CIS 479 Fall 2025 Programming Assignment 1 [Search]

Due 10/22/2025 11:59 pm, Total:[100 marks]

Please submit everything through Canvas and cite appropriate references wherever relevant including ChatGPT.

1 Problems

a) In this programming assignment, you will solve the 8-Puzzle Problem. The 8-Puzzle problem consists of a (3*3) grid with one blank space and eight numbered tiles. The goal is to arrange the tiles in numerical order by sliding them into the empty space. The action space can be thought of as the blank tile moving UP, DOWN, LEFT and RIGHT. You can consider path cost to be +1 for every tile moved. A sample initial and goal configuration looks like this:

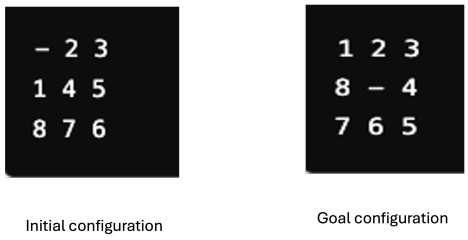

Figure 1: 8-Puzzle Problem (initial and goal configuration

For the above initial and goal configuration, implement the below search strategies:

1. BFS Search [10 marks]
2. DFS Search [10 marks]
3. Iterative Deepening Search [10 marks]
4. Uniform Cost Search [10 marks]

5. Greedy Best First Search with heuristic functions as below
   a)# of misplaced tiles [10 marks]
   b) Manhattan distance [10 marks]
   c) Any other heuristic you can think of? (This is optional, be creative!!)

6. A^* Search with heuristic functions as below
   a)# of misplaced tiles [10 marks]
   b) Manhattan distance [10 marks]
   c) Any other heuristic you can think of? (This is optional)
   d) One way to make A^* search efficient is **Beam search** where you either keep a limit 'k' on the best nodes of the frontier (the ones with higher values of f(n)) or only expand nodes that lie within a threshold. Can you implement Beam search for this problem and notice whether you ended up with an optimal/non-optimal solution. (This is optional)

Please have a single code (eightpuzzle.py) which triggers all the above-mentioned search functions. All the search functions can be in a different python code (however not a strict requirement).The command line input to test your code will be as below:

python eightpuzzle.py –search "BFS/DFS/IDS/UCS/GBS/A*/Beam" –initial '[[-,2,3],[1,4,5],[8,7,6]]' –goal '[[1,2,3],[8,-,4],[7,6,5]]' –heuristic "misplaced/manhattan/other".

Your code should output the following details for each search algorithm, please generate the search output in a different text file named **output_{search}_{heuristic}.txt** for each search algorithm. Below is the naming convention:

1. DFS: output_DFS.txt
2. BFS: output_BFS.txt
3. IDS: output_IDS.txt
4. UCS, misplaced heuristic : output_UCS.txt
5. GBS, misplaced heuristic : output_GBS_misplaced.txt
6. GBS, manhattan heuristic : output_GBS_manhattan.txt
7. GBS, other heuristic : output_GBS_other.txt (optional)
8. A*, misplaced heuristic: output_A*_misplaced.txt
9. A*, manhattan heuristic : output_A*_manhattan.txt
10. A*, other heuristic : output_A*_other.txt (optional)

Your search output in these files should contain the following details:

1. Path: This should output the entire search path (states and actions). You could either output the states as a board or as a 2D list as in command line. Please output the path as: $(s_1, a_1) - > (s_2, a_2) - > .....- > (s_g)$ where s_1 is the initial state and s_g is the goal state. You can print these sequences out as strings in the output file. Please enter a header (BFS/DFS/IDS/UCS/GBS,A^*) for every search algorithm followed by a "**********" before printing out the paths.

2. Path cost

3. Depth of the Search Tree

4. Time taken to run the specific algorithm

5. Number of nodes expanded by the search tree (memory usage)

b) At the end, create a table comparing these various outputs of your implementation for all the different search algorithms (with all the heuristics). This should be submitted as a separate pdf along with the code. [20 marks]

2 Deliverables

1. Code along with proper **read me** document. The folder should contain a eightpuzzle.py code which takes in the relevant command line parameters.

2. Output file: output.txt with the outputs of the various search algorithms (search output, path cost, depth, time and number of expanded nodes). For submission, you can either append all the output to a single output.txt file or keep different output files for different search algorithms with appropriate names.

3. A pdf **report file** containing a table with the algorithms as columns and path-cost, depth, time taken and number of expanded nodes as rows. If you have any interesting heuristic that you used for Informed Search (this is optional), please also describe your intuition behind the heuristics in detail.

3 Resources

1. Consider using libraries like heapq for priority queues or collections.deque for queues.

2. Review lecture materials and the book for search algorithms.

3. You can also ballpark your solutions using the software: https://tristanpenman.com/demos/n-puzzle/ to get an idea.

4. I am attaching a small filler template code which you might find helpful and use it while writing your code. The file "eightpuzzle.py" is the main file that will trigger the search. You have to modify this code as required by taking in command line arguments and also produce the required output for the assignment. Remember, this is not a complete runnable code. It calls "search.py" which consists of a general search problem template that you can fill in. It has predefined functions for BFS, DFS, etc where you can write your code. The code template is taken from The Pacman AI projects developed at UC Berkeley, http://ai.berkeley.edu