National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 1: Agents, Problems, and Uninformed Search

Issued: Week 2 Discussion in: Week 3

Important Instructions:

- **Assignment 1** consists of **Question 5** from this tutorial.
- Your solution(s) must be TYPE-WRITTEN, though diagrams may be hand-drawn.
- You are to submit your solution(s) during your **Tutorial Session in Week 3**.

Note: you may discuss the assignment question(s) 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 cells contain numbers, while some are blank. Our objective is to fill in the blank cells 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 be easily solved by modelling them as a constraint satisfaction problem (which we will cover later in the course). We will instead consider the problem of *generating* Sudoku puzzles.

One possible procedure for doing this is to start with a completed grid (see Figure 2), and iteratively make some cells blank. We will continue blanking out cells 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.

	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 above 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.

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 ascending alphabetical order (e.g., A before B), 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.

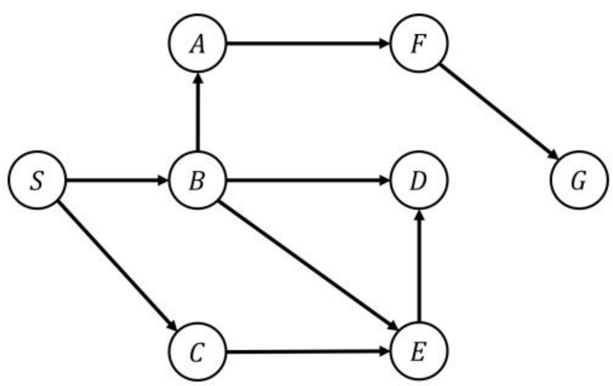

Figure 3: Graph for Question 2b.

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

i. **Depth-First Search** using a **tree search** implementation
ii. **Depth-First Search** using a **graph search** implementation
iii. **Breadth-First Search** using a **tree search** implementation
iv. **Breadth-First Search** with a **graph search** implementation

3. Prove that the **Uniform-Cost Search** algorithm is optimal as long as each action cost exceeds some small positive constant ϵ. You may also assume the same constraints that make Breadth-First Search complete.

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.

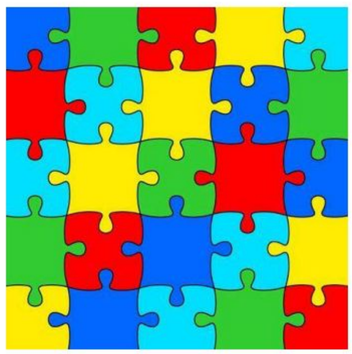

Figure 4: A sample configuration of the jigsaw puzzle.

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. More specifically, you wish to learn how to get from your home at S to the train station at G.

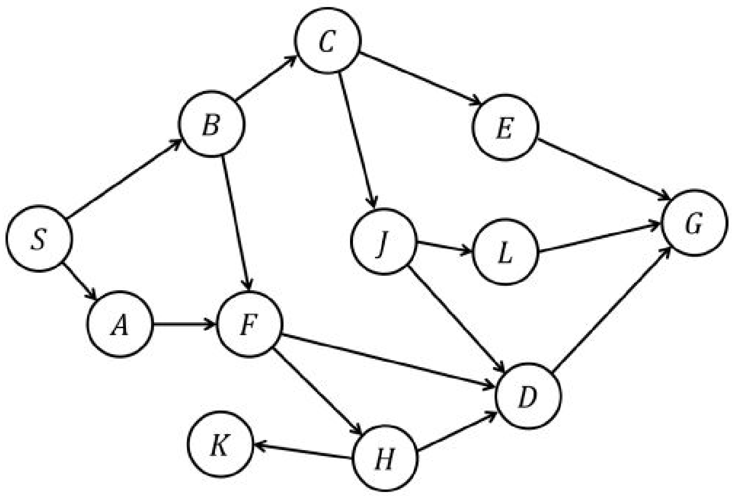

Figure 5: Graphical representation of the city.

Apply the **Depth-First Search** algorithm with a **tree search** implementation. Use ascending alphabetical order to break ties when deciding the priority for pushing nodes into the frontier (i.e., A before B).

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.