LP1 Assignment AI&R 1

Date: 10th October, 2020

Title: 8-Puzzle Problem using A* Algorithm

Problem Definition:

Solve 8-puzzle problem using A* algorithm. Assume any initial configuration and define goal configuration clearly.

Learning Objectives:

- Learn Informed Search Algorithms
- Learn A* Algorithm and its application
- Learn about 8 puzzle problem
- To define Perception, Cognition, Action and Goal clearly

Learning Outcomes:

I will be able to design A* algorithm to solve the 8 puzzle problem.

S/W & H/W Packages:

1. Operating System: 64-bit Open source Linux or its derivative
2. Programming Language: Python3
3. Google Colaboratory(uses Tesla K80 GPU)

Related Mathematics:

Programmer's Perspective

Let S be the system set:

S = {s; e; X; Y; Fme;DD;NDD; Fc; Sc}

s=start state

e=end state

X=input state

X = {X1}

where,

X1 = start state

Y = goal state

Fme is the set of main functions

Fme = {f1,f2,f3}

where,

f1 = initializer

f2 = function to calculate distance

f3 = function to create children nodes

f4 = function to solve puzzle using A* Algorithm

DD= Deterministic Data

NDD=Non-deterministic data: No non deterministic data

Fc=failure case: If start and goal state do not match

Concepts Related to Theory:

8-puzzle Problem

- 8 puzzle is a popular puzzle that consists of 8 tiles and one empty space.
- The puzzle is divided into 3 rows and 3 columns.
- The other 8 tiles have numbers 1 through 8 on it

● The puzzle can be solved by moving the tiles one by one in the single empty space and thus achieving the Goal State.

Perception

● We visualize swapping the empty space with its neighbors

● The empty tile can have 9 possible locations.

● $\exists x.(move(x) \wedge valid\ location(x))$

● The empty space can only move in four directions (Movement of empty space) $\forall x.(shuffle(x) \wedge valid(x) \rightarrow open\ list(x))$

- ○ Up
- ○ Down
- ○ Right
- ○ Left

● The empty space cannot move diagonally and can take only one step at a time.

● $\exists x.(move(x) \rightarrow \sim diagonal(x))$

Cognition

$f(n) = g(n) + h(n)$

Cost of A* Algorithm: $f(n) = g(n) + h(n)$

● g(n) is the cost of the path traversed from the initial state to node n.

● h(n) is the estimated path-cost or the heuristic function cost from node n to the goal node. g(n) = Path Cost

● Each step costs 1, so the path cost is the number of steps in the path.

For any 8-puzzle's tile

- ● $Misplaced(i) = \{ ((x1_i=x2_i)\ \&\ (y1_i=y2_i))\ \forall\ x1_i,y1_i,x2_i,y2_i \in \{0,1,2\} \}$
- ● Where,
- ● $(x1_i, y1_i)$ is the coordinate of the tile with number i in current state
- ● $(x2_i, y2_i)$ is the coordinate of the tile with number i in goal state
- ● $x1_i,y1_i,x2_i,y2_i \in \{0,1,2\}$
- ● $i \in \{0,1,2,3,4,5,6,7,8\}$

- h(n) = Misplaced(i) for 0<=i<=8

Action

Action = {U, L, D, R}

- Action can be represented as a set of 4
- Each Action represents the interchanging of blank tile with the neighbor in one of the 4 directions.
    - Up (U)
    - Down (D)
    - Right (R)
    - Left (L)
- Constraints
    - The neighbor should not be diagonally adjacent
    - The edge positions can interchange in only three directions
    - The corner positions can interchange in only two directions
    - Interchanging takes place one step at a time fmin = min(f1,f2,...,fm)
- Choose the heuristic function with the minimum f-value ○ Where m is the number of states

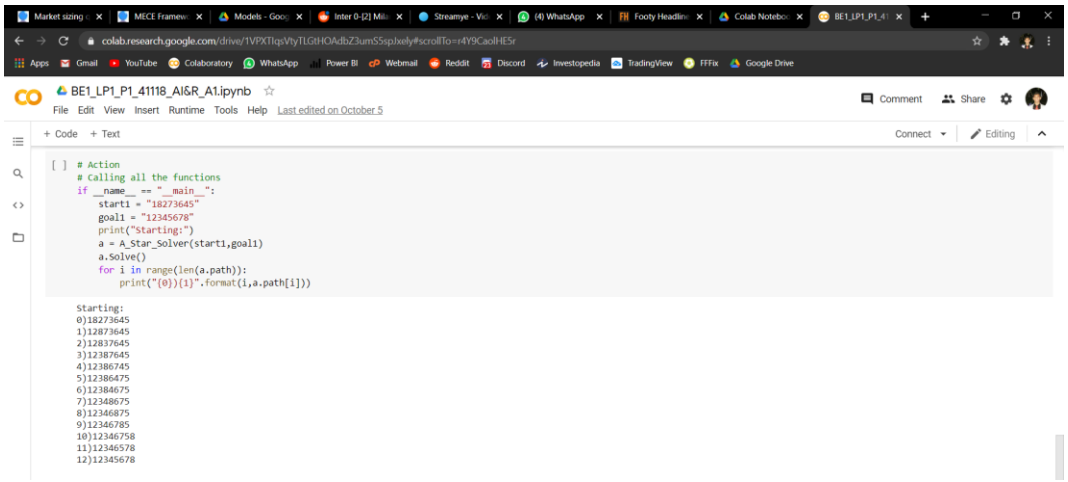
Steps for Execution:

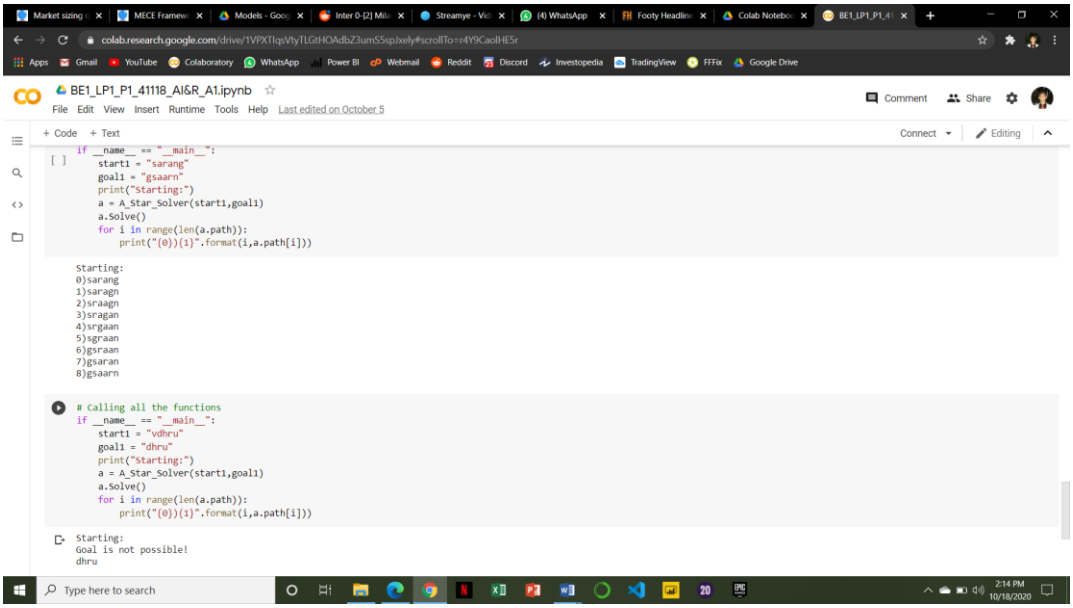
1. Open Google Colaboratory
2. Write code for A* Algorithm
3. Define start and goal states
4. Run the code

Notebook Link:

https://colab.research.google.com/drive/1VPXTIqsVtyTLGtHOAdbZ3umS5spJxely?usp=sharing

Output:





Conclusion:

I have successfully designed A* search algorithm for 8 puzzle and defined Perception, Cognition, Action and Goal for the same.