CONTINUING EDUCATION: STATE SPACE SEARCHING

Peter Jang

Agenda

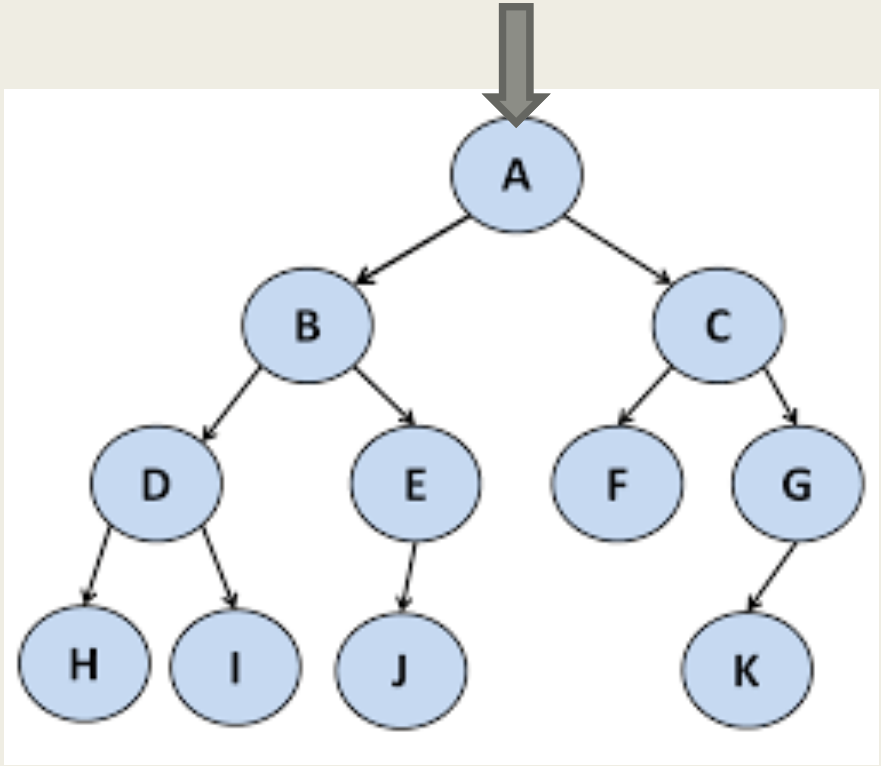
- Introduction
- Review of Breadth First Search
- Object Oriented Programming
- Representing States
- Test and Generate
- Searching through States
- Exercises

GitHub Repository for Exercises

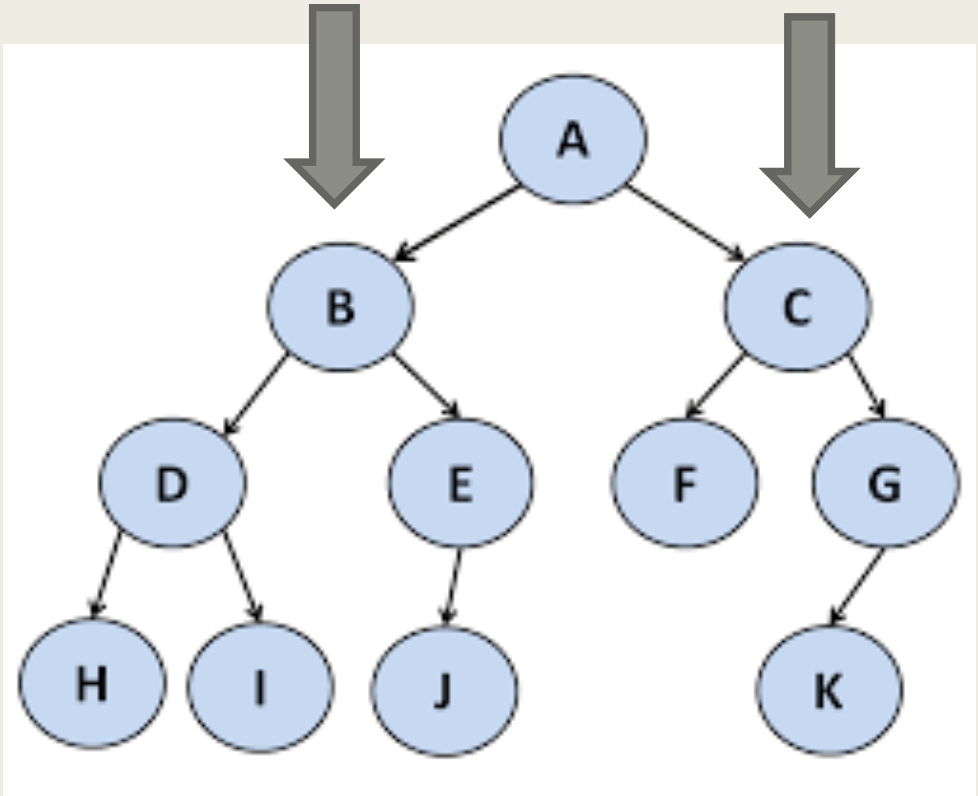
https://github.com/byujan/statespace_searching

Review of Breadth First Search

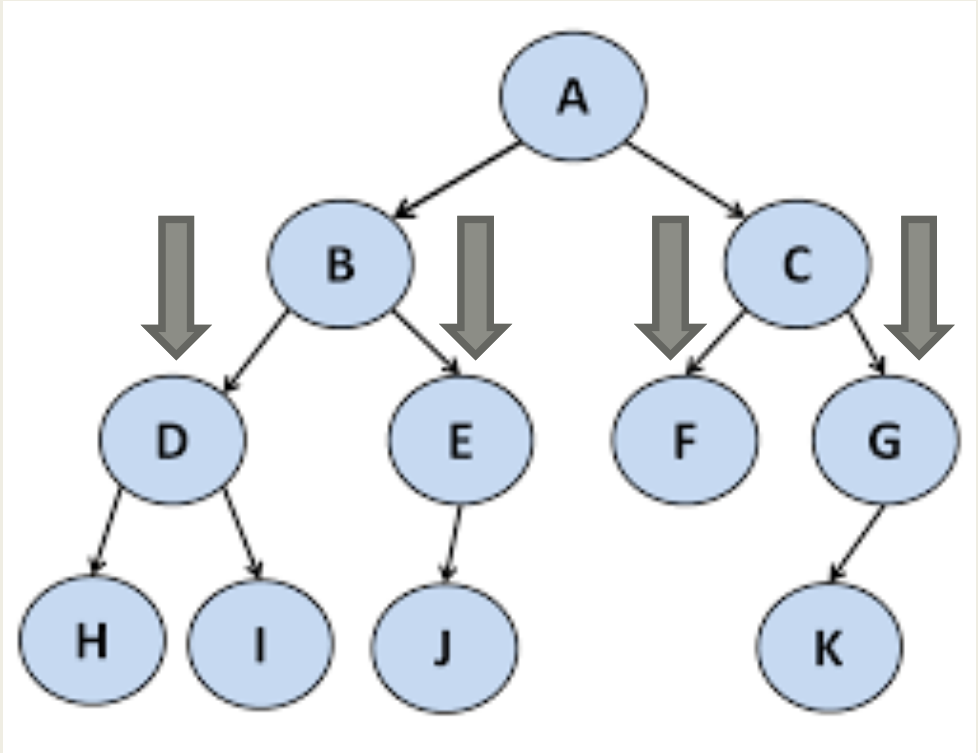
Searches children nodes before searching deeper into the tree



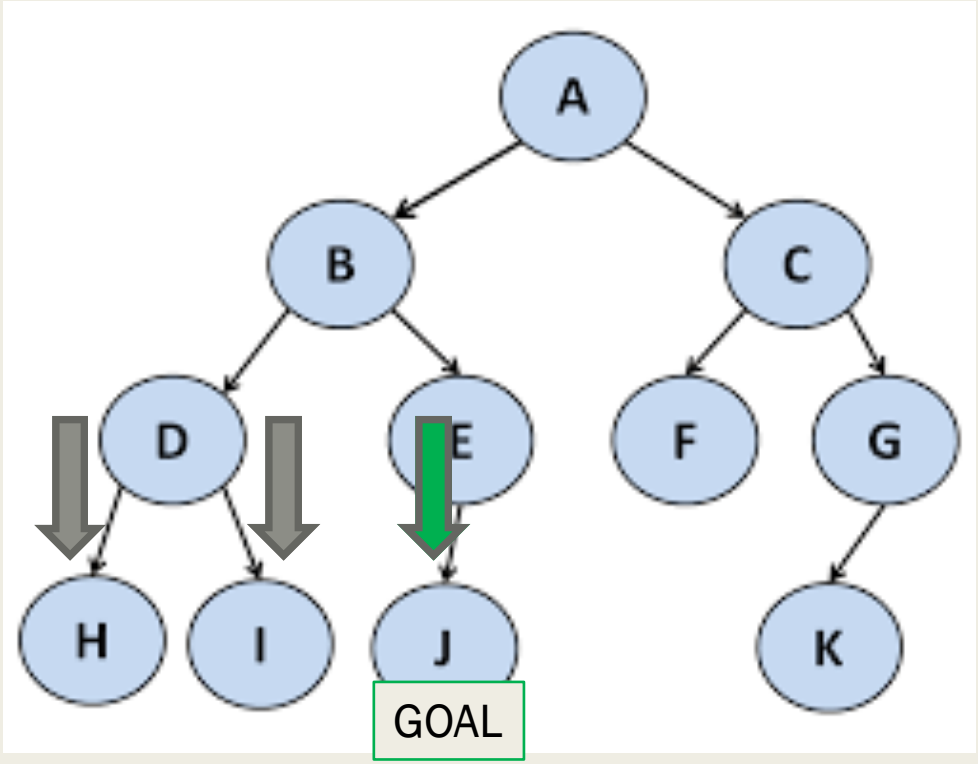
BFS



BFS



BFS



BFS Pseudocode

```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier ← a FIFO queue with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier)  /* chooses the shallowest node in frontier */
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
        frontier ← INSERT(child, frontier)
```

Figure 3.11 Breadth-first search on a graph.

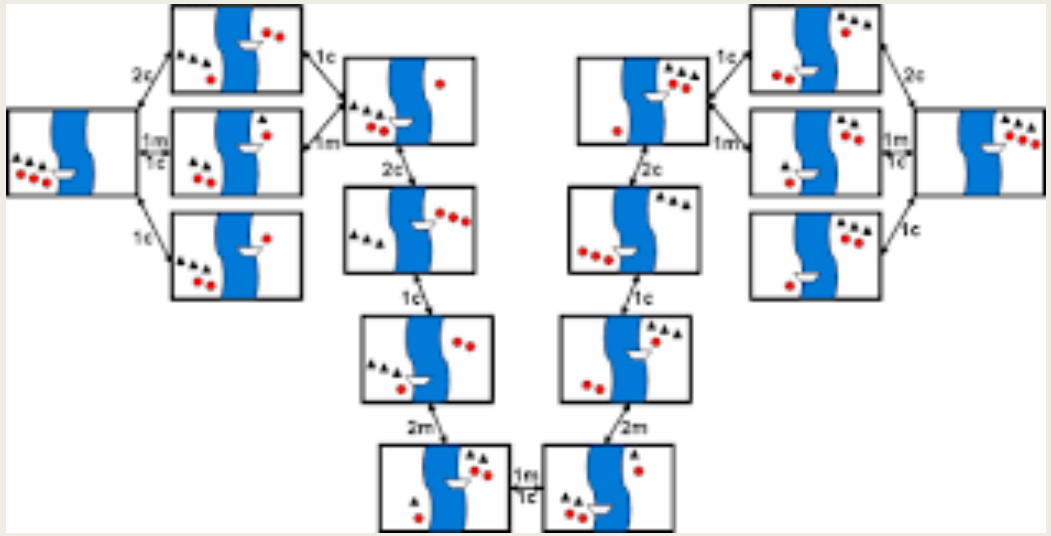
Introduction

State Space Searching (SSS) is the consideration of successive states with the intention of finding a desired goal/state

Heavily Reliant on:

1. Object Oriented Programming
2. Searching Algorithms

Examples of State Space Searching



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Object Oriented Programming

Classes are a general blueprint that contain *properties and methods* to be realized in class objects

Properties are attributes given to a class that describe the object

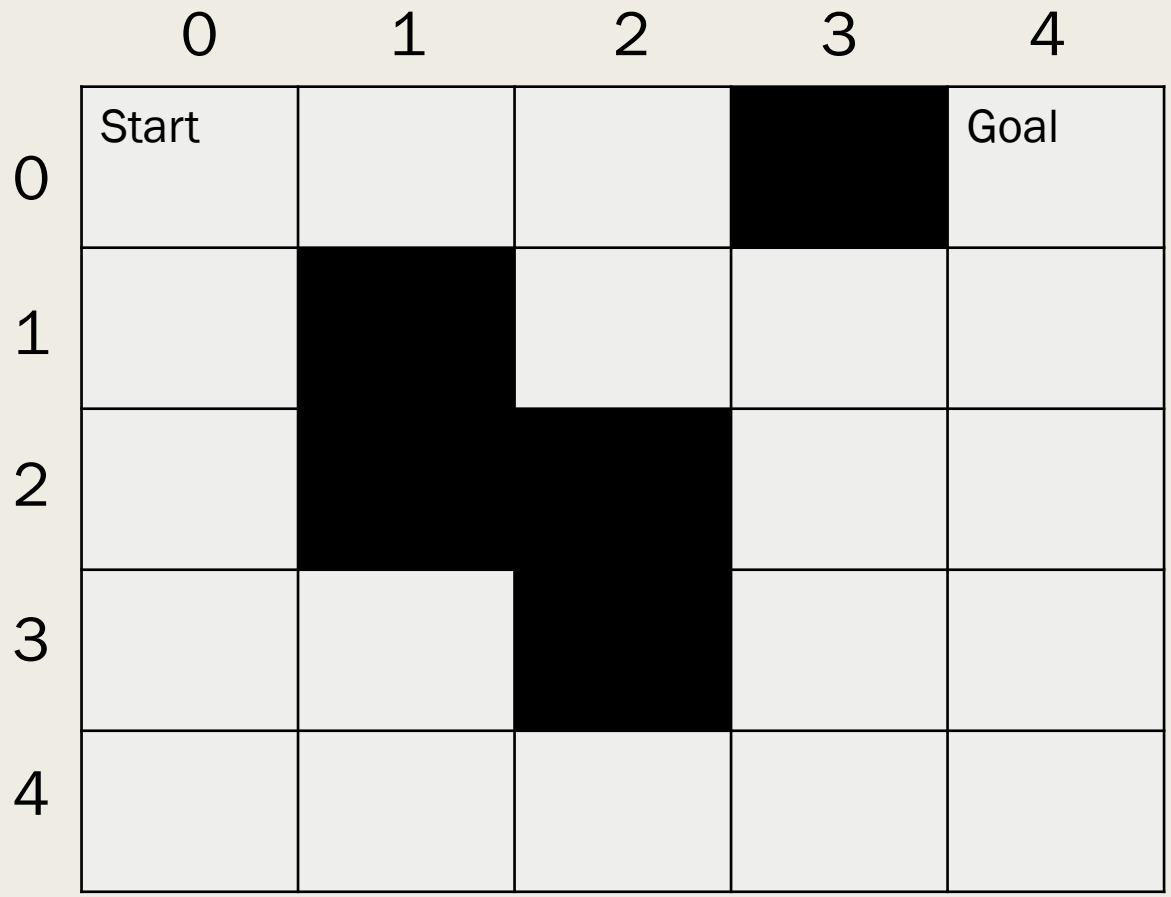
Methods are functions that your class object can perform

- Constructors

OOP Exercises

- Current Position
- Getting the other Positions
- Length of a List
- List of Positions
- Checking a Valid State
- Getting the direction to another Position

Maze State Space Searching



Representing States: Properties

Classes are used to represent States

```
def __init__(self, parent, pos, move, maze):
        self.parent = parent
        self.pos = pos
        self.maze = maze
        self.move = move
```

Representing States: Methods

get_children(self)

- A method to generate successive states to search through

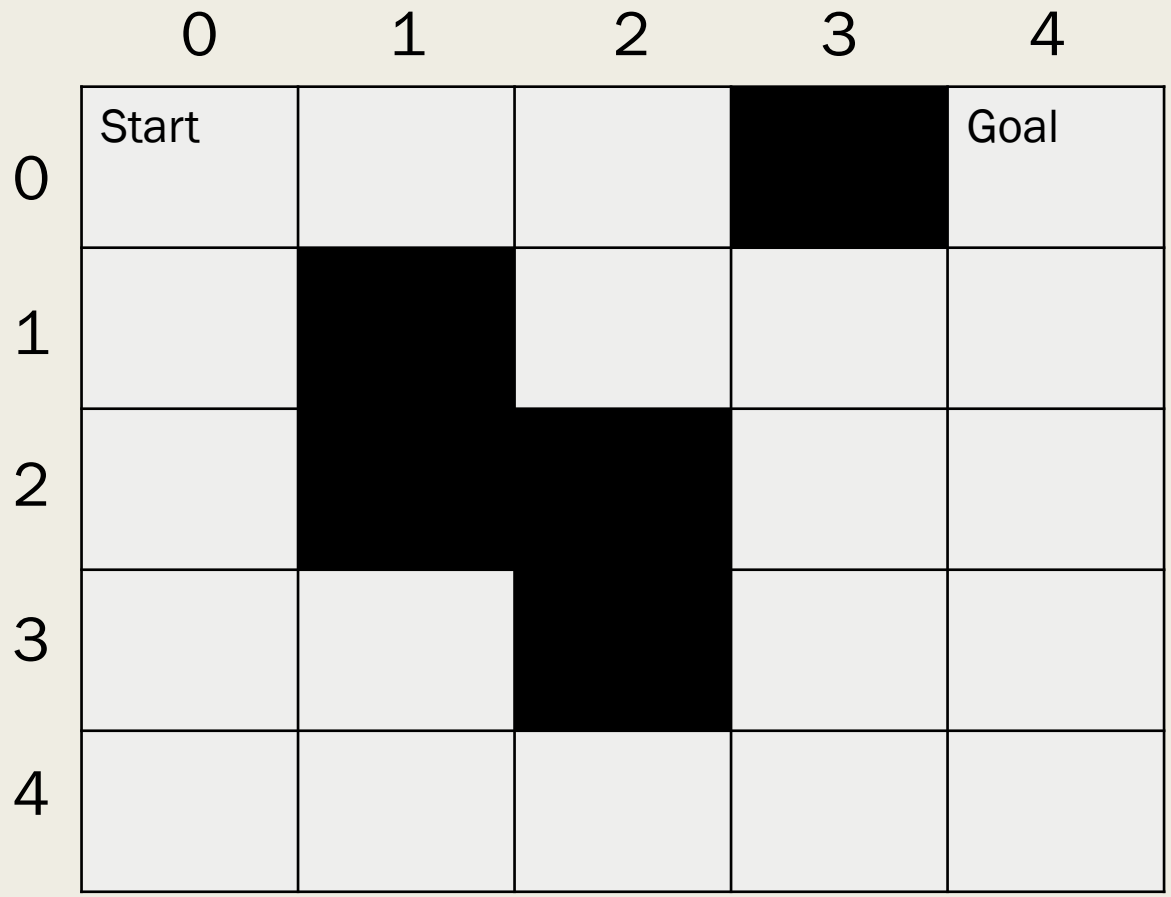
isValid(self)

- A method to check if the states are valid

Test and Generate

- Generate all possible successive states
- Test to see if those moves are valid
- If not valid, do not search the state
  - Decreases searchable states
  - Decreases run time

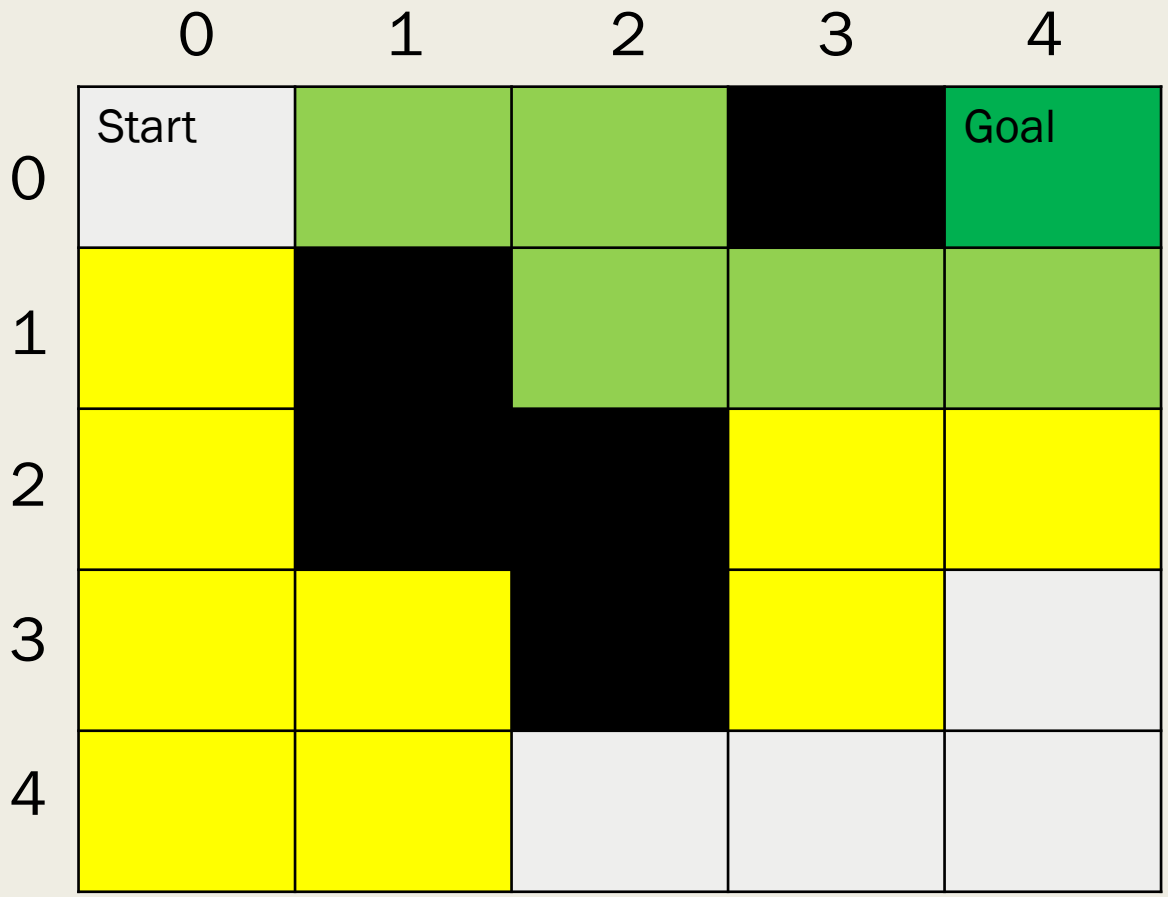
Maze State Space Searching



Test and Generate

- Test Possible Moves from (0,0)
  - [(-1,0), (1,0), (0,-1), (0,1)]
- Generate New States
  - State Position (0,1)
  - State Position (1,0)

Searching through States



Programming Exercises

Conclusion

- Review of Breadth First Search
- Brief overview of Object Oriented Program
- Maze Traversal
  - *State Representation*
  - *Test and Generate*
  - *Searching through States*
- Program Exercises

Questions / AAR