Department of Information Technology
University of the Punjab

IT-306: Artificial Intelligence
by Dr. Mian Muhammad Mubasher

Assignment 01 – Problem Solving by Searching (Transition Table)

Total Marks: 100
Average expected time for completion: 6 hours
Submission Date: **October 31, 2021, 11:59 PM**

Purpose of this programming assignment:

1. Introduce you with programming based on Artificial Intelligence concepts
2. Revise your Data structures concepts
3. Practice of problem solving

Instructions

1. Do not consult your ideas and code with your peers, in case of any problem consult with your instructor.
2. Expected time for this task is 6 hours.
3. No Evaluation for late submissions.

Problem Statement

A software artifact is required which can *solve problems by searching*. The software MUST take input from a file which has been described below. Reading file is mandatory.

Input file comprises of header, state descriptions, rule descriptions and transition matrix. First line of the file is header; it is a space separated triplet of integers (M N T). M represents number of states, N represents number of actions and T represents number of test cases. Header is followed by an empty line. Description of all possible states follows header (after empty line). Each line comprises of one state description. State descriptions follow an empty line. State descriptions are followed by action descriptions (after empty line). An M x N transition matrix of integers follows action descriptions (after empty line) which describe transition of each state after applying each action. Transition matrix is followed by an empty line which is followed by T number of test cases. Each test case is represented in a line. Each line is a pair of strings separated by tab, first string is a state representing initial state and second string is also a state representing final state.

The program must print results on standard output. There must be T number of lines in output, each line must represent output of corresponding test case. An output must be an arrow (->) separated list of actions.

Sample Input File

8 3 2 M represents number of states, N represents number of actions and T represents number of test cases

0 (AgentInRoom1, Room1Dusty, Roam2Dusty) s2
1 (AgentInRoom1, Room1Dusty, Roam2Clean)
2 (AgentInRoom1, Room1Clean, Roam2Dusty) s1
3 (AgentInRoom1, Room1Clean, Roam2Clean) g1
4 (AgentInRoom2, Room1Dusty, Roam2Dusty)
5 (AgentInRoom2, Room1Dusty, Roam2Clean)
6 (AgentInRoom2, Room1Clean, Roam2Dusty)
7 (AgentInRoom2, Room1Clean, Roam2Clean) g2

0 00 Clean A0
1 01 MoveToRoom1 A1
2 10 MoveToRoom2 A2

A(0 1 2)
0 2 0 4
1 3 1 5
2 2 2 6
3 3 3 7
4 5 0 4
5 5 1 5
6 7 2 6
7 7 3 7

(AgentInRoom1, Room1Clean, Roam2Dusty) s1 (AgentInRoom1, Room1Clean, Roam2Clean) g1
(AgentInRoom1, Room1Dusty, Roam2Dusty) s2 (AgentInRoom2, Room1Clean, Roam2Clean) g2

Sample Output

MoveToRoom2->Clean->MoveToRoom1
Clean->MoveToRoom2->Clean

☺ **Don't stop when you are tired, stop when you are done** ☺