Pandit Deendayal Petroleum University

School of Technology

Artificial Intelligence

BTech ICT/CSE Semester VI, Winter Semester 2018-19

Lab 7: Problem Solving using Search Algorithms

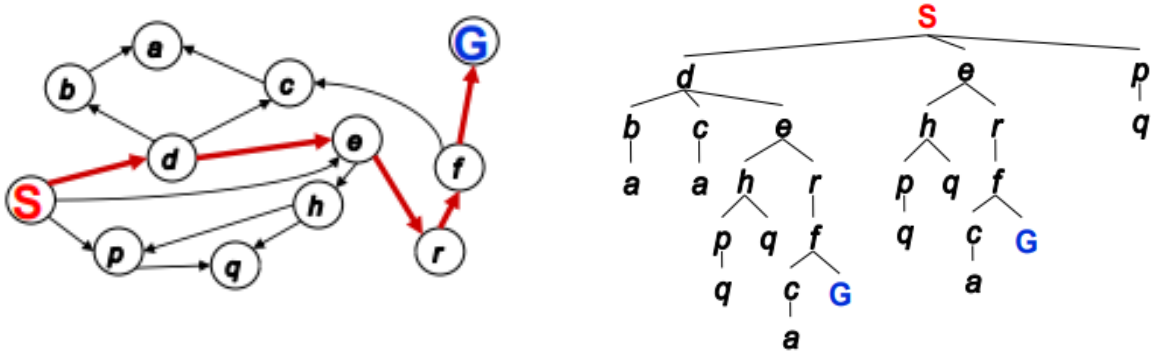
Preamble

An important aspect of intelligence is goal-based problem solving. A wide range of real world problems can be formulated as searches more precisely, as the process of searching for a sequence of actions that take you from an initial state to a goal state. A well-defined problem can be described by:

- Initial state
- Operator or successor function - for any state x returns s(x), the set of states reachable from x with one action
- State space - all states reachable from initial by any sequence of actions
- Path - sequence through state space
- Path cost - function that assigns a cost to a path. Cost of a path is the sum of costs of individual actions along the path
- Goal test - test to determine if at goal state

State Space Graph and State Space Tree

The search problem is formulated as a search graph G=(V, E), where each node of the graph represents a state and the arc (a, b) between the nodes indicates that state a is the resultant of successor function from state b. A search tree of the graph represents the possible actions that help in arriving the goal state. Root node of the tree corresponds to start state and the edges correspond to actions and costs. The following figure illustrates a search graph and its associated search tree for the path shown in red color.



There are many search algorithms to determine the path between start and goal states. This assignment focuses on studying Breadth-First-Search (BFS) and Depth-First-Search (DFS) algorithms and applying them to solve two well known AI problems : 8-puzzle problem and Pac-Man game.

The 8-puzzle problem

The 8-puzzle problem is a puzzle invented and popularized by Noyes Palmer Chapman in the 1870s. It is played on a 3-by-3 grid with 8 square tiles labelled 1 through 8 and a blank square. Your goal is to rearrange the tiles so that they are in order, using as few moves as possible. You are permitted to slide tiles horizontally or vertically into the blank square.

There are four possible actions in the 8-puzzle problem as defined below:

- move empty space (blank) to the left,
- move blank up,
- move blank to the right
- move blank down,. These moves are modeled by production rules that operate on the state descriptions in the appropriate manner.

The following shows a sequence of legal moves from an *initial state* (left) to the *goal state* (right).

	1	3
4	2	5
7	8	6

1		3
4	2	5
7	8	6

1	2	3
4		5
7	8	6

1	2	3
4	5	
7	8	6

1	2	3
4	5	6
7	8	

Pac-Man Problem

The Game Pac-Man is a both challengeable and satisfactory video game that has been the focus of some important AI (Artificial Intelligence) research. The typical version of Pac-Man is a one-player game where the human player maneuvers the Pac-Man character around a maze, attempting to avoid "ghost" characters while eating dots initially distributed throughout the maze as shown in Figure 1.

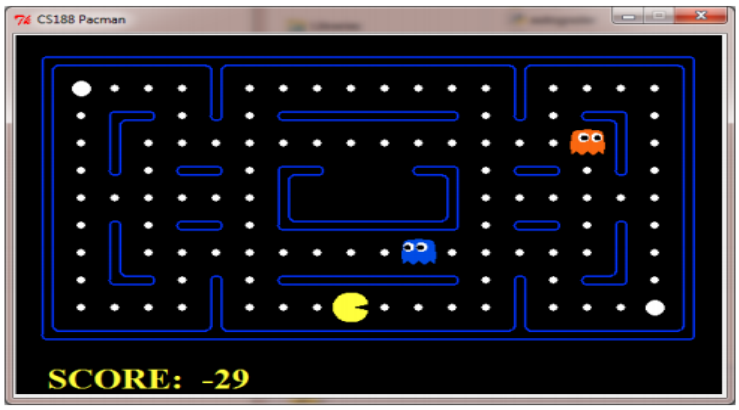
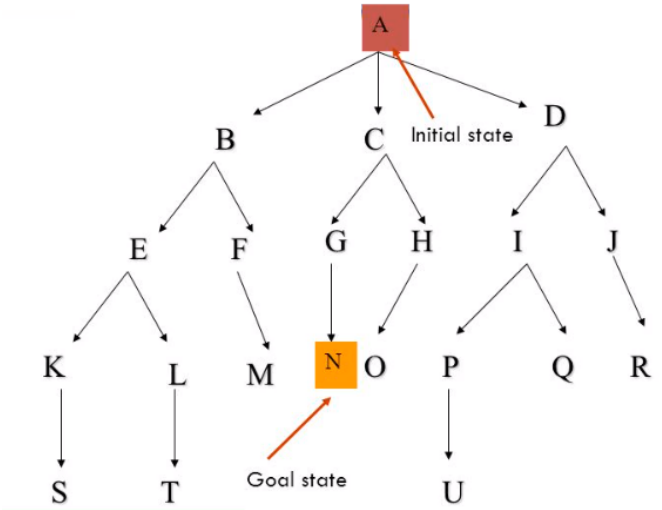


Figure 1: The Pac-Man World

State space search algorithms can be used here to help the Pac Man to find paths through his maze world, both to reach a particular location and to collect food efficiently [1].

Exercise 1:

a.) Apply BFS and DFS algorithms on the following search space to find the path between start and goal states. Also compute the cost of both searches in terms of number of nodes visited.



b.) Write a function bfs(T, s, g) that performs BFS on the given problem's search space T and return the path from starting state s to goal state g.

c.) Write a function dfs(T, s, g) that performs DFS on the given problem's search space T and return the path from starting state s to goal state g.

d.) Use the above graph as test case and verify your results.

Exercise 2:

a.) Given below are the initial and final states of an instance of 8-puzzle problem. Draw the partial search tree for the same indicating a sequence of moves leading from the initial to the goal state.

Initial state

	5	2
1	8	3
4	7	6

Goal state

1	2	3
4	5	6
7	8	

b.) Write a program to solve the 8-puzzle problem using the functions that you have written for BFS and DFS. The program should be generic for any configuration of initial and goal states.

c.) Test your program with above configuration of initial and goal states.

Exercise 3:

Let us consider a variant of PacMan problem where goal is to locate the food and there are no ghosts. You may use a matrix to represent the Maze as shown below where character P represents the Pac Man, character '*' represents food , character '#' represents wall and character '-' represents free space for movement. Pac Man can not move when there is wall.

#	#	#	#	#	#	#	#
#	-	-	-	-	-	-	#
#	-	#	#	#	-	-	#
#	-	#	*	#	-	-	#
#	-	-	-	#	-	-	#
#	-	#	#	#	-	-	#
#	P	-	-	-	-	-	#
#	#	#	#	#	#	#	#

Maze size : 8 x 8

Initial state : Position of Pac man : (7 , 2)

Goal state : Position of Food : (4, 4)

Next Possible move for Pac Man :

{(7,3), (6,2)}

a.) Draw the search tree for the above Pac Man Game configuration. Also identify the paths from initial to goal state using both BFS and DFS algorithms.

b.) Write a program to help the Pac Man to locate food using the search functions algorithms that you have written.

c.) Using the sample configuration shown above, check which search algorithm helped Pac Man to locate the food efficiently

Image Segmentation

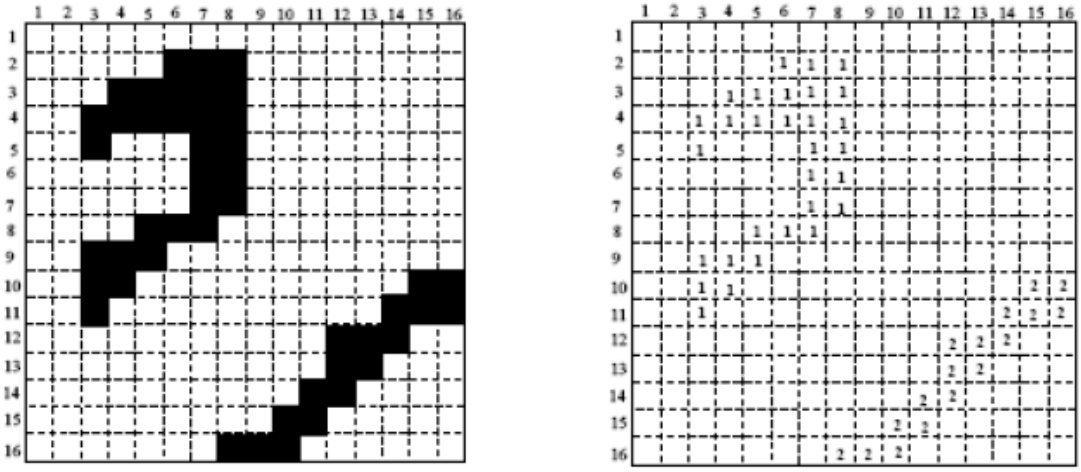
Image segmentation is the process of partitioning a digital image into multiple segments (or regions), which is a widely used pre-processing step in various computer vision applications including object recognition. An example of 16 x 16 binary image and its corresponding labelled image with two regions are shown in Figure 2.

Procedure to find connected components or regions (assuming regions are defined by white pixels)

1. Scan the input binary image to find an unlabelled white pixel and assign it to a new Label L.

2. Recursively assign the label L to all of its white pixel neighbours.

3. Stop if there are no more white pixels; else go to step 1.



Input: binary Image M

Output: Labelled binary Image

Figure 2.

The neighbors of the pixel M(i, j) are simply the closest pixels to it. The 8-neighbors of pixel M(i, j) is shown below

M(i-1, j-1)	M(i-1, j)	M(i-1, j+1)
M(i, j-1)	M(i, j)	M(i, j+1)
M(i+1, j-1)	M(i+1, j)	M(i+1, j+1)

Exercise 4:

Implement a simple system that automatically counts the number of regions in the sample binary images attached with this. Output of the program should be the number of regions along with a labelled image. You may use DFS or BFS algorithms to find the neighbours.

References :

1. http://inst.eecs.berkeley.edu/~cs188/pacman/project_overview.html