

CPSC 352 Artificial Intelligence

Programming Project 1

February 22, 2021

Due Friday, March 12, 2021 (11:59 pm)

In this project, you will implement two search algorithms for the 8-puzzle and analyze their relative performance.

Recall that this problem is defined as follows:

States: A state description includes the location of each of the 8 tiles and the blank one.

Initial State: Any state can be the initial state. The states are partitioned into two sets, such that half of them can achieve one solution, and half of them the other.

Actions: These track the movement of the blank space: *Left, Right, Up, Down*. (These are not all possible in every state.)

Transition Model: Given a state and an action, this returns the updated state.

Goal test: Does the current state match the goal?

Path cost: Each step costs 1, so that path cost is just the number of moves.

Search Algorithms: For the first search, use BFS. For the second search, implement A* Search using heuristic $h1$. (For the final 5% of the credit, also implement $h2$ and compare.)

We will use the following two goal states:

1	2	3
4	5	6
7	8	

1	2	3
8		4
7	6	5

Parity 0

Parity 1

Data: You should allow for either the input of a specific initial configuration (for test purposes), or randomly generate a configuration. Input format should be a list of digits separated by spaces in row-major order, such as

1 2 3 8 0 4 7 6 5

Where "0" represents the blank tile. With either input, your program should determine the parity and therefore which goal state to use.

Output: One run of the program should execute each search strategy on the same input. Show the initial state, the first five moves taken by each strategy, the total number of moves required, and the total number of search tree nodes explored.

Analysis: Run your program on at least ten random configurations and compile the results. Write up a discussion of the results. You should compare the results of the two (three) algorithms and consider the relationship between the length of the solution found and the number of search tree nodes explored.

Grading Rubric:

Design and Clarity:	15%
Basic Functionality:	55%
Full correctness:	15%
Analysis:	10%
Heuristic h_2 :	5%