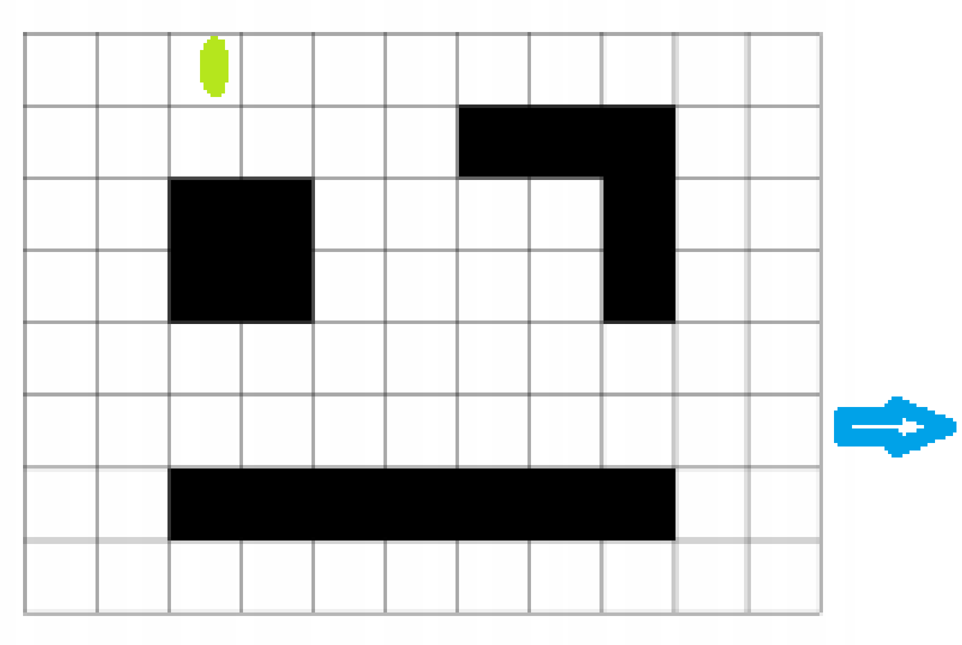
Assignment – Grid Search Using A* Algorithm

Problem statement: You are stranded somewhere on an 11x8 grid. The green dot in the below figure represents your present location on the grid. You are low on food and water and you are running out of energy to survive. Going from one cell to the next consumes one unit of your precious energy. Needless to say, you have an urgent need to exit this grid in as few steps as possible. There is only one cell through which you can exit the grid and that cell is represented by the blue arrow. The grid also has some barriers which you cannot cross but can go around. You can only move right, left, up and down to a cell which is within the grid and which is not a barrier.

The environment is static, fully observable and deterministic.

Although odds are stacked against you, all hope is not lost. You are an intelligent agent and have good understanding of various search algorithms using which you can exit the grid



1) In order to use one of the informed search strategies, which heuristic function h(n) would you use? Is your heuristic both admissible and consistent? Please explain.
2) How will you go about defining the total cost function f(n)
3) Implement Iterative deepening A*(IDA*) search $^{++}$ in python to find the path to the exit cell avoiding the barriers. Your output should contain the path taken by the agent (e.g. [(3,8),(4,8),(4,7)………] and also the total cost incurred in taking that path, total number of nodes expanded to get this state, maximum number of nodes kept at the memory at any point in time

4) Implement Greedy Best First Search in python to find the path to the exit cell avoiding the barriers. Your output should contain the path taken by the agent (e.g. [(3,8),(4,8),(4,7)………] and also the total cost incurred in taking that path, total number of nodes expanded to get this state, maximum number of nodes kept at the memory at any point in time
5) Compare the following for both IDA* and Best First Search

    a. Path taken to exit the grid
    b. Cost of this path. Is the path optimal for both?
    c. Total Number of nodes expanded to get this state [time complexity]
    d. Maximum number of nodes kept at the memory at any point in time [space complexity]

We will look to evaluate the following aspects of your submission

1. Your representation of environment
2. Representation of fringe and the methods associated with fringes like empty, pop (use this for a function to remove a node from fringe, even if the fringe is not a stack) and insert node
3. Implementation of IDA* and Best First Search with the necessary checks
4. Reporting of path taken, cost of path, time and space complexity
5. Quality of documentation

You will be provided with the python notebook template which stipulate the structure of code and documentation. You are free to add as many code cells as possible.

You can use any python library for your convenience. However, the search code involving IDA* and Best First Search and the fringe must be completely original.

Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.

[++] Iterative deepening A* search is an algorithm that borrows from both A* and iterative deepening DFS algorithms. Iterative-deepening-A* works as follows: at each iteration, it performs a depth-first search, cutting off a branch when its total cost f(n) = g(n) + h(n), exceeds a given threshold. This threshold starts at the estimate of the cost at the initial state, and increases for each iteration of the algorithm. At each iteration, the threshold used for the next iteration is the minimum cost of all values that exceeded the current threshold.