Intro to Artificial Intelligence
Assignment 1: Search algorithms

Summer, 2021

General Instructions

Teams: Assignment should be completed by teams of two students. No additional credit for working individually. Use this form https://forms.gle/nFWwmcncohXTVbDE7 to inform the TAs you either a) have a team (only one member of the team) b) looking for a team or c) working alone (discourage).

Submission: Submit a PDF file to Sakai. For the programming questions, submit your code in a compressed file. DO NOT submit documents in *Word*, raw text, images, etc. One submission per team is enough. Be sure to submit before the deadline (see Sakai). You have unlimited submission, partial submissions are encourage! Code submitted must run on *ilab* machines in order the be graded.

Program Demonstrations: Each team will have 10 minutes to demonstrate the implementation to one TA as well as answer questions on a date scheduled after the deadline. Be sure to prepare ahead to show the program running.

LaTeX: Extra credit (10%) for submitting answers using *LaTeX*. If you choose to do this, submit all files (*.tex) as a separate compressed file.

Plagiarism: Each team must implement and answer questions independently. Indicate any external sources used for your submission. If plagiarism is detected, the assignment will receive 0 points.

Description

Consider a grid world like the one shown in Fig. 1, where an agent can move freely in the non-blocked cells. The objective of this assignment is to implement and compare search algorithms that will allow the agent to move from a given initial state to given goal state without colliding.

Properties of the agent:

- Is deterministic
- Moving left or right has a cost of 1
- Moving up or down has a cost of 2

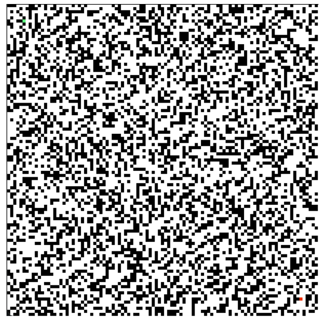

Figure 1: An example grid world

- Cannot move diagonally

Your program must:

- Read the *problem.txt* file
- Read the specified grid world
- Solve the problem using the specified algorithm
- Show the resulting path and its cost

The *problem.txt* file

By lines (see example on Sakai):

0. Size of the maze N - integer
1. Start state xy - two integers: $0 \leq x, y < N$
2. Goal state xy - two integers: $0 \leq x, y < N$
3. Algorithm - integer from 0 to 4
4. Maze - integer representing the maze to use

Algorithms:

0. Select one of: Breadth-first search or Uniform-cost search
1. Select one of: Depth-limited, Iterative deepening depth-first search or bidirectional search
2. A* using one of h_0, h_1 or h_2
3. A* using one of h_3, h_4 or h_5
4. A* using your own heuristic

Heuristics:

- h_0 - Euclidean distance
- h_1 - Manhattan distance
- h_2 - Infinity norm (Max of x,y)
- $h_3(n) = min\{h_i(n), h_j(n)\}, h_i \neq h_j; i, j \in [0, 2]$
- $h_4(n) = w * h_i(n)(1 - w) * h_j(n), w = rand(0, 1); h_i \neq h_j; i, j \in [0, 2]$
- $h_5(n) = max\{h_i(n), h_j(n)\}, h_i \neq h_j; i, j \in [0, 2]$

1 Questions

Setup [5 points]: *(Code)* Be able to read the files and show the grid world with the initial state of the agent.
A well defined problem [5 points]: *(pdf)* Describe the problem to be solved. What kind of task environment is this?
Algorithm 0 [10 points]: *(Code)* Implement and run the selected algorithm on 50 environments.
Algorithm 1 [10 points]: *(Code)* Implement and run the selected algorithm on 50 environments.
Algorithm 2 [10 points]: *(Code)* Implement and run the selected algorithm on 50 environments.
Algorithm 3 [10 points]: *(Code)* Implement and run the selected algorithm on 50 environments.
Algorithm 4 [10 points]: *(Code)* Implement and run the selected algorithm on 50 environments.
Analysis [20]: *(pdf)* Answer (and explain why / why not):

- Is the implemented Algorithm 0 complete? optimal? What about the other one?
- Is the implemented Algorithm 1 complete? optimal?
- For algorithm 2, is the selected heuristic consistent? admissible?
- For algorithm 3, is the selected heuristic consistent? admissible?
- For algorithm 4, is the selected heuristic consistent? admissible? Explain how the choice was made and if it is better than the other ones.

Results [20 points]: *(pdf)* Show, compare (plot) and explain your results.