MADHUSHALINI MURALI/ A20513784/ mmurali5@hawk.iit.edu

CS 480 Fall 2022 Written Assignment #01

Due: **Tuesday, September 20, 11:00 PM CST**
Points: **20**

Instructions:

1. Use this document template to report your answers. Name the complete document as follows:

```
LastName_FirstName_CS480_Written01.doc
```

2. Submit the final document to Blackboard Assignments section before the due date. No late submissions will be accepted.

Objectives:

1. (8 points) Analyze an agent-environment system and apply the PEAS agent description.
2. (12 points) Demonstrate your understanding of a simple informed search algorithm.

Problem 1:

Consider the robotic Rubik's cube solver shown on below (fig. 1). You are welcome to watch a short video about its development online at https://www.youtube.com/watch?v=x4O8pojMF0w.

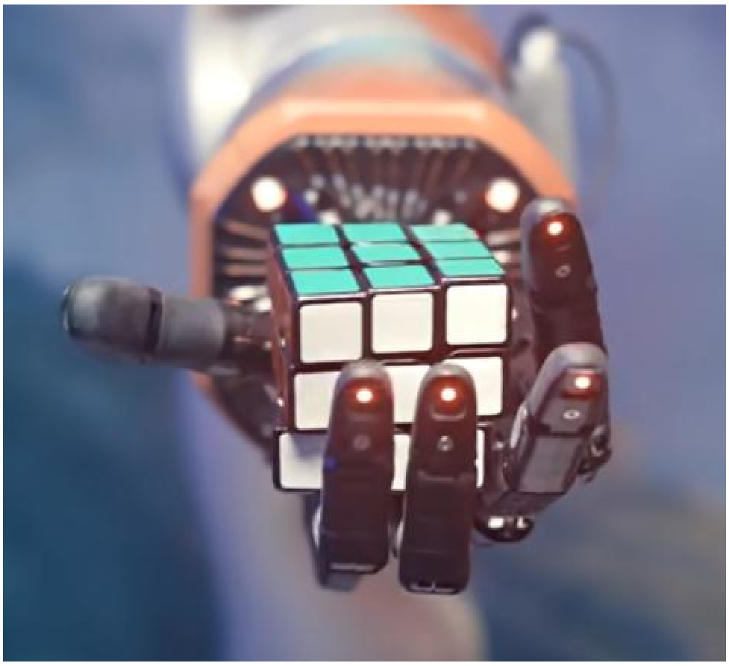


Figure 1: Open AI robotic Rubik's cube solver (screen shot from: https://www.youtube.com/watch?v=x4O8pojMF0w).

Your task is to:

- decide what is the agent and what is the environment in this system **[1 pt]**:

Agent	**Environment**	**Explanation**
Agent is the **Robot-arm**	Environment is the **Rubik's Cube** .	Agent is the **Robot-arm** which performs the action. Environment is the **Rubik's Cube** on which the action is being performed.

- analyze the system and apply the the PEAS (Performance measure, Environment, Actuators, Sensors) description **[3 pts]**:

Performance measure	**Environment**	**Actuators**	**Sensors**	**Explanation**
Accuracy in solving the Rubik's cube, time taken to solve, Mechanism of the Robot-arm in handling the cube without dropping it.	Rubik's cube placed on the Robot's hand.`	Coordination movement of fingers, joints and wrist to keep the cube inside the hand without falling. Accelerator to make the movement on the arm.	Vision camera, Motion sensor,air pressure sensor.	The ultimate goal of the Robot's arm is to finish solving the Rubik's cube with accuracy and within the stipulated time. Hence these factors add up to the performance measure . Since the Robot's arm acts only on the Rubik's cube, it is the environment. To solve the Rubik's cube, the actions depends on the movement of fingers, wrist and joints and the acceleration of the arm. Hence they form the actuators. Based on the solving algorithm, the sensors used will be a camera, motion sensor to continuously monitor the Rubik's cube's movement . Along with this an air pressure sensor is also used to detect if there's any external air pressure given to disrupt the flow of action.

- Specify the properties of this environment. Justify your decisions **[4 pts]**:

Property	**Your choice**	**Explanation**
Fully observable?	Yes	The environment is fully observable because the Robot is able to see the complete state of the Rubik's cube at all points of time.

Multiagent?	No	The environment does not involve more than one agent as the only agent is the Robot's arm.
Deterministic?	Yes	The robot knows the next state through the actions done in the current state.
Episodic?	No	As there is dependency between the current and the next state, the environment is not episodic.
Dynamic?	Yes	The environment is changing when the Robot is taking time to decide on the action. The environment can change when there are external factors like air pressure, other mechanical actions which will disturb the robot-arm's action.
Discrete?	Yes	The Robot is able to solve the Rubik's cube by using finite number of actions.
Known to Agent?	Yes	The Robot knows the output for all the probable actions. Hence the environment is known to the agent.

Problem 2:

Consider the **undirected** graph presented below (fig. 2). Each node represents a single state (you can assume that each state represents a city on a map). If two states are neighbors, there is an edge between them.

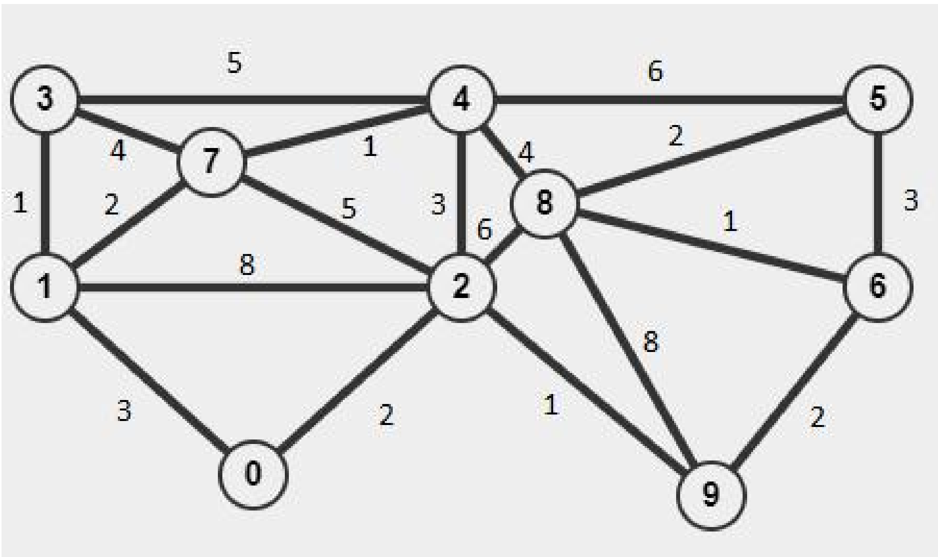


Figure 2: An undirected graph.

Assume that edge weights represent **driving distances between <u>cities/states in miles</u>**.

Your task is to utilize the **Hill Climbing algorithm to find a shortest (minimum cost) path** between two states provided data. Here are the steps:

- assume that:

■ **repeated states are <u>NOT</u> allowed**,
■ If you are at some state X and the next BEST action leads you to some already visited state, you stop your trip at state X.

■ select two states / cities (initial and goal states) at random under the condition that there is at least two (2) states between your initial and state goals (that would correspond to **<u>at least</u>** three (3) actions),

■ apply the Hill Climbing algorithm and show all steps / actions in Table A below,

■ provide a search tree diagram illustrating the path chosen along with evaluation function values and all alternatives (you can paste in a scan or a photo of a hand-drawn diagram or use some software to create it).

NOTE: The algorithm may get "stuck" and not reach the goal state.

My initial state is 3 and my goal state is 6.

TABLE A: Algorithm steps / actions [6 pts]				
Current state	Available actions and their costs	Selected action	Resulting state	Explanation / comments
3 (Initial state)	Action-1; Cost-1 Action-7; Cost-4 Action-4; Cost-5	Moving from state 3 to state 1.	State 1	The initial state is 3. There are 3 available actions 1,7 and 4 from this state but the state 1 has the minimum cost(cost=1). Hence as per Hill Climbing Algorithm, the state with minimum cost is visited. Hence state 1 is visited now.
1	Action-3; Cost-1 Action-7; Cost-2 Action-2; Cost-8 Action-0; Cost-3	Moving from state 1 to state 7.	State 7	The state 1 is now visited and there are 4 available actions 3,7,2,0 from state1. State 3 is already a visited state which cannot be revisited even though the cost is minimum for state 3 from state 1. Hence we are considering the next possible minimum state from state 1 which is state 7 (cost=2). Hence state 7 is visited now.
7	Action-1; Cost-2 Action-3; Cost-4 Action-4; Cost-1 Action-2; Cost-5	Moving from state 7 to state 4.	State 4	The state 7 is now visited and there are 4 available actions 1,3,4,2 from state7. The states 1 and 3 cannot be revisited as they are already visited states. The next possible state with minimum cost is state 4 (cost=1). Hence state 4 is visited now.
4	Action-7; Cost-1 Action-2; Cost-3 Action-8; Cost-4 Action-5; Cost-6	Moving from state 4 to state 2.	State 2	The state 4 is now visited and there are 5 available actions 7,2,8,5,3 from state4. The states 7 and 3 cannot be

	Action-3; Cost-5			revisited as they are already visited states. The next possible state with minimum cost is state 2 (cost=3). Hence state 2 is visited now.
2	Action-4; Cost-3 Action-8; Cost-6 Action-9; Cost-1 Action-7; Cost-5 Action-1; Cost-8 Action-0; Cost-2	Moving from state 2 to state 9.	State 9	The state 2 is now visited and there are 6 available actions 4,8,9,7,1,0 from state2. The states 1,7 and 4 cannot be revisited as they are already visited states. The next possible state with minimum cost is state 9 (cost=1). Hence state 9 is visited now.
9	Action-2; Cost-1 Action-8; Cost-8 Action-6; Cost-2	Moving from state 9 to state 6.	State 6 (Goal state)	The state 9 is now visited and there are 3 available actions 2,8,6 from state2. The state 2 cannot be revisited as it is a already visited state. The next possible state with minimum cost is state 6 (cost=2). Hence state 6 is visited now. Since state 6 is my goal state, the hill climbing algorithm is not applied on state 6.

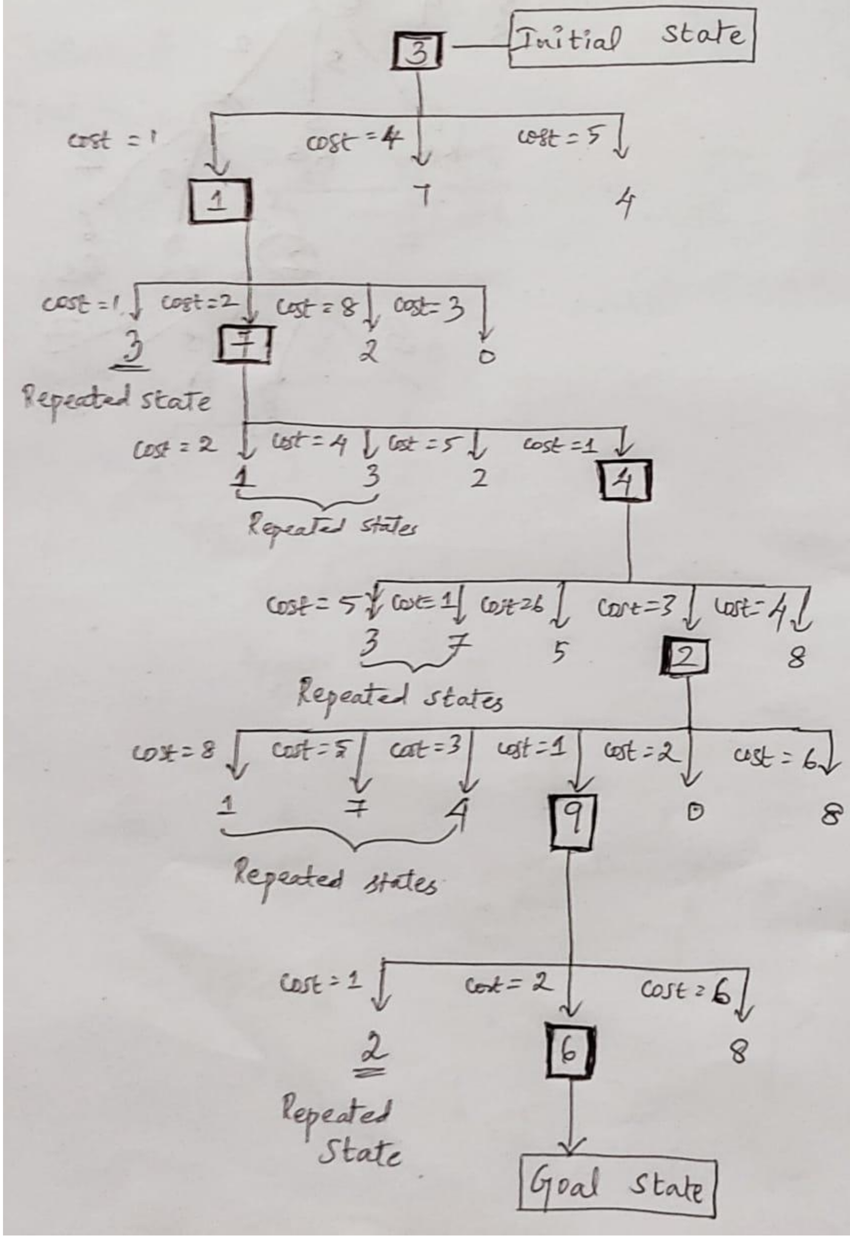
Tree search diagram [6 pts]

Question - 2

Hill Climbing Algorithm

Initial State - 3

Goal State - 6



Did the Hill Climbing algorithm pick the best (lowest total cost in miles) path?

Your answer:

According to me, the Hill Climbing algorithm did not pick the best path.
I have chosen the initial state as 3 and the goal state as 6.
As per the Hill climbing algorithm, adding the costs of my path gives 10.
(1+2+1+3+1+2=10).
Path: 3→1→7→4→2→9→6

But the shortest path is 3→1→0→2→9→6. Adding the costs for this path gives 9 (1+3+2+1+2=9).

Hence the shortest path gives me the lowest total cost paths in miles compared to the Hill climbing algorithm.