DS251

Assignment 1

Due Date - 15th February EOD

1. N-Puzzle or sliding puzzle consists of N tiles where N can be 8, 15, 24, and so on. The puzzle is divided into sqrt(N+1) rows and sqrt(N+1) columns. Eg. 15-Puzzle will have 4 rows and 4 columns and an 8-Puzzle will have 3 rows and 3 columns. The puzzle consists of N tiles and one empty space where the tiles can be moved. Possible operations include Up, Down, Left, and Right. Start and Goal configurations (also called state) of the 8-puzzle are provided. The puzzle can be solved by moving the tiles one by one in a single empty space and thus achieving the Goal configuration. Solve the given problem using the following algorithms: (a) Breadth First Search Algorithm (b) Depth First Search Algorithm and (c) A* Algorithm with a suitable heuristic.

7	2	4
5		6
8	3	1

Start State

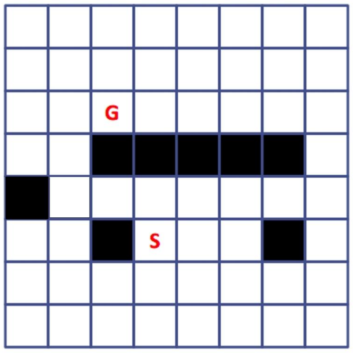
	1	2
3	4	5
6	7	8

Goal State

Show (using pen/paper) how much memory and how many steps are required to run each algorithm.

Write a code in Python to implement this. Show visualization of tiles in the terminal itself. The code should run for higher values of N also.

2. Consider a squared room whose floor space is divided into equal-sized grids. Suppose there is a mobile robot (MR) in one grid, and it wants to move to another grid that is at some distance from the starting grid of the robot. The condition is that some of the grids are occupied with obstacles(black boxes). Plan a trajectory path for the robot so that the robot reaches the goal position (G) from the initial position (S) without passing the black boxes. It should try to minimize the number of steps. The legal steps to be followed are in the order: UP, LEFT, RIGHT, DOWN.



Write a code in Python to implement the A* search with suitable heuristic function.

3. Consider the game of tic-tac-toe where X is the MAX player. Given the game board below where it is X's turn to play next, show the game tree.

```
__| X |__
0 |__ |__
X|__ | 0
```

The evaluation function on all the leaf nodes is given by:
Evaluation(s) = $8X_3(s) + 3X_2(s)+ X_1(s) – (8O_3(s) + 3O_2(s) + O_1(s))$ Where $X_n(s)$ is the number of rows, columns, or diagonals in state s with exactly n number of X's and no O's and $O_n(s)$ is the number of rows, columns, or diagonals in states with exactly n number of O's and no X's.

Use the minimax algorithm and alpha-beta pruning algorithm to calculate the optimal move. (pen/paper)

4. A Moon rover has to leave the lander, for collecting rock samples from three places Rock1, Rock2 and Rock3 (in any order) and return to the lander. Assume that it has a navigation module that can take it directly from any place of interest to any other place of interest. So, it has primitive actions go-to-lander, go-to-rock-1, go-to-rock-2, and go-to-rock-3.

We know the time it takes to traverse between each pair of special locations.
Lander - Rock 1 = 3 hrs
Lander - Rock 2 = 4.5 hrs
Lander - Rock 3 = 5 hrs
Rock 1 - Rock 2 = 2 hrs
Rock 1 - Rock 3 = 1 hr
Rock 2 - Rock 3 = 1.5 hrs

Our goal is to find a sequence of actions that will perform this task in minimum time. The constraint is the rover should visit each rock location only once.

a. Formulate this problem as a search problem by specifying the state space, initial state, path-cost function, heuristic function and goal test. Try to be sure that the state space is detailed enough to support solving the problem, but not redundant.
b. Solve this problem with an appropriate heuristic search algorithm. (pen/paper)

5. Let's say, you are going to spend a month in the wilderness. Only thing you are carrying is the backpack which can hold a maximum weight of 40 kg. Now you have different survival items, each having its own "Survival Points" (which are given for each item in the table). Some of the items are so essential that if you do not take them, you incur some additional penalty.

Here is the table giving details about each item.

Item	Weight	Survival Value	Penalty if not taken
Sleeping Bag	30	20	0

Rope	10	10	0
Bottle	5	20	0
Torch+Battery	15	25	-20
Glucose	5	30	0
Pocket Knife	10	15	-10
Umbrella	20	10	0

Formulate this as a genetic algorithm problem where your objective is to maximise the survival points. Write how you would represent the chromosomes, fitness function, crossover, mutation etc. And then write a python program to find the solution of this problem using GA. The program should have the capability to handle a different number of items with different values than what is given in the table.