Tutorial-4

1. What are the five components to design a problem. Define these components for the following
a) 8-puzzle problem
b) Travelling Salesman problem.

→ A problem can be formulated using these components:
1) States
2) Initial State
3) Actions
4) Transition model
5) Goal test.
6) Path cost (Optional).

a) 8 puzzle problem:
- States: A state specifies the location of each of the 8 tiles and blank in one of the 9 squares.
- Initial State: Any state can be designated as initial state.
- Actions: In this ~~no~~ problem action is movement of blank space left, right, ~~top~~ up or down.
- Transition model: Applying a specific action on a given state, this returns the resulting state.
- Goal test: This checks whether the state matches goal configuration i.e. all tiles are at right place or not.
- Path cost: Each step costs 1, thus the path cost is number of steps in path.

b) Travelling Salesman Problem:
- States: All cities acts as node in graph so there are two ~~st~~ possible states for each node visited or not visited.
- Initial State: Initially all nodes are not visited.
- Action: In this problem action is movement from one city to another which are not-visited previously

- ___Transition model___: Applying a specific action on a given state, the current city is marked as visited
- ___Goal test___: This checks whether state matches goal configuration or not i.e all cities are visited or not.
- ___Path cost___: Each step cost equal to weight of edge between two nodes, thus path cost is summation of all weights in path.

2. Which parameters are used for measuring the performace of the solved problem. Design a solution for solving "Color Map Problem" and comment on its performance parameters.

→ - Performance of a solution can be evaluated using these four parameters:

1) Completeness
2) Optimality
3) Time Complexity
4) Space Complexity

- The solution for four colour map problem can be derieved by representing the map as a graph in which region represents node & edges represents the ~~neigh~~ adjacent regions. Then graph colouring algorithm is used to assign colours to each node such that different colour should be assigned to each adjacent node.
- It iteratively assigns the colors to nodes based on colors of its neighbour ~~[illegible]~~ until all nodes are coloured.
- It provides completeness but sometimes for more densed map this algorithm ~~is not~~ may not give optimal ~~complete as it may require more than four color.~~ solution.

- ~~It may be optimal as we can use some better approach for more densed graph~~
- Time complexity is $O(b^{b+m})$
- space complexity is $\cancel{O(b^{b})}$ ~~or~~ $\cancel{O(b^{b+1})}$ $O(b+m+1)$.

3. a) We will define the coordinate system so that the center of the maze is at (0,0) and the maze itself is a square from (-1,-1) to (1,1). Problem formulation:
- Initial state: Robot at (0,0) facing North
- Goal test: Either $|x|>1$ or $|y|>1$
- Successor function: Move forward in any direction N-S-E-W
- Cost function: Total distance moved by robot.

State space is infinitely large since robot's position is continuous.

b) The state will record the intersection of the robot currently at along with the direction it is facing at the end of each corridor leaving the maze we will have exit node. Assume some node corresponds to centre of maze.
- Initial state: ~~con~~ Robot is at center facing ~~at~~ North
- Goal test: At an exit node
- Successor f^n: Move to the next ~~inters~~ intersection in front of us if there is one turn to face a new direction.
- Cost function: Total distance.

In this also state space is infinitely large but it is small as compared to previous one as robot move in single dir^n till intersection of corridor.

c) - Initial state: centre of maze
- Goal test: At an exit node
- Successor function: can move in any direction N-S-E-W.
- Path cost: Total distance it moved.

We no longer need to keep track of the robot's orientation since it's irrelevant to predict the outcome of actions & not part of goal test.