Artificial Intelligence

Assignment 1

Due date: October 15, 2022

Carefully organize your code and put comments Clearly state your assumptions (if any) at the beginning of your code For each problem you have to submit your solution (Python3 code) as a single `.py` file

A1. **Comparing Different Runs of 8-Puzzle** [5]

Take ten different (random) starting states for the 8-Puzzle problem and do the followings:

i. Apply a suitable algorithm to check whether the goal state (of your choice) is reachable or not.

ii. If the goal is reachable, report the number of moves required to reach that goal; report 'Infinity' otherwise.

iii. Your code should first output the goal state, and then for each of the ten start states it would report the required moves to reach the goal along with that start state.

A2. **Comparing Different Solutions of 8-Puzzle** [5]

i. For the 8-Puzzle problem implement the following solution strategies:

1. Simple BFS,
2. A* (with Manhattan distance heuristic), and
3. A* (with misplaced tiles heuristic also known as Hamming distance heuristic)

ii. Consider a random goal state. Among all possible starting states of the 8-Puzzle problem, determine how many of them are solvable against your choice of goal. Print your goal state and the number of solvable states.

iii. For each of the aforementioned three strategies, report the average number of states explored considering all the solvable states. Thus, your code should print three numeric values corresponding to the three strategies.

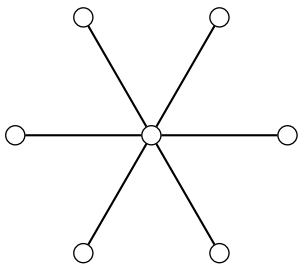
A3. **Solving the Channel-Coloring Problem** [5]

A *channel-coloring* of an undirected graph G is a vertex coloring of G where two vertices can share the same color only if they are at a distance greater than two. It has application in allocation frequency channels for wireless communications. Note that the distance between two vertices in a graph is expressed as the number of edges in the shortest path joining the two vertices; if no such path exists, the distance is assumed to be infinite, and the vertices may get the same color.

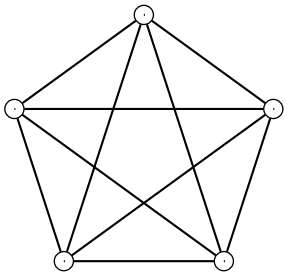
A *channel-coloring number* of an undirected graph G, is the minimum number of colors required to obtain a valid channel-coloring of G.

i. Use any one of the searching algorithm to determine the channel-coloring number of a given graph. Clearly state your assumptions in the code.

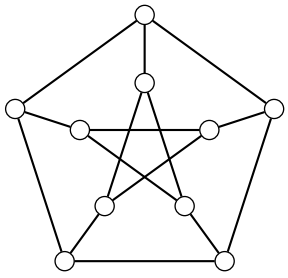
ii. Apply your algorithm on the following three graphs, and report its channel-coloring number.



Star Graph ($K_{1,6}$)



Complete Graph (K_5)



Petersen graph

A4. **Solving the Channel-Coloring Problem using CSP Solver** [5]

i. Formulate the channel-coloring Problem as a Constraint Satisfaction Problem (CSP)

ii. Use any solver to encode your formulation, so that for any undirected graph G it should output the valid channel-colorings of G.

iii. Apply the above CSP code to deduce the channel-coloring number of the above three graphs.