Continuing Education: Searching Algorithms

Peter Jang

Agenda

- Introduction
- Time Complexity - O(n)
- List Search
  - Linear/Sequential Search
  - Binary Search
- Graph Search
  - Breadth First Search
  - Depth First Search
- Python Exercises
- Code Review

Github Repository For Exercises

https://github.com/byujan/searching_algorithms

Introduction

Searching Algorithms are a series of instructions that retrieves an end goal stored within some data structure.

End Goal:

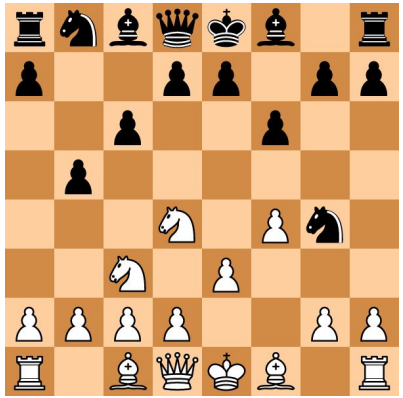
- Integer
- Object
- Solution

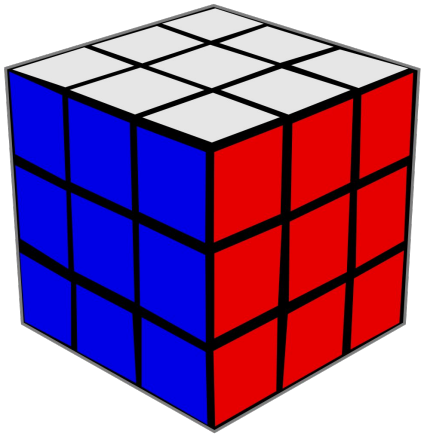
Data Structures:

- Lists
- Matrices
- Graphs

Applications of Searching Algorithms

- Navigation Apps
- Game Playing Agents
  - Go
  - Chess
  - Tic Tac Toe
- Solving Puzzles
  - 2048
  - River Crossing
  - Rubiks Cube
- General Problem Solving





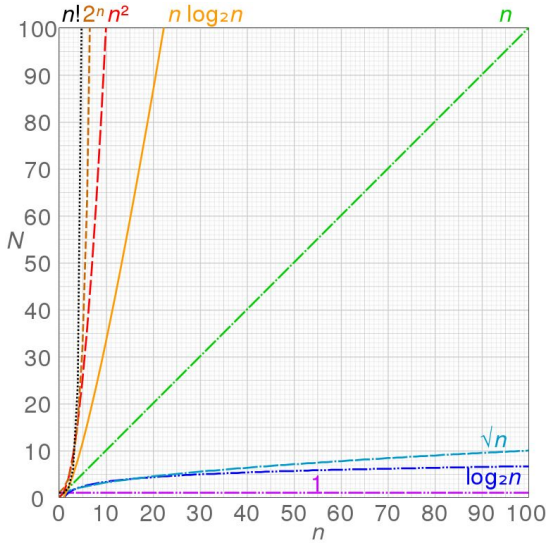
Time Complexity

Time Complexity is an analysis of a computing problem in which we define the time as a function of the problem size and try to estimate the growth of the execution time with respect to the dynamic problem size

Big O Notation

1. Efficiency
2. Time Factor
3. Space Complexity

Always refers to worst case scenario



Big O Notation : O(n)

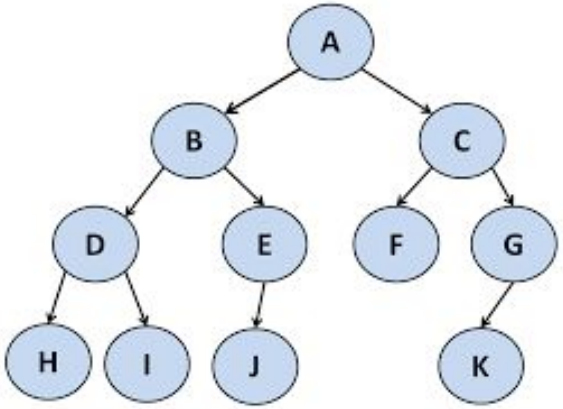
0 → N

12	9	3	15	21	43	25	1

0 → N

0 → N

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



652
+471
3

Linear / Sequential Search

A list search method in which every value is checked to see if it is the goal you are searching for

Goal : 17

Current: 6

List	6	15	5	3	9	3
Indices	0	1	2	3	4	5

Linear / Sequential Search

Goal : 17

Current: 15

List	6	15	5	3	9	3
Indices	0	1	2	3	4	5

Linear / Sequential Search

Goal : 17

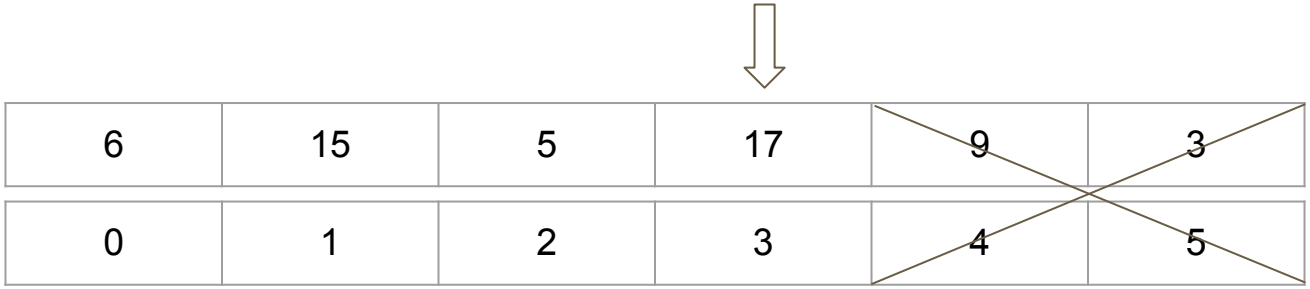
Current: 5

List	6	15	5	3	9	3
Indices	0	1	2	3	4	5

Linear / Sequential Search

Goal : 17

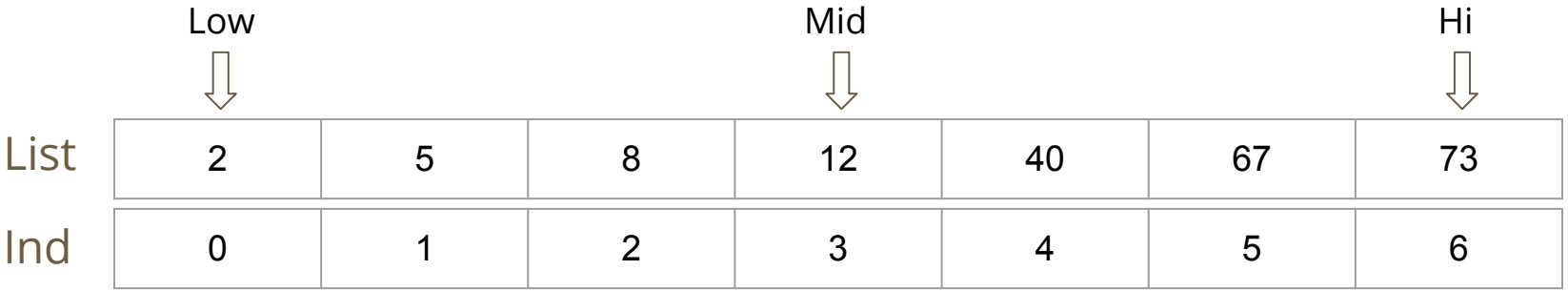
Current: 17



Returns the index at which Goal = Current : 3

Binary Search

A list search method in which the sorted list to be searched is halved at each iteration in order to find the goal

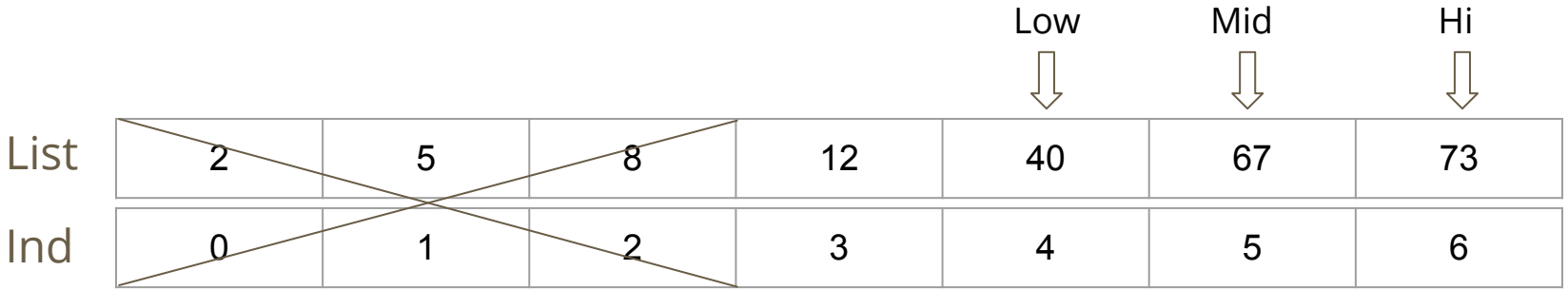


Goal: 40

Current: 12

Binary Search

A list search method in which the sorted list to be searched is halved at each iteration in order to find the goal

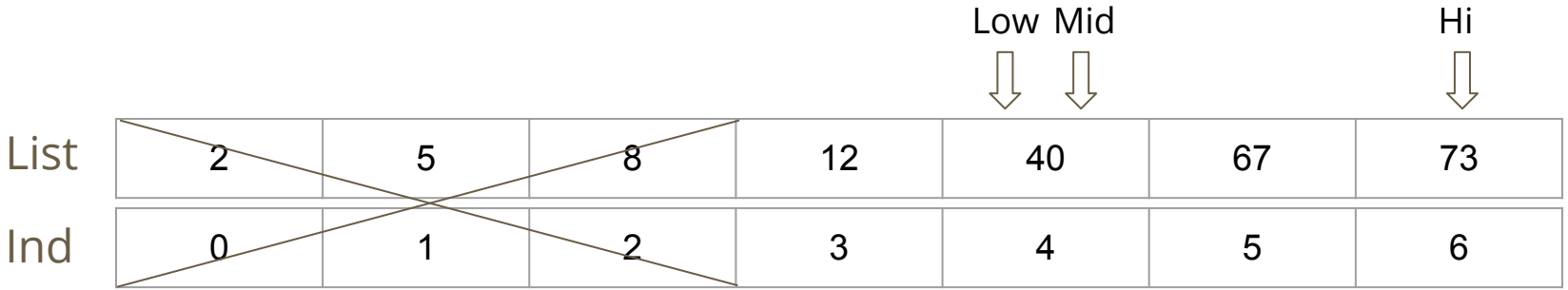


Goal: 40

Current: 67

Binary Search

A list search method in which the sorted list to be searched is halved at each iteration in order to find the goal



Goal: 40

Current: 40

Because Goal = Current, returns the index of Current: 4

Breadth First Search

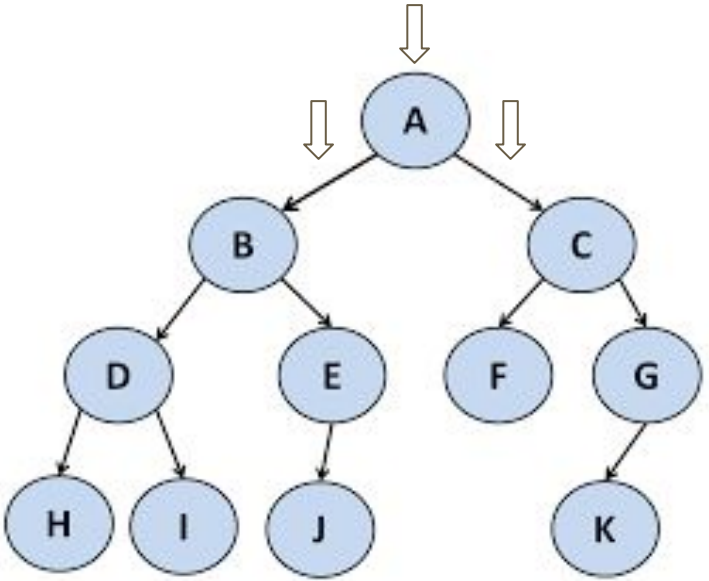
A tree/graph search algorithm that starts at an arbitrary point in the graph and explores all of the neighbor nodes

Goal: I

Current: A

Frontier: [B, C]

Visited: [A, B, C]



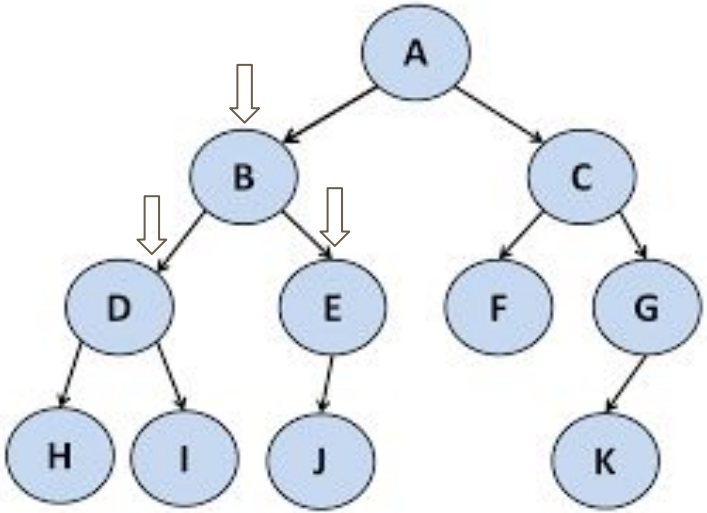
Breadth First Search

Goal: I

Current: C

Frontier: [C, D, E]

Visited: [A, B, C, D, E]



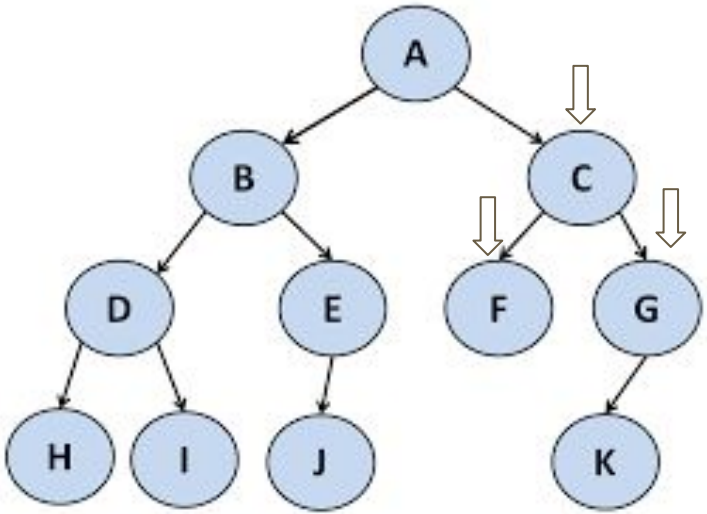
Breadth First Search

Goal: I

Current: C

Frontier: [D, E, F, G]

Visited: [A, B, C, D, E, F, G]



Breadth First Search

Goal: I

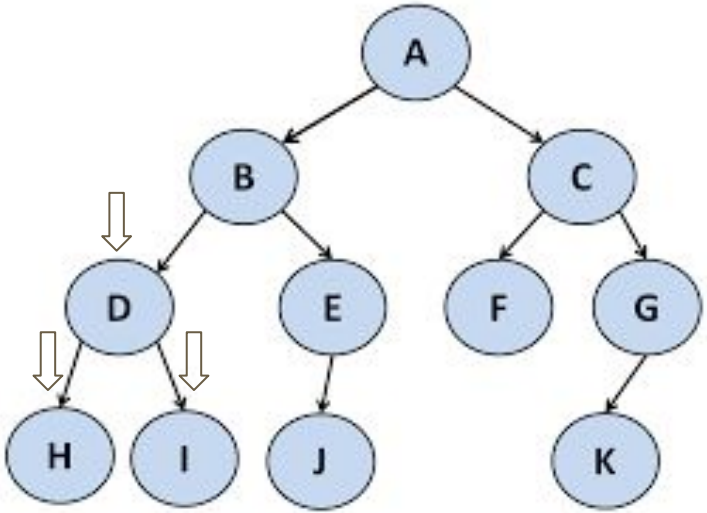
Current: D

Frontier: [E, F, G, H, I]

Visited: [A, B, C, D, E, F, G, H, I]

One of currents children is the Goal State. Follow back up the tree and return the path to I

[A, B, D, I]



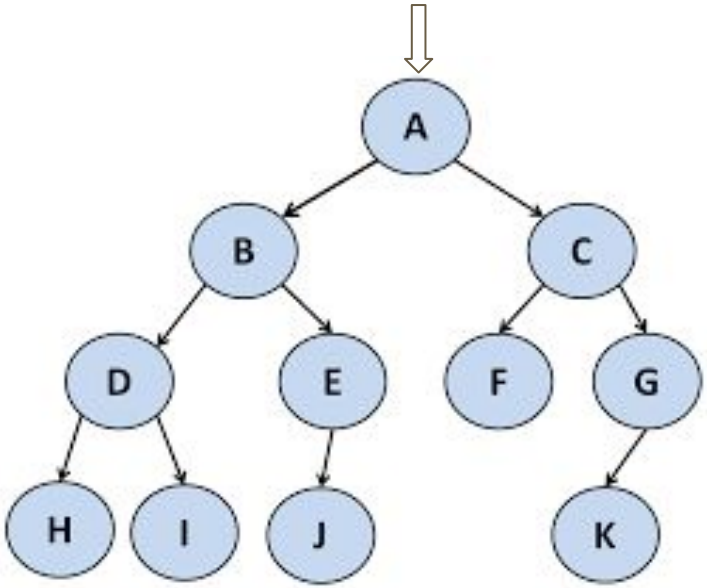
Depth First Search

A tree/graph search algorithm that starts at an arbitrary point in the graph and explores all the way down one branch until the program hits an edge

Goal: J

Current: A

Visited: [A]

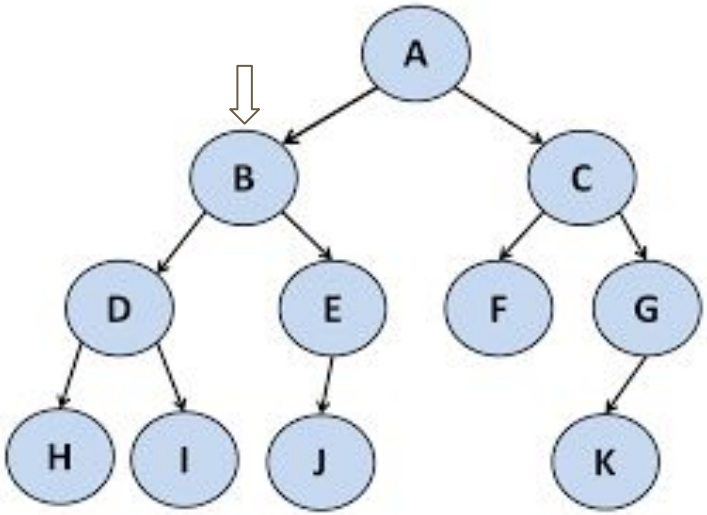


Depth First Search

Goal: J

Current: B

Visited: [A, B]

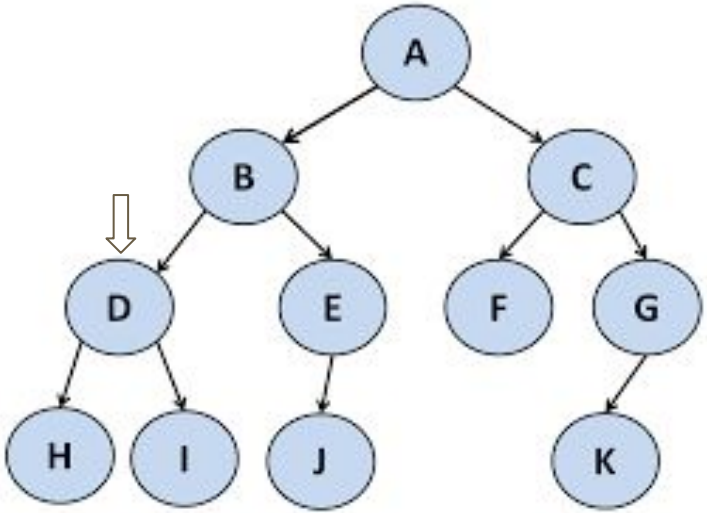


Depth First Search

Goal: J

Current: D

Visited: [A, B, D]

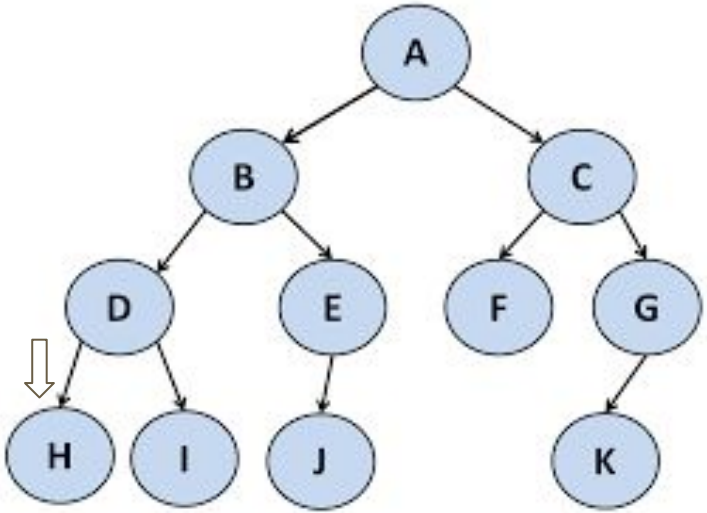


Depth First Search

Goal: J

Current: H

Visited: [A, B, D, H]

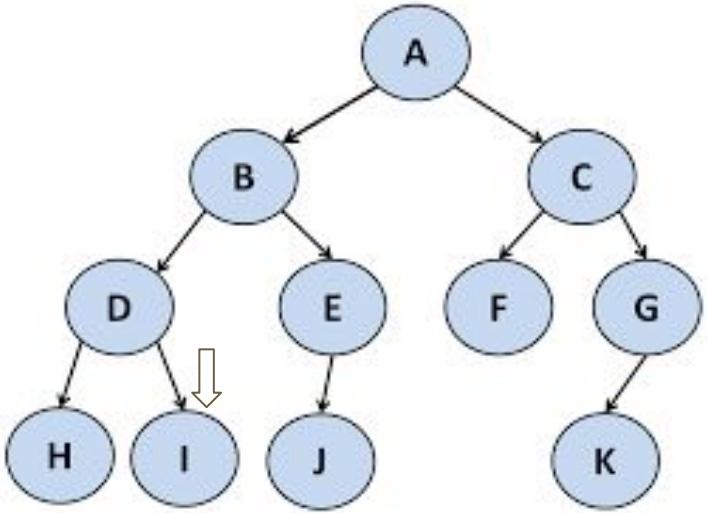


Depth First Search

Goal: J

Current: I

Visited: [A, B, D, H, I]

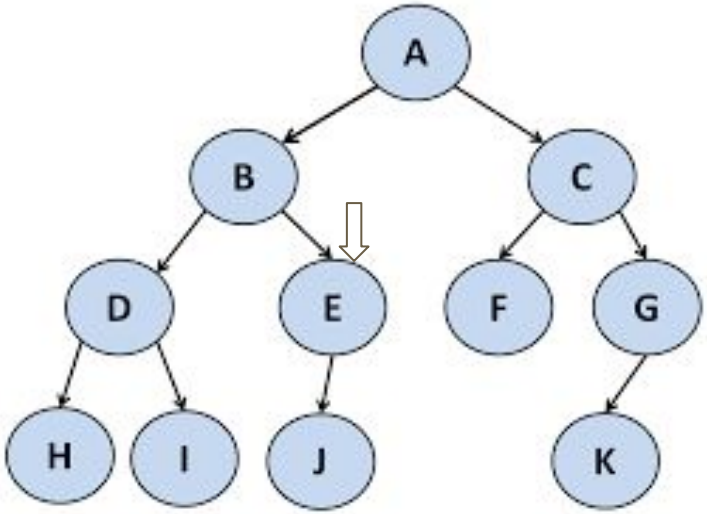


Depth First Search

Goal: J

Current: E

Visited: [A, B, D, H, I, E]



Depth First Search

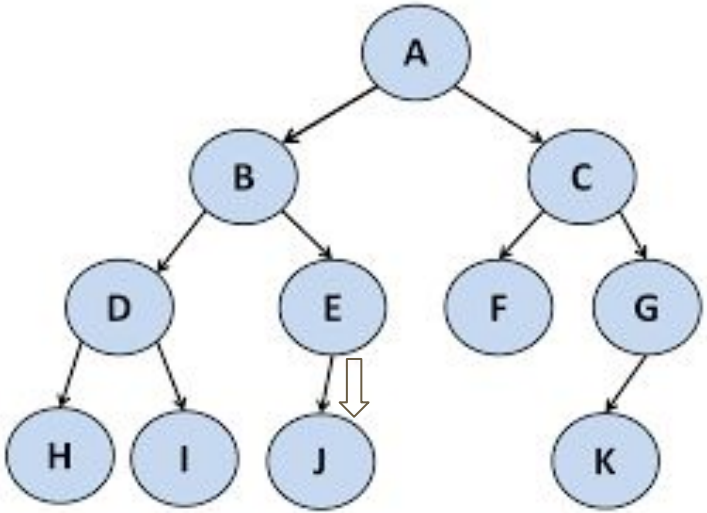
Goal: J

Current: J

Visited: [A, B, D, H, I, E, J]

As Current = Goal, the program should return the path to the goal:

[A, B, E, J]



Breadth First Search vs. Depth First Search

BFS

- requires more memory
- Returns the shortest path to the goal

DFS

- performs quicker
- DFS is not guaranteed to find/return an answer

Conclusion

- Time Complexity - O(n)
- List Search
  - Linear/Sequential Search
  - Binary Search
- Graph Search
  - Breadth First Search
  - Depth First Search
- Exercises
- Code Review

AAR / Questions