Homework 2 - Search

Due September 14, 2023

Harry Potter (HP) and Albus Dumbledore (AD) are searching for horcruxes. There are (initially) two horcruxes (HX1, HX2).

They live in a world where there are four locations:

- Hogwarts (hw)
- Hogsmeade (hm)
- London (ln)
- Gringotts (gs)

There are bidirectional magical passages between these locations: hw - hm, hm - ln, hm - gs, ln - gs.

The following actions are possible at each timestep:

- Harry moves through a magical passage (cost 1.0)
- Dumbledore moves through a magical passage (cost 2.0)
- Harry destroys a horcrux in his current location (cost 3.0)
- Dumbledore destroys a horcrux in his current location (cost 1.0)

At each timestep exactly one action is taken.

The initial state is as follows: Harry is in London, Dumbledore is at Hogwarts, the first horcrux is hidden in Hogsmeade and the second horcrux is in a locker in Gringotts.

The goal is to reach one of the states where there are no horcruxes.

P1: Model the state representation (1 pt)

Develop a representation of the problem.

Describe how do you label the states. List at least five states as representative examples.

Describe how you label the possible actions and transitions. List at least five actions and their associated transitions.

P2: State counts (1 pt)

How many states are in your representation?

How many goal states are in your representation?

How many edges are in the search state graph?

If you cannot provide an exact answer, please provide an estimate, or an upper value. Explain how you reached the estimate.

P3: Search tree (1 pt)

Estimate the average branching factor b of the search tree. Explain how you reached this estimate.

Estimate the maximum depth of the search tree. Explain.

P4: Depth first search (1 pt)

Trace the **depth first search algorithm** for solving the problem. If you don't reach a goal state, stop after 10 steps.

For each step, list the node that had been chosen for expansion, the newly expanded nodes, and the new fringe.

P5: Breadth first search (1 pt)

Trace the **breadth first search algorithm** for solving the problem. If you don't reach a goal state, stop after 10 steps.

For each step, list the node that had been chosen for expansion, the newly expanded nodes, and the new fringe.

P6: Iterative deepening (1 pt)

Trace the **iterative deepening algorithm** for solving the problem. If you don't reach a goal state, stop after 10 steps.

For each step, list the node that had been chosen for expansion, the newly expanded nodes, and the new fringe.

P7: Uniform cost search (1 pt)

Trace the **uniform cost search algorithm** for solving the problem. If you don't reach a goal state, stop after 10 steps.

For each step, list the node that had been chosen for expansion, the newly expanded nodes, and the new fringe.

P8: Python programming (1 pt)

Implement the solution for problems P4, P5, P6 and P7 in Python.

P9: Heuristics and A* (1 pt)

Propose an admissible heuristic for the problem. Explain why it is admissible.

Trace the **A* algorithm** for solving the problem. If you don't reach a goal state, stop after 10 steps.

For each step, list the node that had been chosen for expansion, the newly expanded nodes, and the new fringe.

P10: Implement A* (1 pt)

Implement your solution for problem P9 in python.

What to submit?

- Text document with solutions to the problems. For the programming problems, the text document should contain output listings (of reasonable length).
- A single python file, that provides the solutions to problem. Running the python file should print out the solutions.