Artificial Intelligence

COMS 4701 – Summer 2016

Home Work n°1: Search Agents

Due Monday June 6thth, 2016 @11:59pm

Problem 1: Search Algorithms

In this assignment you will create an agent to solve the 8-puzzle for an n by n array (the n-puzzle). You may visit mypuzzle.org/sliding for a refresher of the rules of the game. Begin by writing a class to represent the state of the game at a given turn including parent and child nodes. Implement and compare several search algorithms to operate on your problem class and evaluate their performances on several instances. Collect statistics on the cost of path, nodes expanded, depth of the stack/queue, maximum stack/queue size and running time. Discuss the pros and cons of your knowledge representation. Make sure that your algorithms generate and print the correct solution for an arbitrary problem instance and given array size. Standard academic honesty policy applies (solutions isomorphic to those that exist online will be detected and considered plagarism).

Assumptions

1. The grid is an $n \times n$ array whose components are either numbers from the set $\{1, \cdots, n-1\}$ and one blank
2. The blank may be swapped with a component in one of the four directions (UP, DOWN, LEFT and RIGHT), one move at a time.
3. The cost of moving from one configuration of the board (one node) to another is the same and equal to one. Thus, the total cost of path is equal to the number of moves made from start to the goal position.

Example

n = 3, Cost of path = 3, nodes expanded = 5, depth of stack/queue = 4, max stack/queue size = 5, running time = 13 milliseconds, path to solution: UP, LEFT, LEFT

$$parent = \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 4 & \\ \hline 6 & 7 & 8 \\ \hline \end{array} \qquad \Longrightarrow \qquad child = \begin{array}{|c|c|c|}\hline 1 & 2 & \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array}$$

$$parent = \begin{array}{|c|c|c|}\hline 1 & 2 & \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array} \qquad \Longrightarrow \qquad child = \begin{array}{|c|c|c|}\hline 1 & & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array}$$

$$parent = \begin{array}{|c|c|c|}\hline 1 & & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array} \qquad \Longrightarrow \qquad child = \begin{array}{|c|c|c|}\hline & 1 & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & 8 \\ \hline \end{array}$$

Questions

1. (25 points) Given an instance of the problem, implement a **Breadth-First Search** to obtain a path from the start to the goal position.
2. (25 points) Given an instance of the problem, implement a **Depth-First Search** strategy to obtain a path from the start to the goal position.
3. (25 points) Given an instance of the problem, implement an **A* Search** to obtain a path from start to the goal position. State the choice of heuristic you implement explicitly.
4. (15 point) Summarize and compare the results of the above three approaches w.r.t. to:
   (a) cost of the path (exact numbers),
   (b) number of nodes expanded
   (c) maximum depth of the stack/queue
   (d) memory requirements of each approach (exact numbers/general view), and
   (e) running time (exact numbers).
5. (5 points) Justify your choice of a heuristic.
6. (5 point) Discuss your knowledge representation.
7. (10 points) **(Optional)** Implement an Iterative Deepening A* search (IDA*) and compare it to the other methods implemented above.

Files to submit

```
hw1_UNI.py           - Completed Python code

hw1_report_UNI.pdf   - Completed PDF Report. Use of LaTeX is optional but preferred.
```

The file hw1_output1.txt is expected to contain the following: solution to each algorithm with the cost of the path, in the following order BFS, DFS and A* search (e.g., see hw1_output_example.txt)