CSC384H Tutorial 1

Search Problems & Uninformed Search

Summer 2025

Search Problem

In order to formalize a problem as a search problem, we need to define the following:

1. State Space: A state is a representation of a *configuration* of the problem domain. The state space is the set of all states included in our model of the problem.
2. Initial State: The starting configuration.
3. Goal State: The configuration one wants to achieve.
4. Actions (or State Space Transitions): Allowed changes to move from one state to another.

Optionally, we may use the following:

1. Costs: Representing the cost of moving from state to state.
2. Heuristics: Help guide the heuristic process.

Problem example: **Make Change**.
We start with 0 cents and we want to reach some number of cents between $[0, 5, 10, 15, ..., 500]$ using the least number of coins. We can add $5, 100, 25, 100$, or 200 cents.

Search problem formulation with the goal of X cents:

- States: Integers between 0 and 500 that are divisible by 5.
- Initial State: 0
- Goal State: X
- Actions: Add $5, 10, 25, 100$, or 200

Questions

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).

Sudoku puzzles can be solved easily after being modelled as a CSP (which will be covered later in the course). We will consider the problem of *generating* Sudoku puzzles. In particular, consider the following procedure: start with a completely full number grid (which is a valid sudoku solution) (see Figure 2), and iteratively make some squares blank. We continue blanking out squares as long as the resulting puzzle has exactly one unique solution.

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.

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 puzzle in Figure 1.

Complete 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) Assuming that ties (when pushing to the frontier) are broken based on alphabetical order, specify the order of the nodes that would be explored by the following algorithms. Assume that S is the initial state, while G is the goal state.

   You should express your answer in the form $S - B - A - F - G$ (i.e. only showing the state of each search node, all uppercase letters, delimited by the dash ($-$) character), which, for example, corresponds to the exploration order of S, B, A, F, then G.

   i. Depth-First Search with no cycle-checking implementation.

   ii. Breadth-First Search with no cycle-checking implementation.

   iii. Breadth-First Search with cycle-checking implementation.

   (b) Using the same graph as above, perform Iterative Deepening Search (IDS) with a starting depth limit of 1. Express your answer separated by depth limit in the same form as above $S-B-A-F-G$

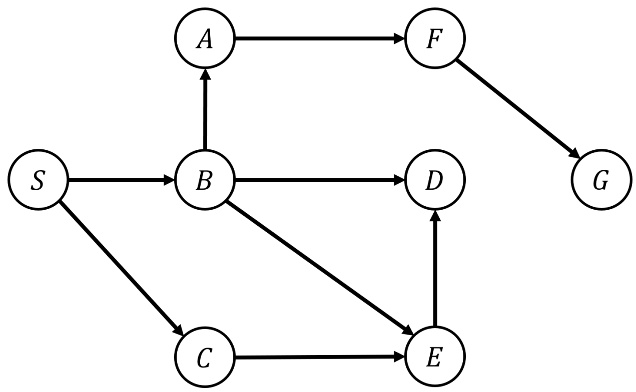

Figure 3: Graph for question 2(b).

3. (a) The **Breadth-First Search** algorithm is complete if the state space has infinite depth but finite branching factor.

   Determine if the statement above is True or False, and provide a rationale.

   (b) The **Breadth-First Search** algorithm can be complete even if zero step costs are allowed.

   Determine if the statement above is True or False, and provide a rationale.

   (c) Given that a goal exists within a finite search space, the **Breadth-First Search** algorithm is cost-optimal if all step costs from the initial state are non-decreasing in the depth of the search tree. That is, for any given level of the search tree, all step costs are greater than the step costs in the previous level.

   Determine if the statement above is True or False, and provide a rationale.

UCS

Consider the search problem of figure 4. Execute UCS (without cycle-checking) starting from node **S**, and the goal node being **G**.

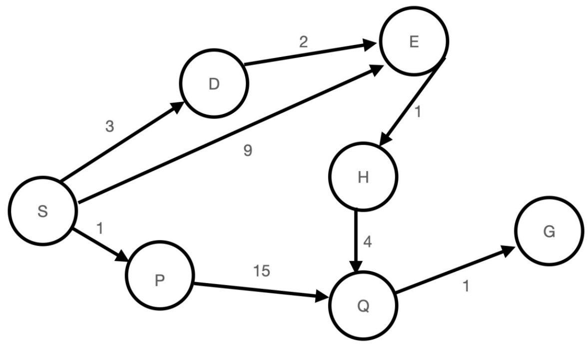

Figure 4: A search problem.