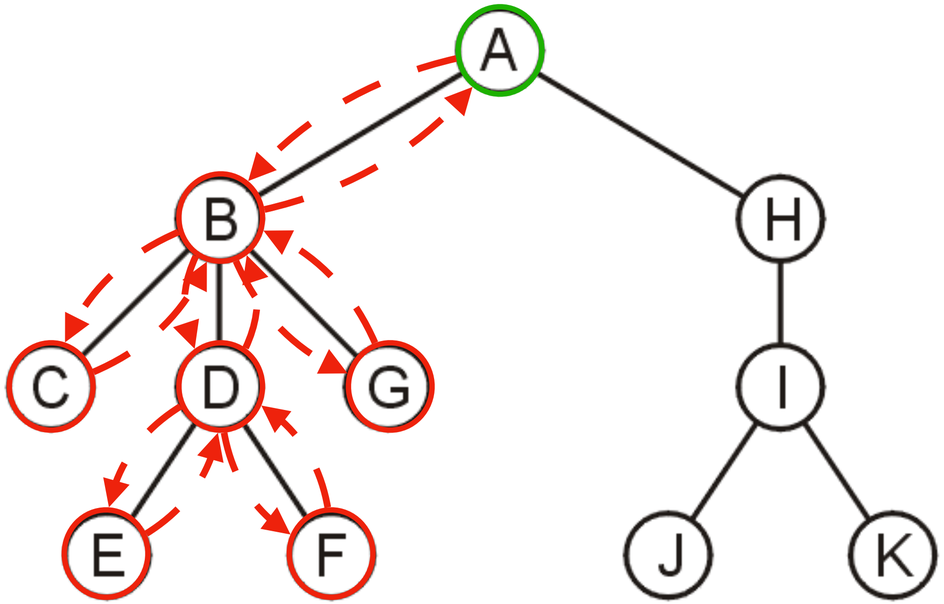


Demo

DFS Tree

- Pop B, go back to A

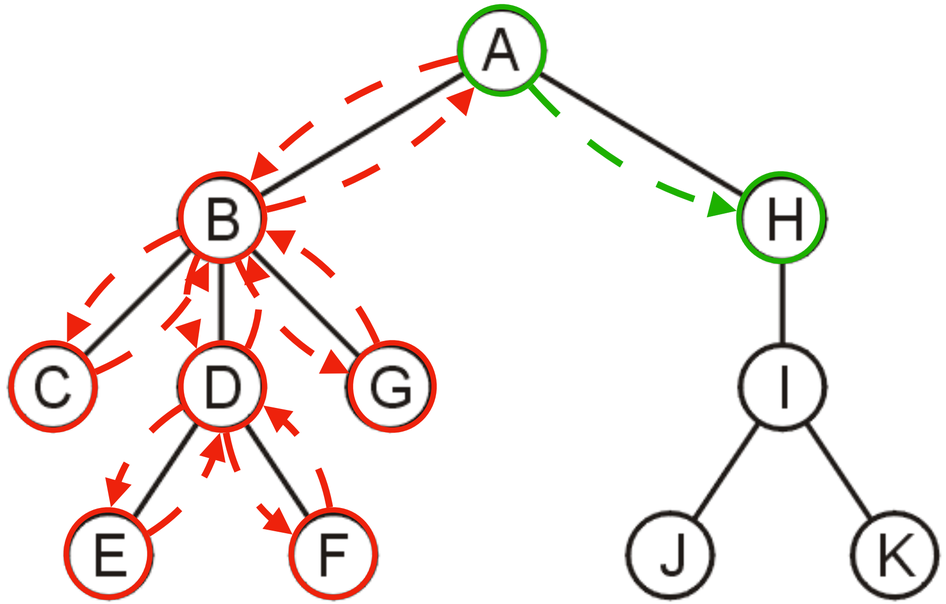
A



Demo

DFS Tree

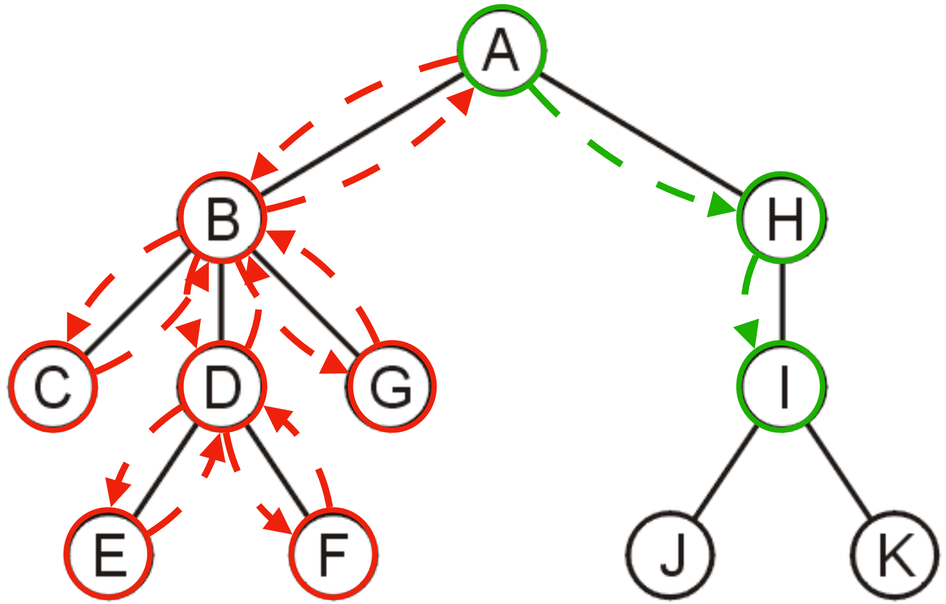
- Push H



Demo

DFS Tree

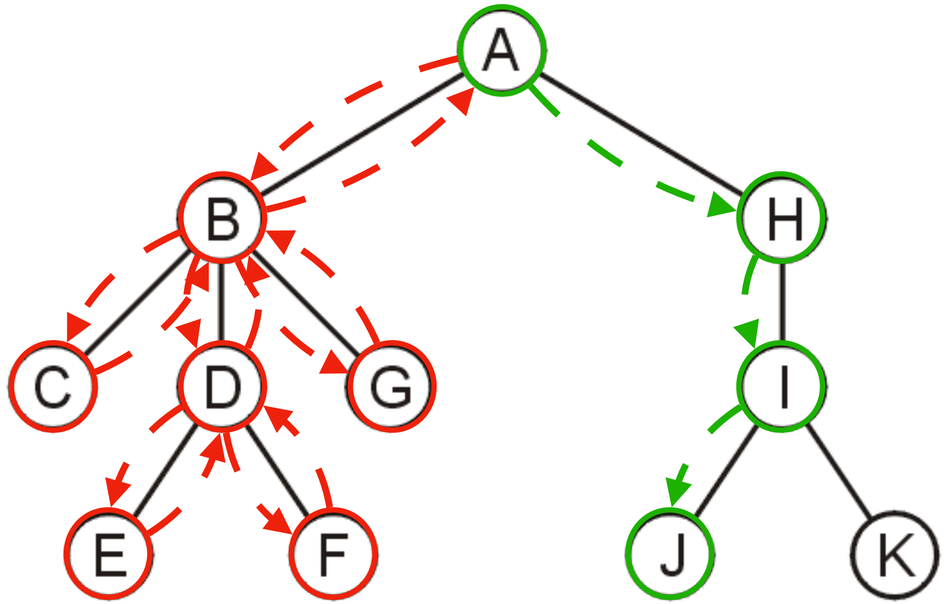
- Push I



Demo

DFS Tree

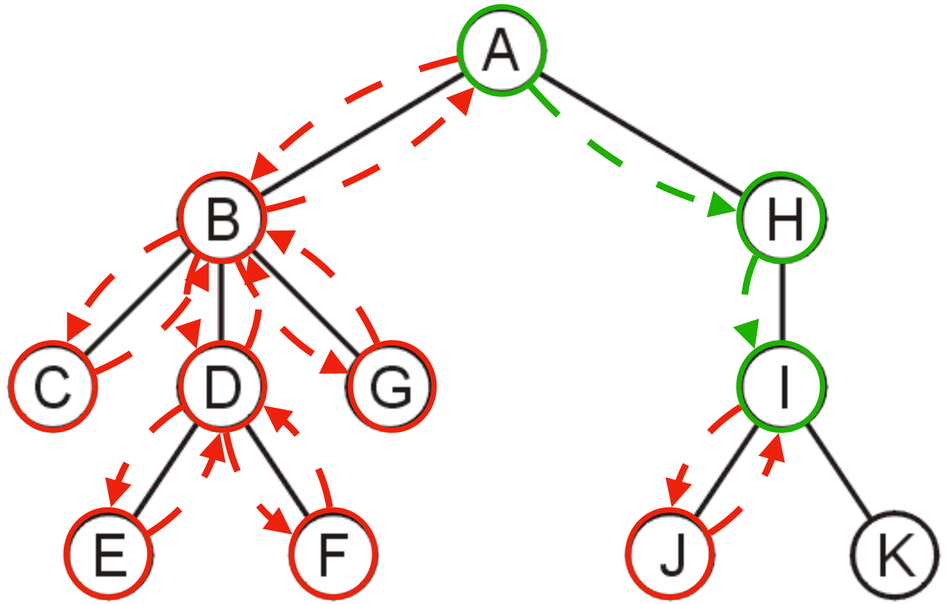
- Push J



Demo

DFS Tree

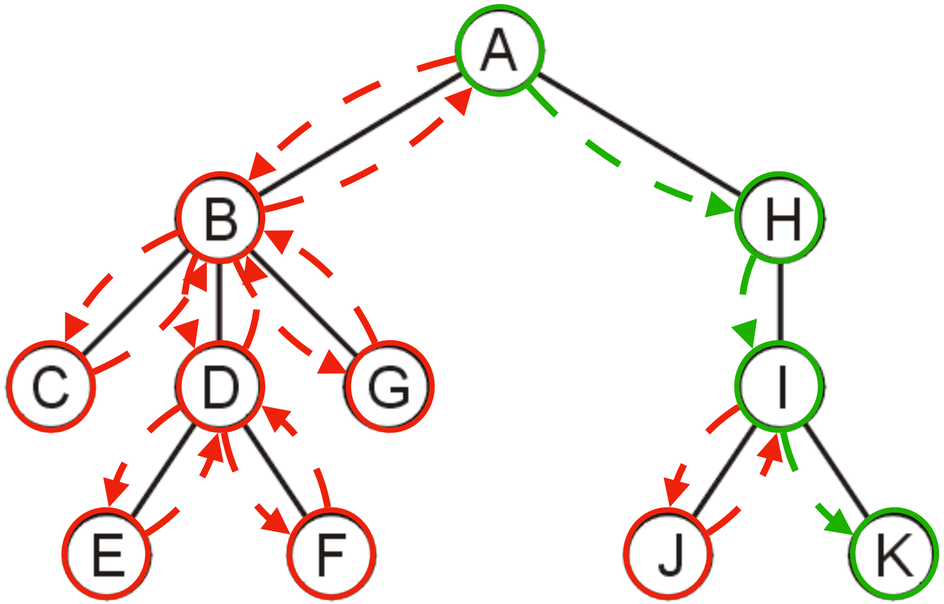
- Pop J, go back to I



Demo

DFS Tree

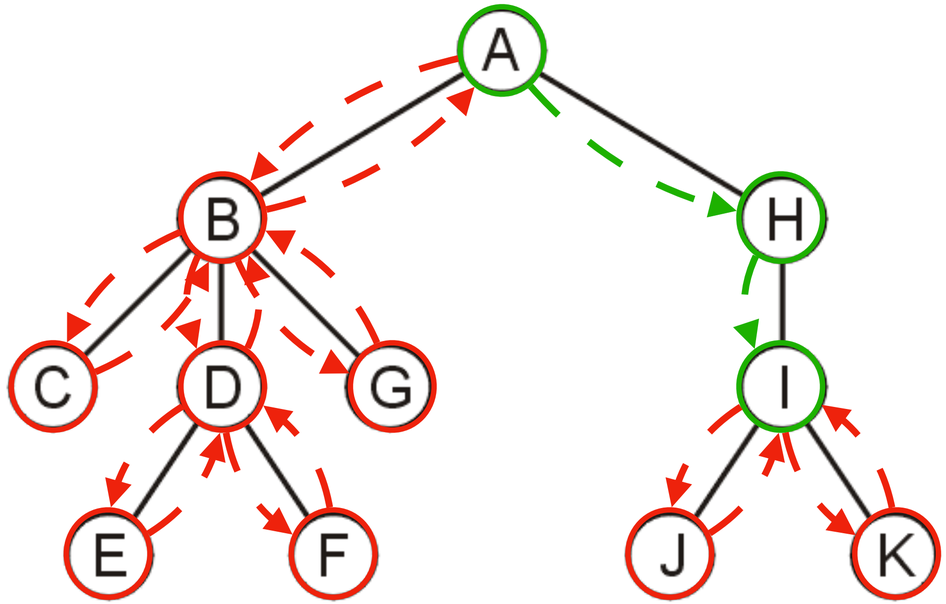
- Push K



Demo

DFS Tree

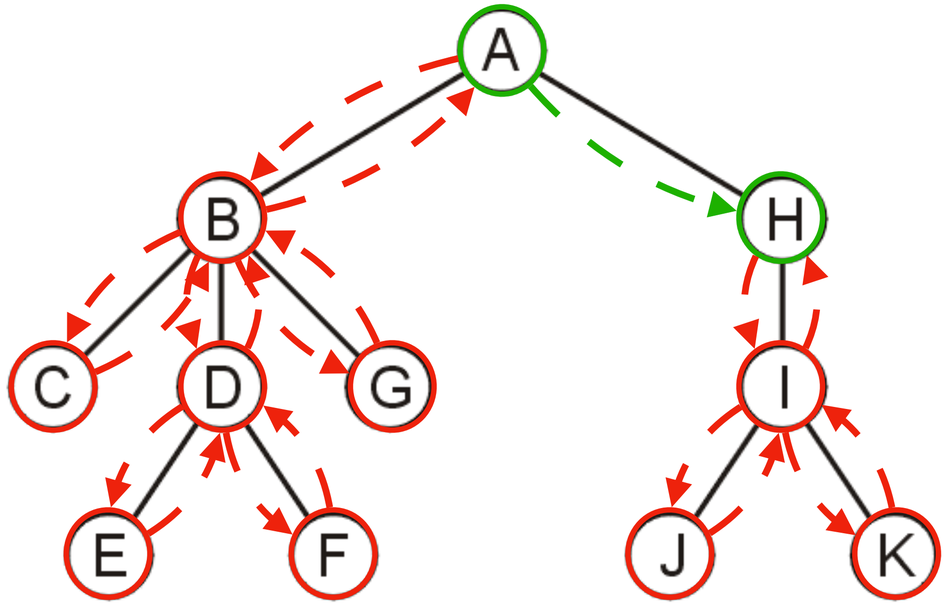
- Pop K, go back to I



Demo

DFS Tree

- Pop I, go back to H



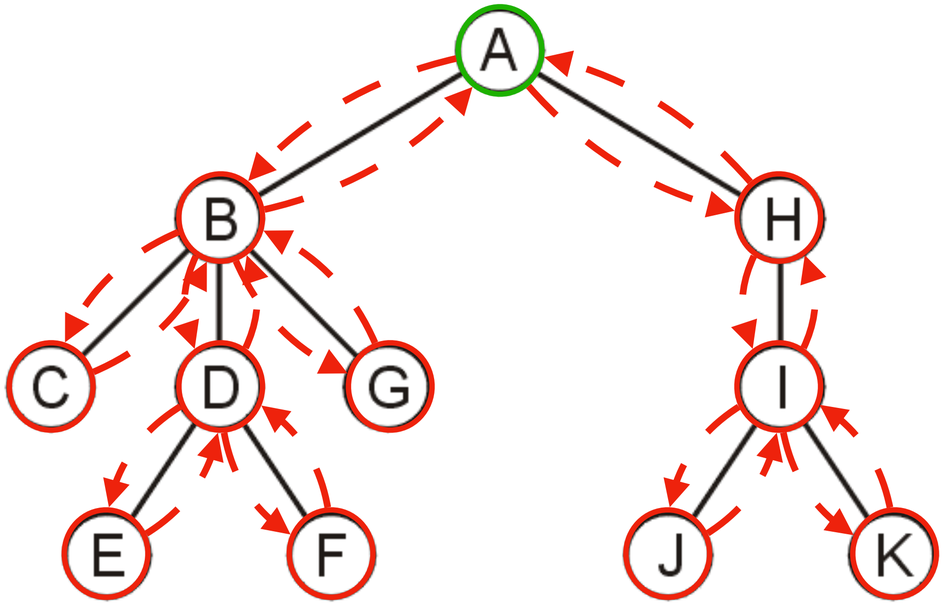
Demo

P1
Searching

DFS Tree

- Pop H, go back to A

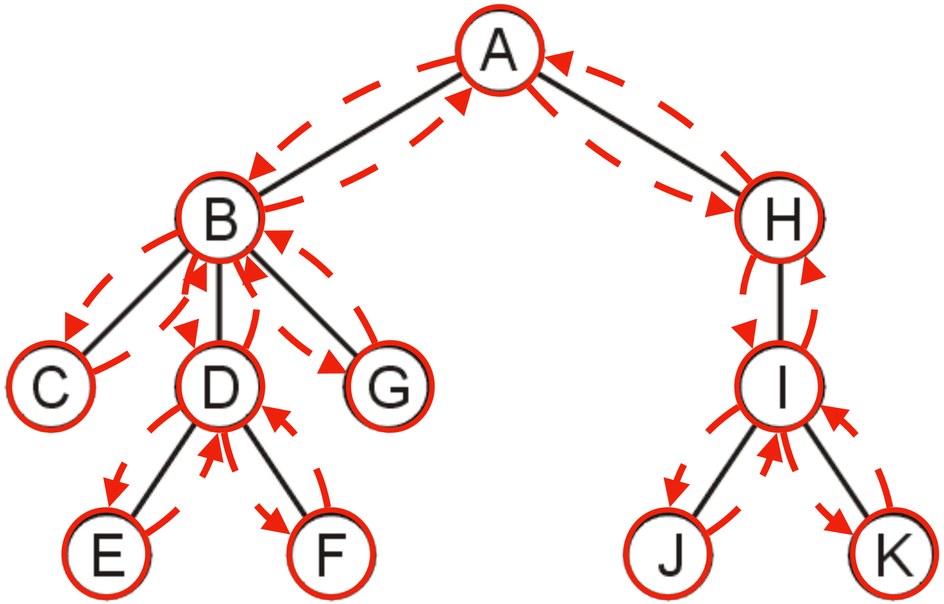
A



Demo

DFS Tree

- Pop A, traversal completed
- Traversal Sequence
  A-B-C-D-E-F-G-H-I-J-K



Demo

DFS Tree Analysis

Demo

DFS Tree Analysis

- Assuming N nodes, M connections, what is the time and space complexity?

Demo

DFS Tree Analysis

- Assuming N nodes, M connections, what is the time and space complexity?
  - Time complexity: $O(N + M)$

Demo

DFS Tree Analysis

- Assuming N nodes, M connections, what is the time and space complexity?
  - Time complexity: $O(N + M)$
    - Is it $\Theta(N + M)$ as well? Why?

Demo

DFS Tree Analysis

- Assuming N nodes, M connections, what is the time and space complexity?
  - Time complexity: $O(N + M)$
    - Is it $\Theta(N + M)$ as well? Why?
  - Space complexity: $\Theta(N + M)$

Demo

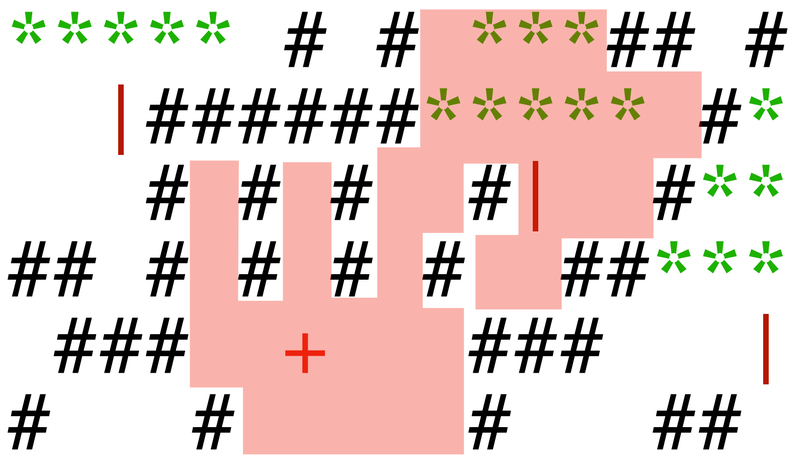
DFS Tree Analysis

- Assuming N nodes, M connections, what is the time and space complexity?
  - Time complexity: $O(N+M)$
    - Is it $\Theta(N+M)$ as well? Why?
  - Space complexity: $\Theta(N+M)$
    - Is it $\Theta(N+M)$ as well? Why?

Demo

DFS Example

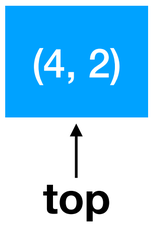
- Village of the Sorcerer
  - Certain areas of the map are not reachable (fences, houses)
  - Certain areas might be entirely blocked off by fences
  - +: player
  - Light Red: player accessible regions



Demo

DFS Example

- We start by initialising a Stack to store coordinates
- push(playerCoordinate), which is 4,2
  - `push((4,2));`
  - In C++, you can simulate this using 2 `int`-based stacks

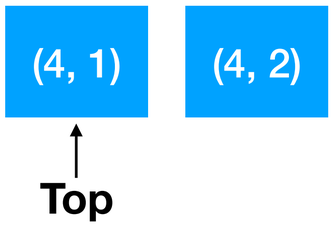


	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#				#
4	#		+	#	#
5		#		#	

Demo

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (4,2);`
  - `push(4,1)`



	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#				#
4	#			#	#
5		#		#	

Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (4,1);`
  - `push(3,1);`

	0	1	2	3	4
0	*	#			
1	#	#	#	#	#
2	#		#		#
3	#				#
4	#			#	#
5		#		#	

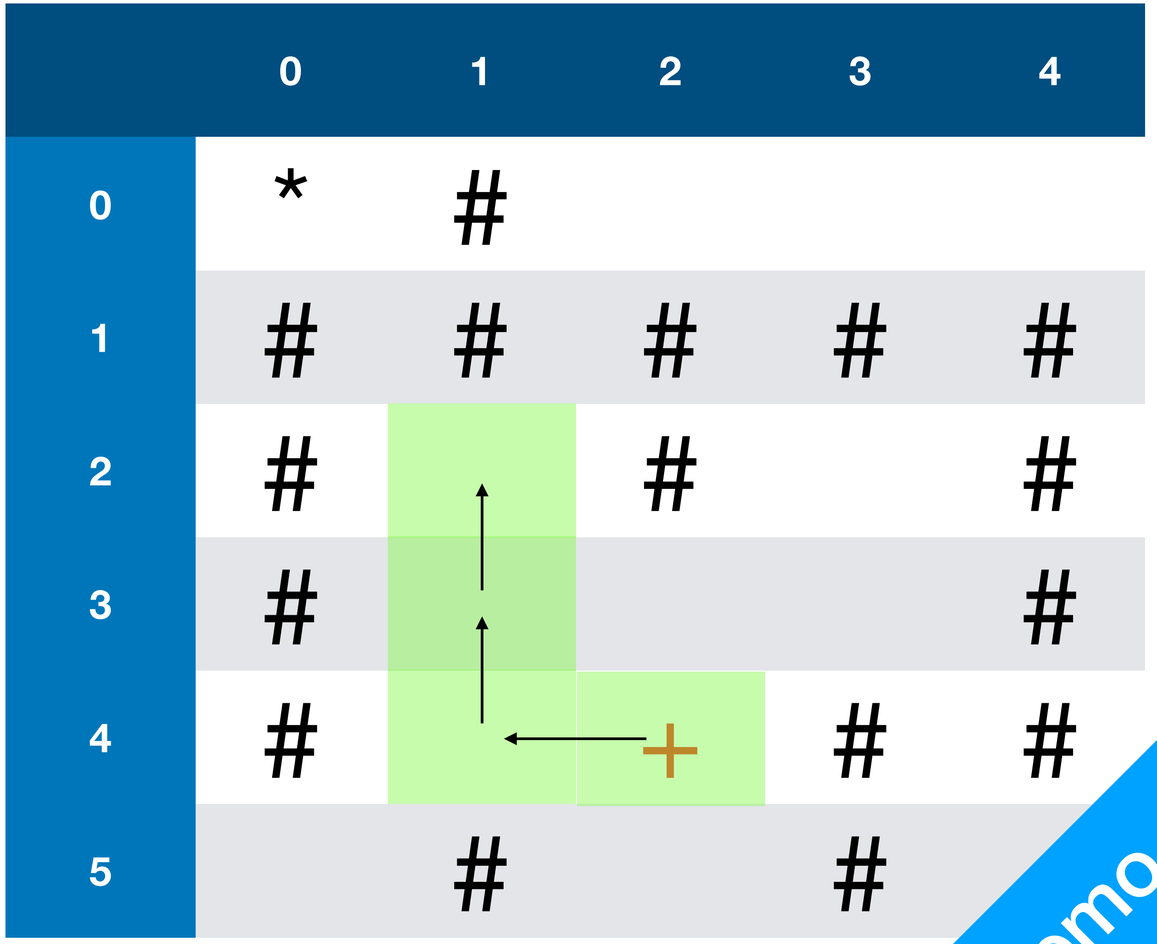
Demo

1. P030

P1
Searching

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,1);`
  - `push(2, 1)`

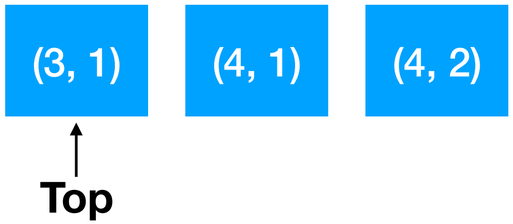


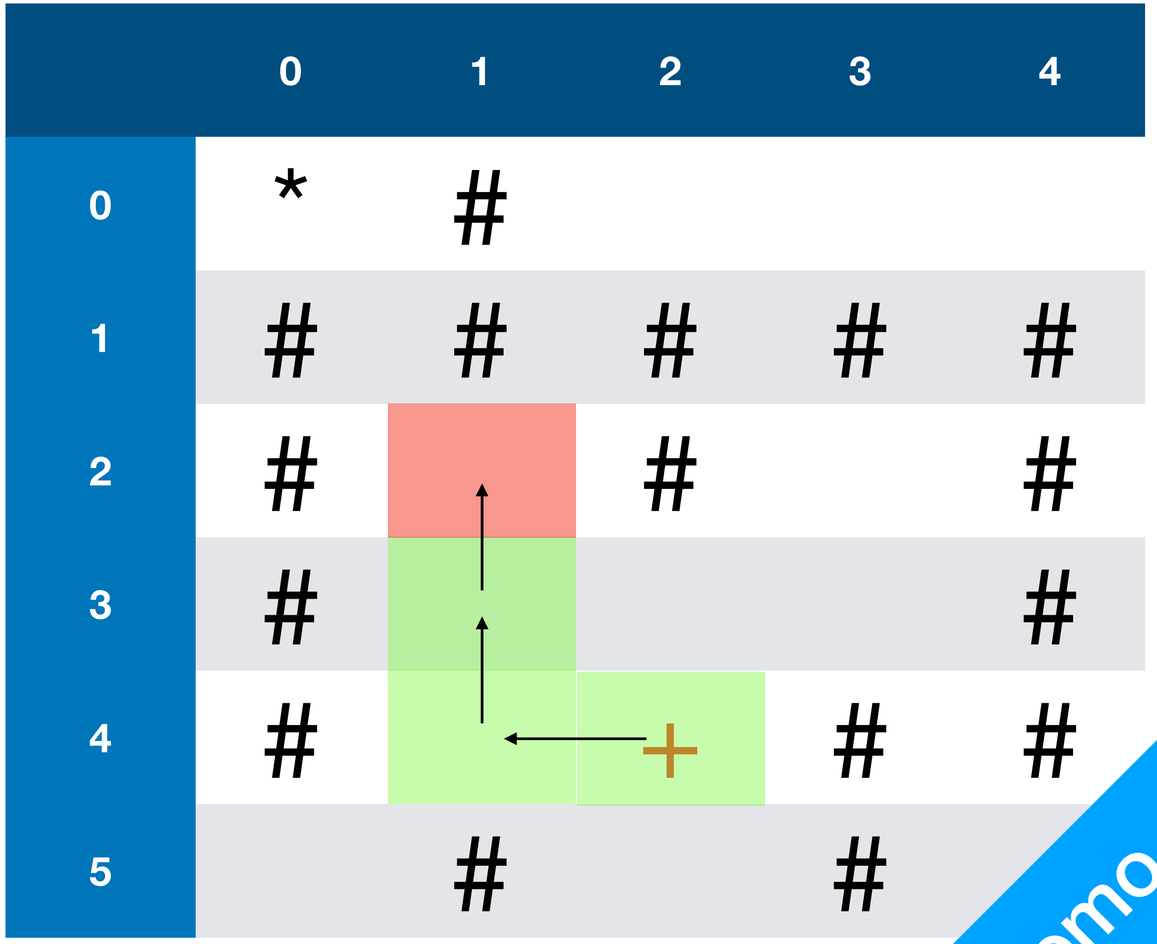
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (2,1);`
  - `pop()`



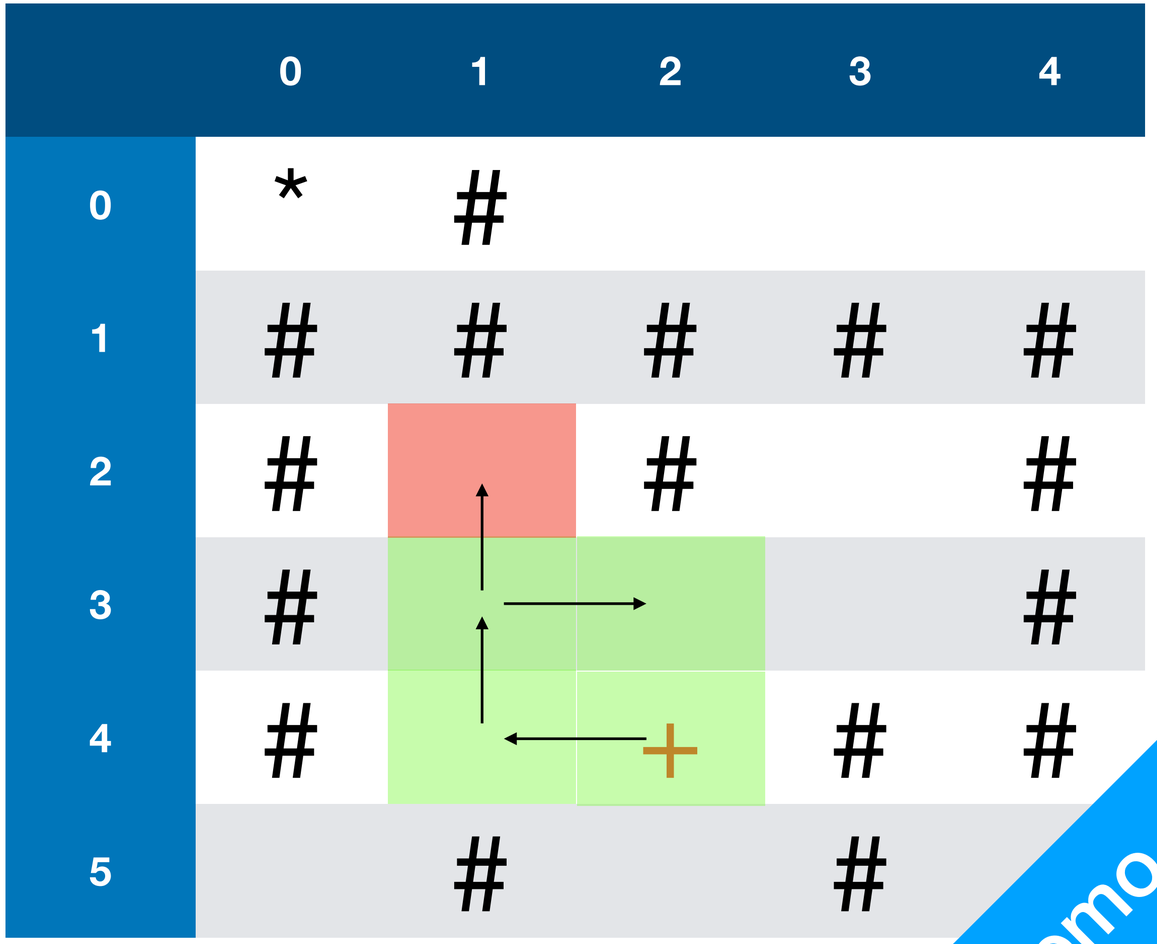


Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,1);`
  - `push(3,2)`

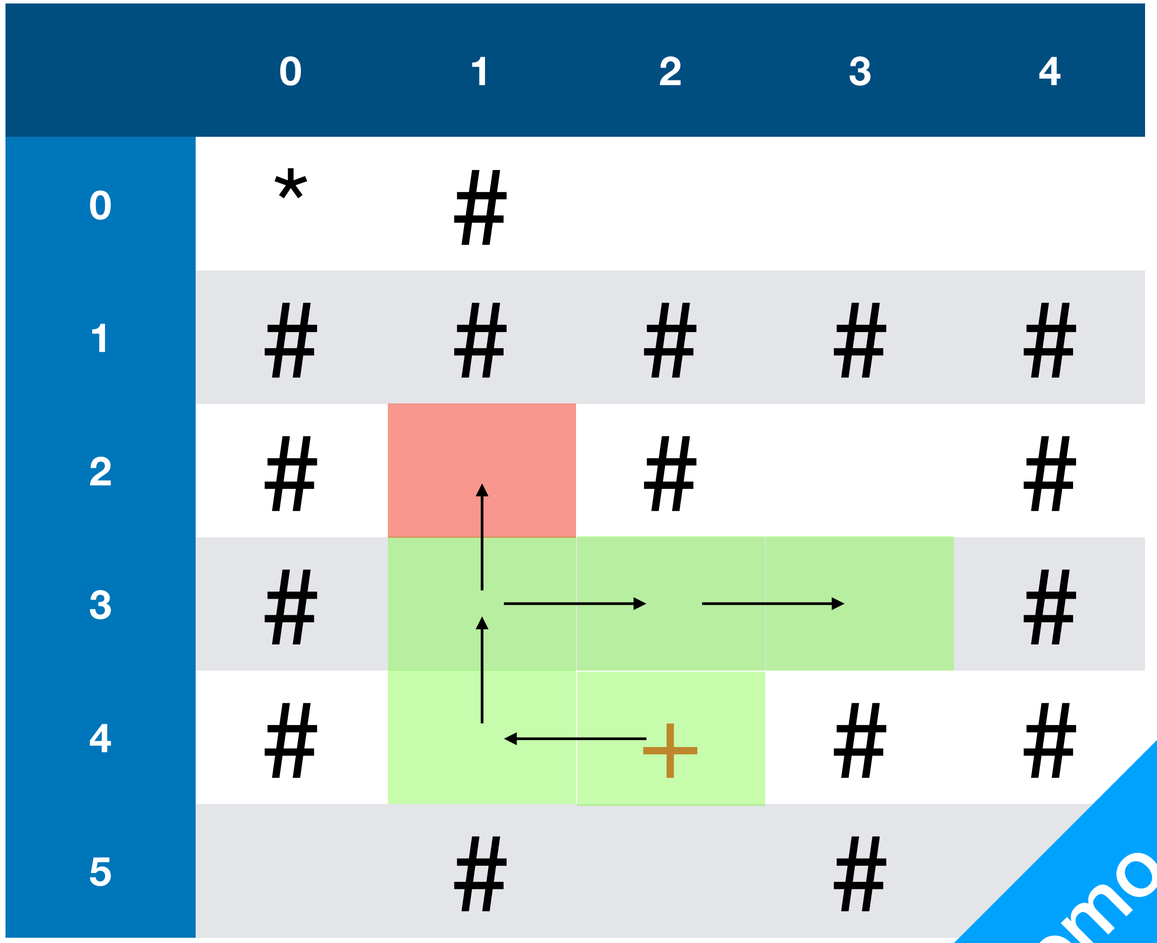


Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,2);`
  - `push(3,3)`

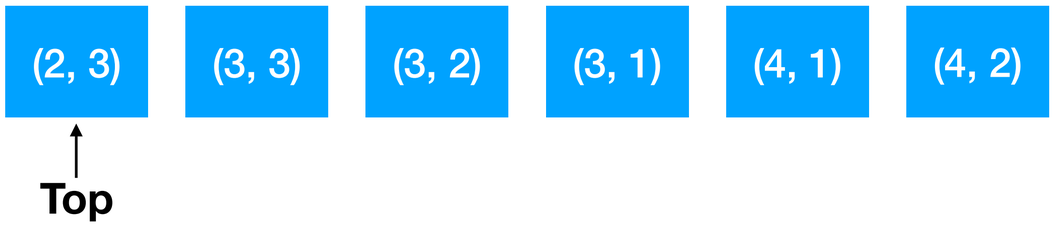


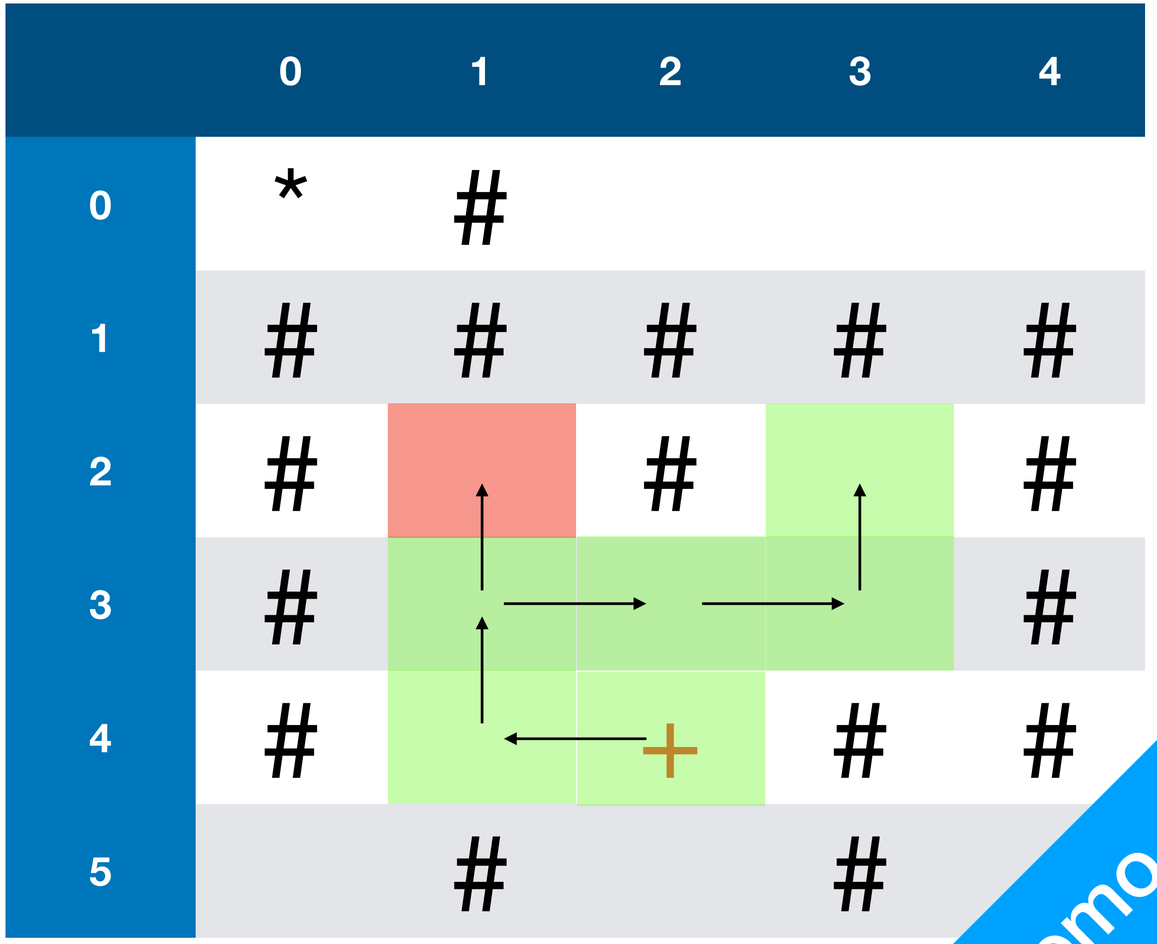
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,3);`
  - `push(2,3)`



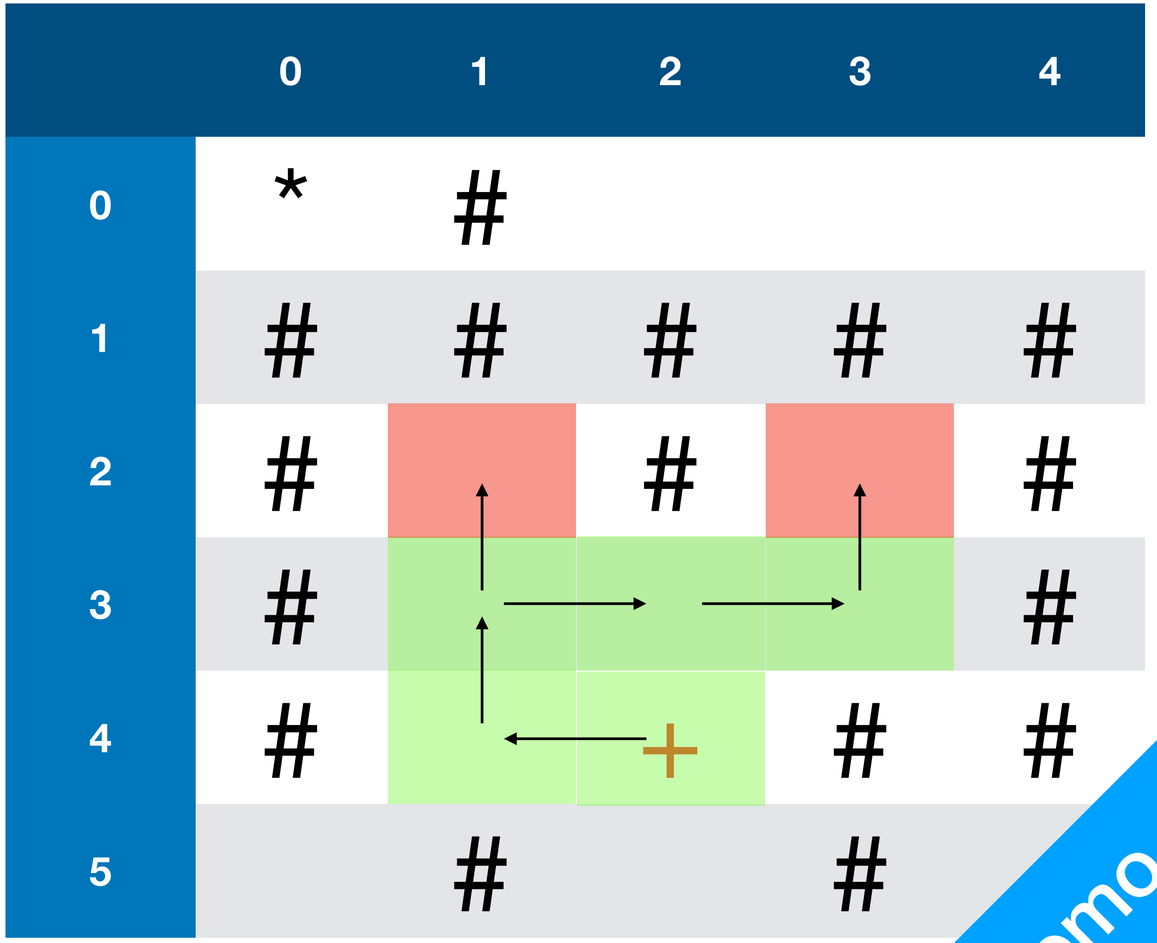


Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (2,3);`
  - `pop()`

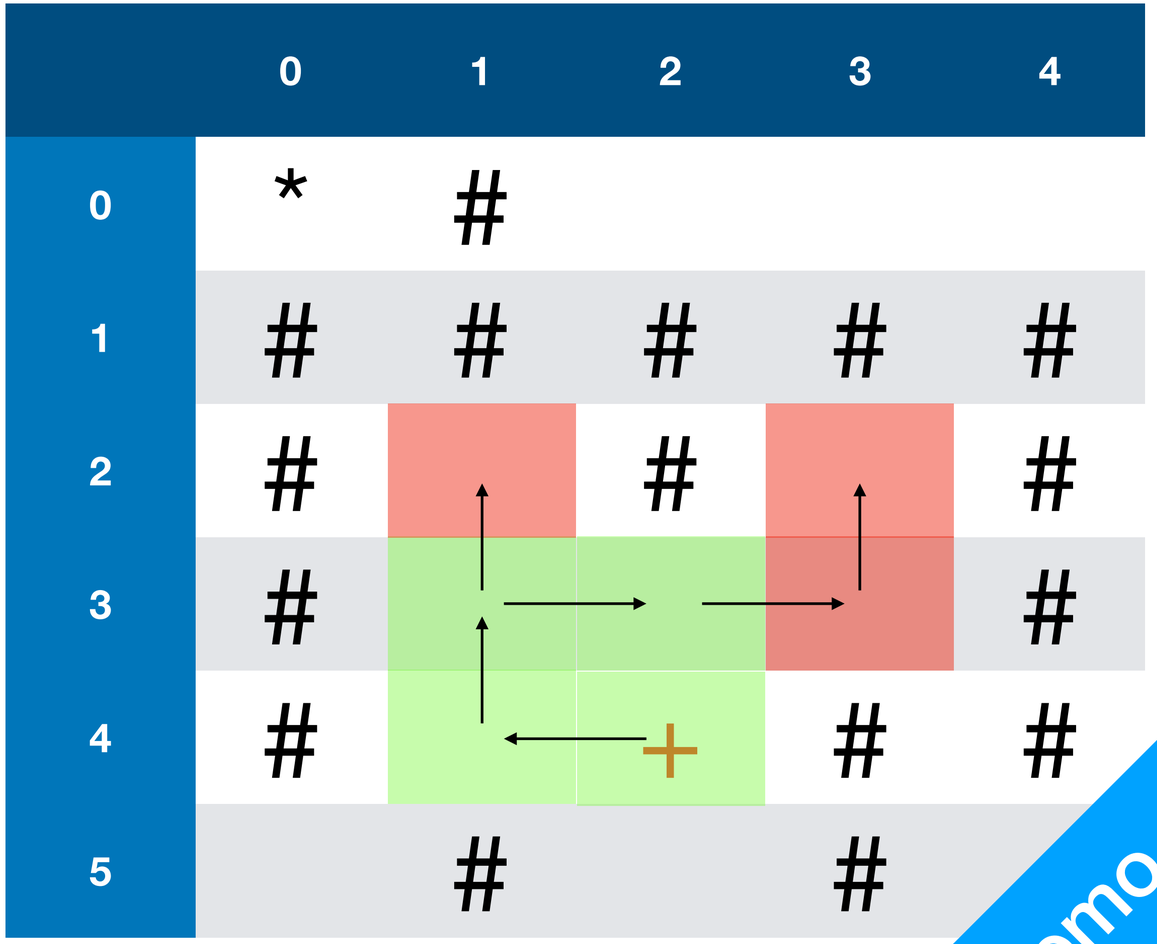


Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,3);`
  - `pop()`

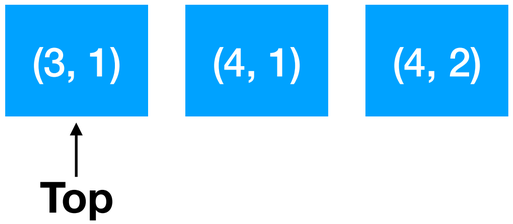


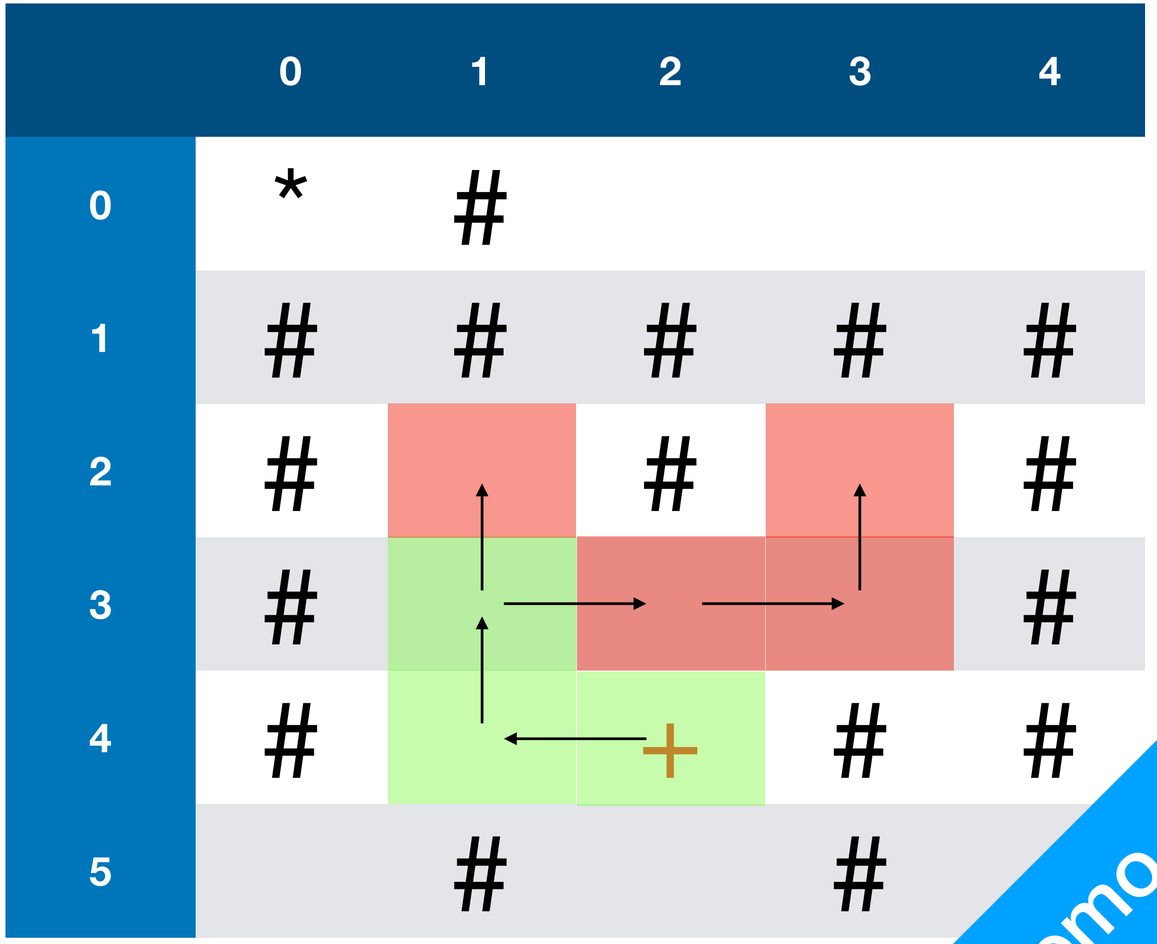
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,2);`
  - `pop()`



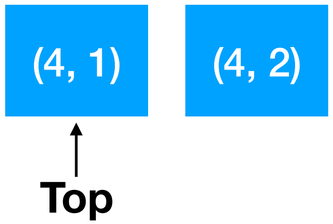


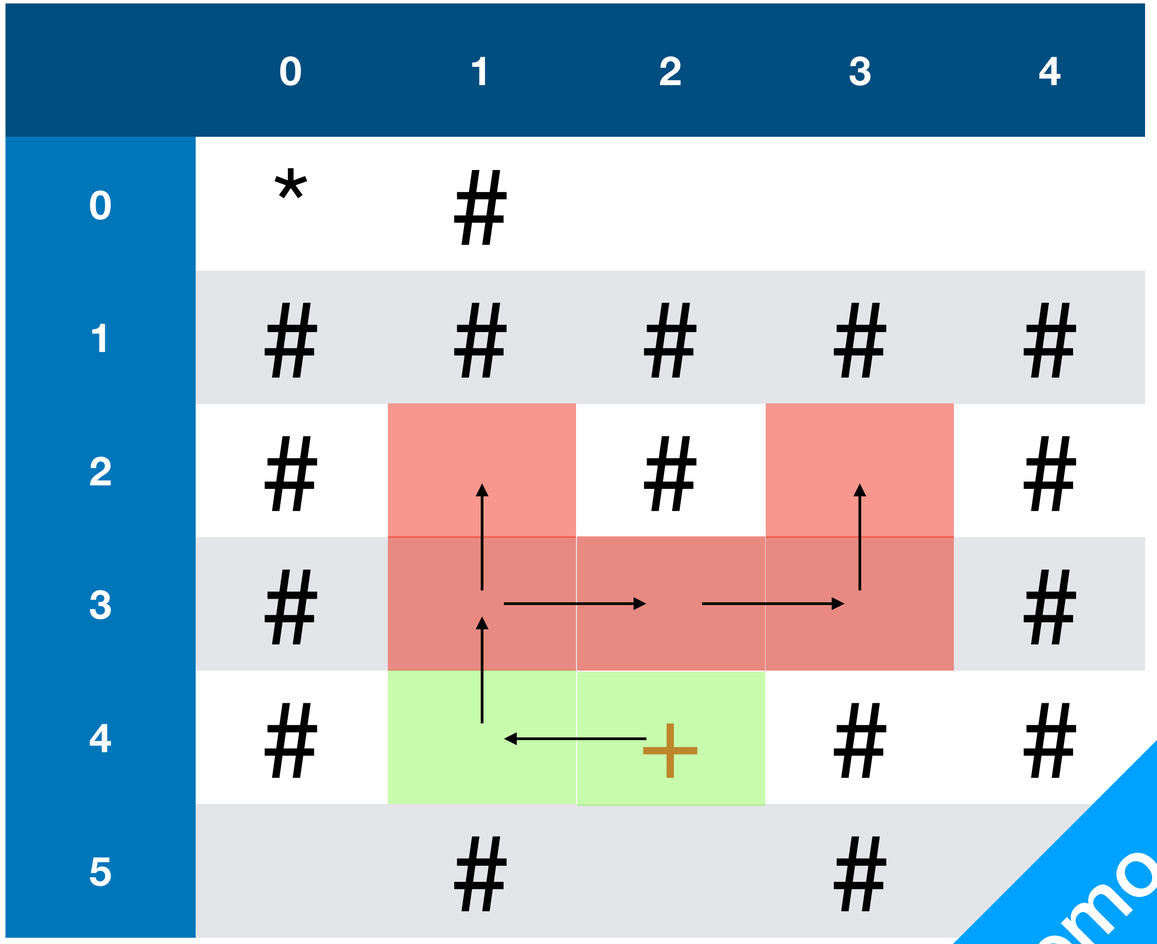
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (3,1);`
  - `pop()`



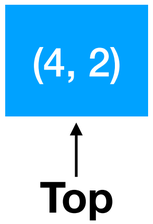


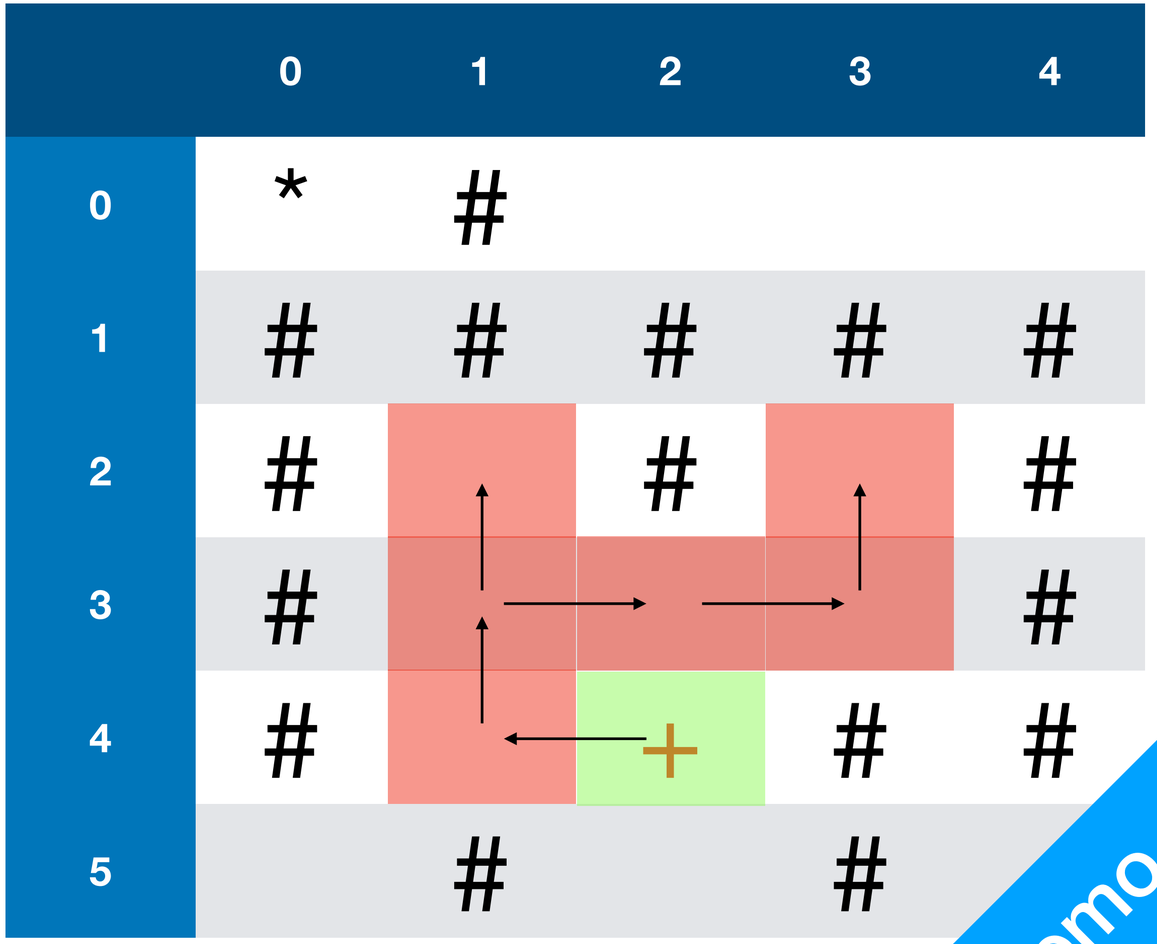
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (4,1);`
  - `pop()`



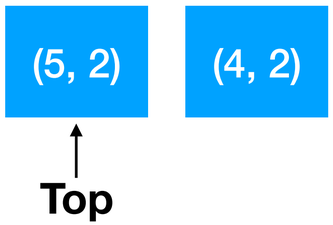


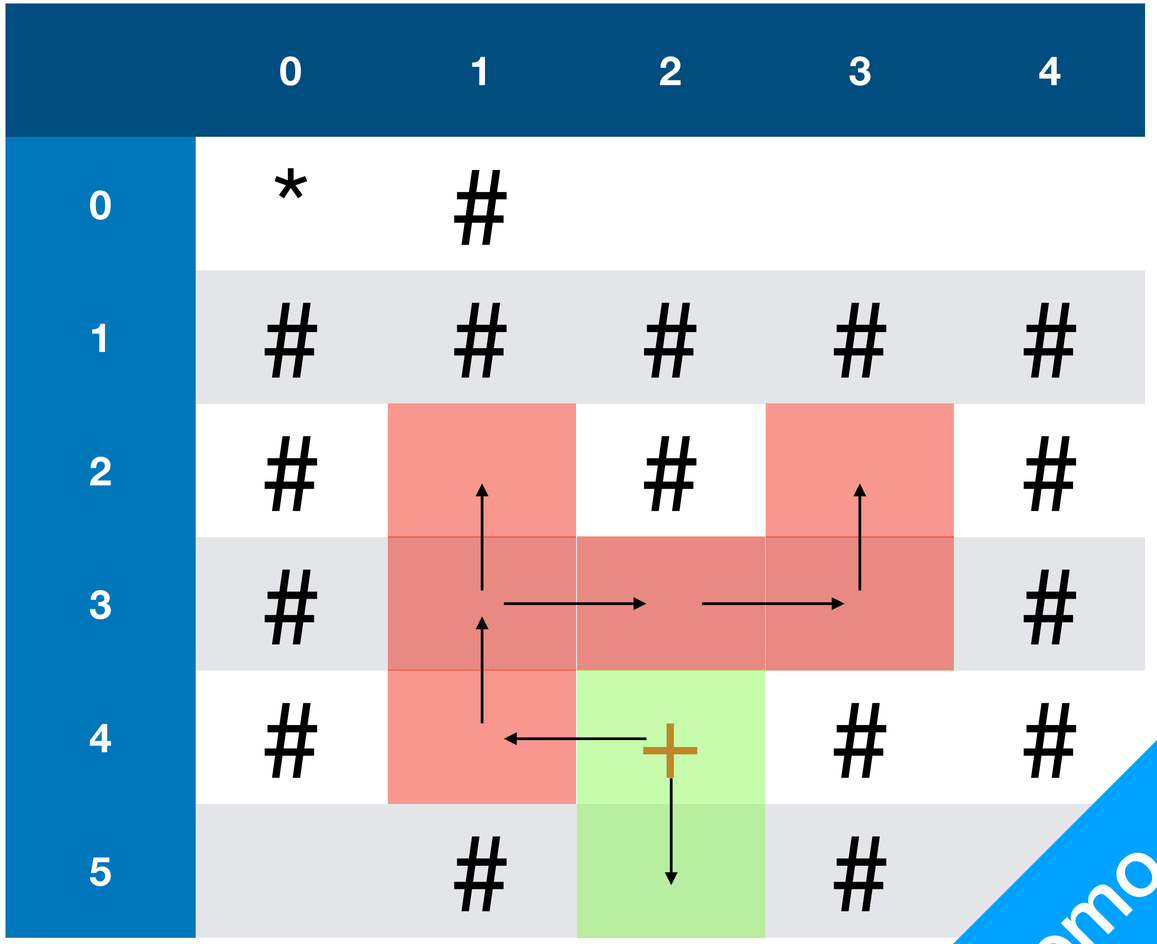
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (4,2);`
  - `push(5,2)`



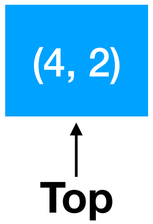


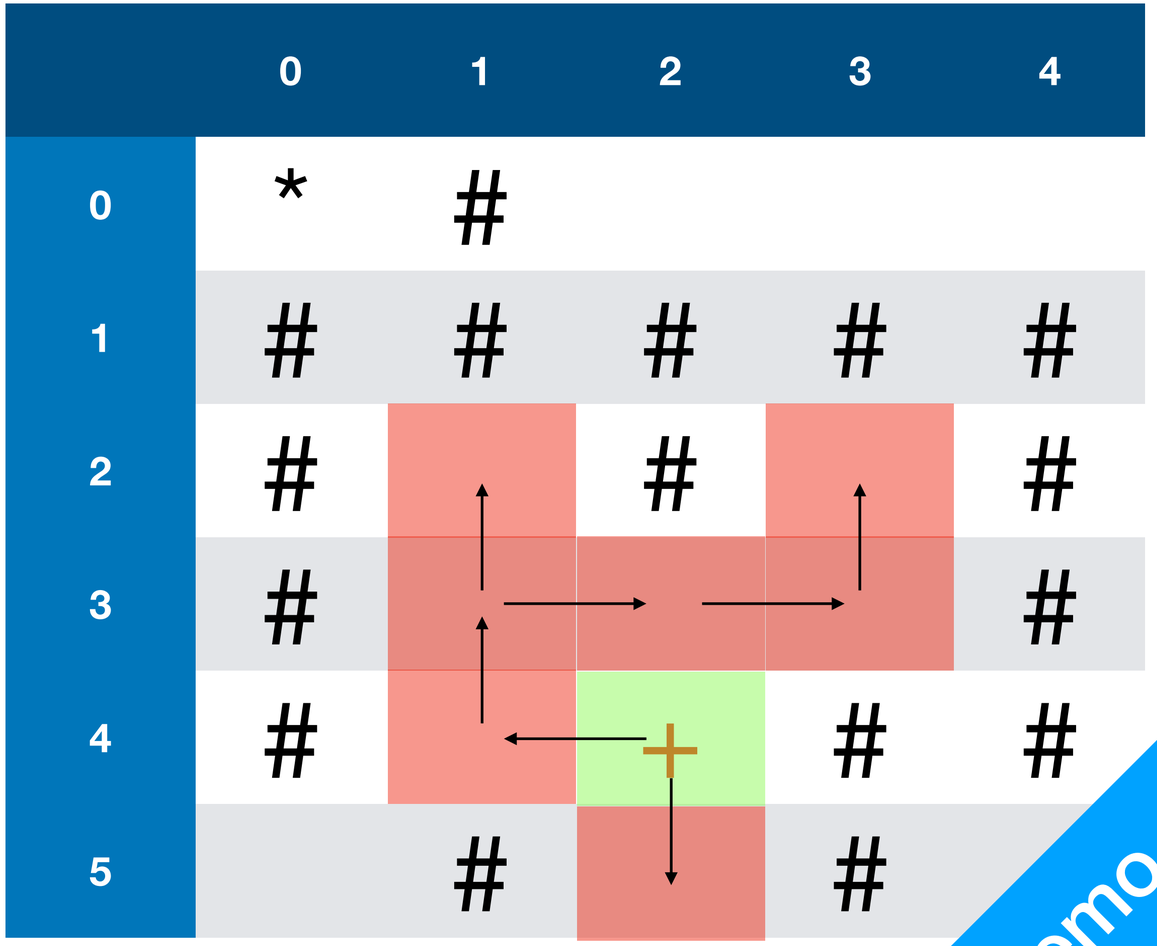
Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (5,2);`
  - `pop()`



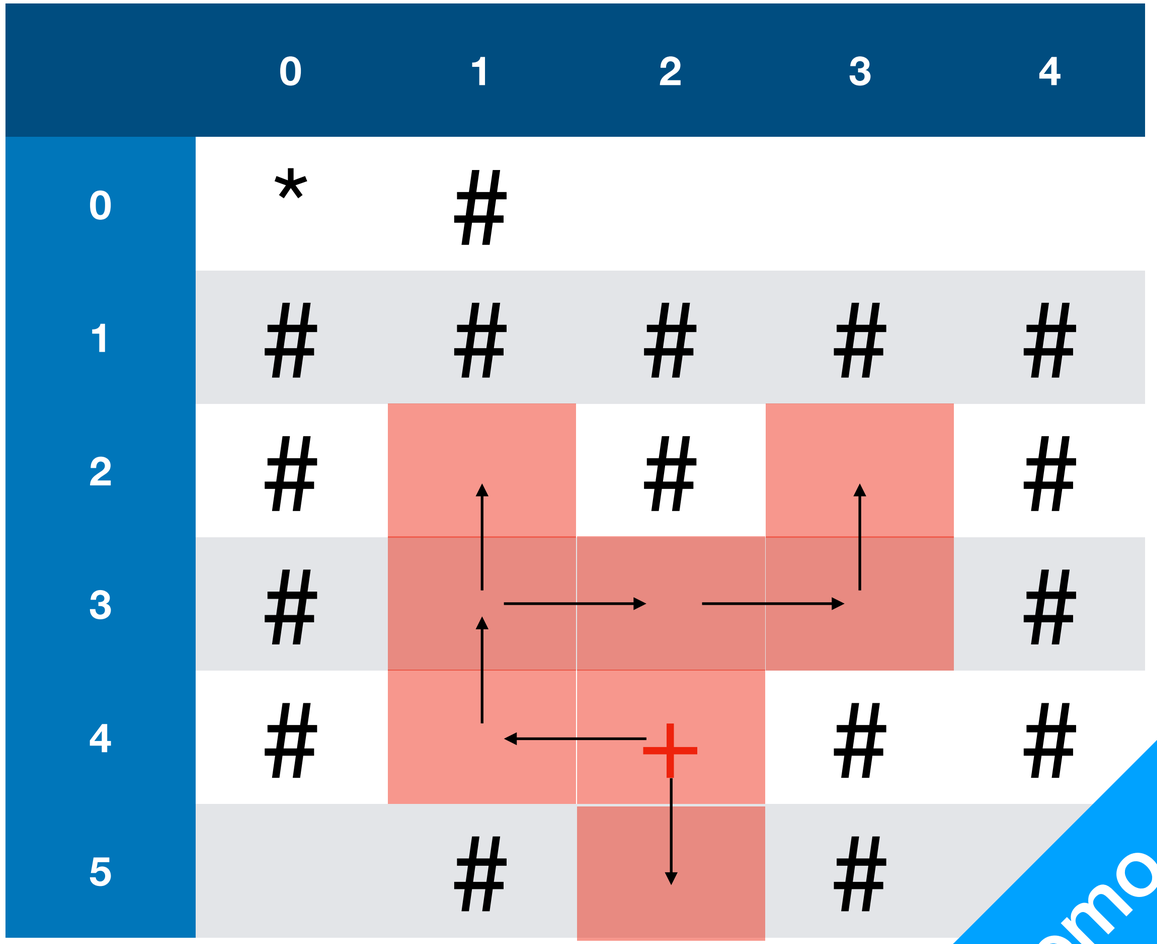


Demo

1. P030

DFS Example

- For every frontal element, traverse randomly until cannot further
  - `top() = (4,2);`
  - `pop()`



Top

DFS Example

Demo

DFS Example

- Assuming map size $N \times M$, what is the time and space complexity?

Demo

DFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$

Demo

DFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?

Demo

DFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?
  - Space complexity: $O(N \times M)$

Demo

DFS Example

- Assuming map size $N \times M$, what is the time and space complexity?
  - Time complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?
  - Space complexity: $O(N \times M)$
    - Is it $\Theta(N \times M)$ as well? Why?

Demo

Summary

- Lecture 7: Data Structure 1-4
  - Linked List
  - Stack
  - Queue
  - Θ and Big-O Notation
  - BFS, DFS

Future