National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 1: Agents, Problems, and Uninformed Search

Issued: January 17, 2022 Discussion in: Week 3

Important Instructions:

- **Assignment 1** consists of **Questions 4 and 5** from this tutorial.
- Your solutions for this tutorial must be TYPE-WRITTEN.
- You are to submit your solutions on LumiNUS by **Week 2, Sunday, 2359 hours**.
- Refer to LumiNUS for submission guidelines

Note: you may discuss the content of the questions with your classmates, but you must work out and write up your solution individually. Solutions that are plagiarised will be heavily penaltised.

---

1. Sudoku is a popular number puzzle that works as follows: we are given a 9×9 square grid; some squares have numbers, while some are blank. Our objective is to fill in the blanks with numbers from $1 - 9$ such that each row, column and the highlighted 3×3 squares contain no duplicate entries (see Figure 1).

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

Figure 1: A simple Sudoku Puzzle

Sudoku puzzles can solved easily after being modelled as a CSP (which we will cover later in the module). We will consider the problem of *generating* Sudoku puzzles.

One possible procedure for doing this is to start with a completely full number grid (see Figure 2), and iteratively make some squares blank. We continue blanking out squares as long as the resulting puzzle can be completed in only one way.

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

Figure 2: Solution to the Sudoku puzzle in Figure 1.

(a) Determine the properties of the above problem from the perspective of an intelligent agent planning a solution. Complete the table below.

The completed table is as follows.

	Environment Characteristic	**Sudoku Puzzle**
1	Fully vs Partially Observable	
2	Deterministic vs Stochastic	
3	Episodic vs Sequential	
4	Discrete vs Continuous	
5	Single vs Multi-Agent	
6	Static vs Dynamic	

(b) Define the search space for the problem of generating a Sudoku Puzzle by completing the following.

- Give the representation of a state in this problem.
- Using the state representation defined above, specify the initial state and goal state(s).
- Define its actions.
- Using the state representation and actions defined above, specify the transition function T. (In other words, when each of the actions defined above is applied to a current state, what is the resulting state?).

2. (a) Describe the difference between **Tree Search** and **Graph Search** algroithms.

(b) Assuming that ties (when pushing to the frontier) are broken based on alphabetical order, specify the order of the nodes checked (i.e., via the goal test) by the following algorithms. Assume that S is the initial state, while G is the goal state.

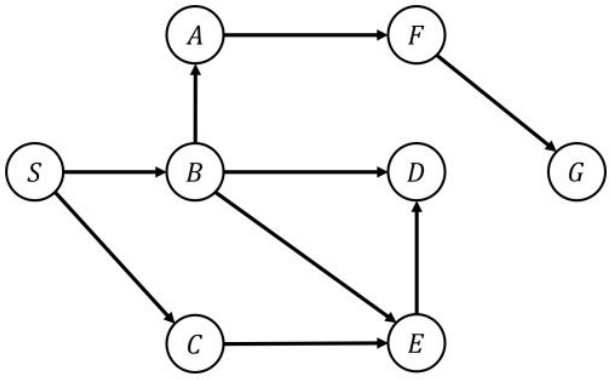

Figure 3: Graph for Question 2b.

You should express your answer in the form S–B–A–F–G (i.e., no spaces, all upper-case letters, delimited by the dash (–) character), which, for example, corresponds to the order S, B, A, F, and G.

i. Depth-First Search with tree-based implementation
ii. Depth-First Search with graph-based implementation
iii. Breadth-First Search with tree-based implementation
iv. Breadth-First Search with graph-based implementation

3. Prove that the **Uniform-Cost Search** algorithm is optimal as long as each action cost exceeds some small positive constant ϵ.

4. You are given an n-piece unassembled jigsaw puzzle set (you may assume that each jigsaw piece can be properly connected to either 2, 3, or 4 pieces), which assembles into an $(m \times k)$ rectangle (i.e., $n = m \times k$). There may be multiple valid final configurations of the puzzle. Figure 4 illustrates an example.

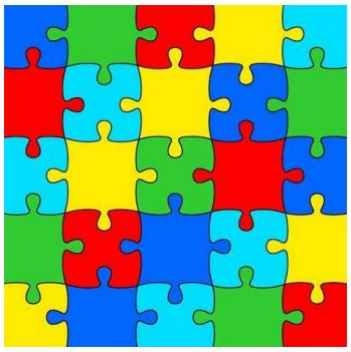

Figure 4: A sample configuration of the jigsaw puzzle.

Formulate the above as a search problem. More specifically, define the following:

- State representation
- Initial state
- Actions
- Transition model
- Step cost
- Goal test

If necessary, you may identify the assumptions you have made. However, assumptions that are contradictory to any instruction in the question, or that are unreasonable, will be invalid.

5. You have just moved to a strange new city, and you are trying to learn your way around. Most importantly, you want to learn how to get from your home at S to the MRT station at G.

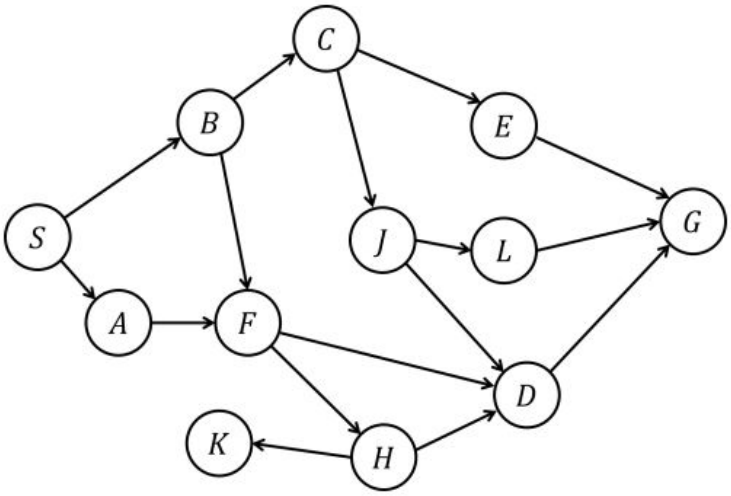

Figure 5: Graphical representation of the city.

Apply the **Depth-First Search** algorithm with **tree-based implementation**. Use alphabetical order to break ties when deciding the priority for pushing nodes into the frontier.

Determine the final path found from the start (S) to the goal (G)? Note that you **MUST** express your answer in the form S–B–C–J–L–G (i.e., no spaces, all uppercase letters, delimited by the dash ($--$) character), which, for example, corresponds to the exploration order of S, B, C, J, L, then G.