Problem Set 1 - Submission

A0219739N - Le Van Minh

August 30, 2023

1.

(a)

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

(b) Define the search space:

- **State $s_i \in S$:** is the layout of the current Sudoku board, represented as a 2D array
- **Initial state s_0:** A full board with no hole in it, with the rules of Sudoku board satisfied
- **Goal state:** The goal test returns true when there's no more valid action to be taken at the current state
- **Actions:** at each state s, there's a set of available actions A, each action deletes a square that can be perfectly recovered
- **Transition function T:** $T(s_i, a_j) = s_t$ where s_t is the is the same as s_i except a square is blank when it is erased by action a_j

2. (a)
   - **Tree Search:** Explore all paths including redundant paths.
   - **Graph Search:** Only explore unvisited paths (v1) or non-redundant paths (v2).

   (b) Assume early goal checking when pushing the nodes to the frontier:

   i. **DFS (tree):** $S - B - C - E - D - A - D - E - D - D - F - G$
   ii. **DFS (graph):** $S - B - C - E - D - A - F - G$
   iii. **BFS (tree):** $S - B - C - A - D - E - E - F - D - D - G$
   iv. **BFS (graph):** $S - B - C - A - D - E - F - G$

3. Supposed there's an optimal result (v, c) such that v is a goal node with cheapest path cost c. For UCS to be optimal, it has to encounter (v, c) before any possible goal. This is true because the nodes being goal checked by UCS are in cost-increasing order, so if c is the cheapest goal path cost, v is the first goal to be checked.
   Proof of cost-increasing order: When a node v_0 is popped from the frontier F, it adds it

unvisited adjacent nodes V_0 to the frontier. Frontier after the popping $F' = F \cup V_0 \setminus \{v_0\}$. All nodes in F have more expensive path cost than v_0, and each nodes in V_0 has path cost at least as much as $cost(v_0) + \epsilon$, assuming cheapest edge cost $\geq \epsilon > 0$. Therefore, all nodes in the new frontier F' are more expensive than $cost(v_0) \Rightarrow$ the next popped node is more expensive than $v_0(QED)$.

4.
- **State representation:** The current assembling of the jigsaw table
- **Initial state:** The empty jigsaw table
- **Actions:** Given the current state of the puzzle, pick an unused piece that can fit adjacently to the current board, and put it in, or remove one of the connected pieces.
- **Transition model:** Putting the chosen piece into the current table to get the resulting state
- **Step cost:** Actions have equal costs
- **Goal test:** Test if the current board has n correctly assembled pieces

5. Assume early goal check when pushing new nodes to the frontier. The order of goal checking is: $S - A - B - C - F - D - H - D - K - G$