

Assignment: A* to solve the modified n-puzzle problem

Suppose the standard n-puzzle problem is modified so that there are two empty locations (instead of one) allowing you to pick any one tile adjacent to any one of the two empty locations and move it to the adjacent empty location. You are given a starting configuration and are supposed to rearrange the tiles by move them into the empty locations as necessary to end up in a given goal configuration. Figure 1 shows an example of a starting and a goal configuration.

1	4	-	7
9	2	3	5
6	-	10	13
8	11	14	12

Starting configuration

1	4	7	5
9	2	3	-
-	11	10	13
6	8	14	12

Goal configuration

Figure 1: Sample starting/goal configurations. Two '-'s are the empty locations.

Q1)

Implement the A* algorithm in any programming language of your choice to solve this modified n-puzzle problem. Your program should be able to accept as command line input, two tab delimited files of the form provided together with this assignment as sample starting configuration and goal configuration. The output should be a *text* file containing a sequence of moves of the form [(tile_number, move)]. E.g., see the sample output file provided. (1,up), (2,down), (3,left) (8, right).

- Experiment with the two heuristics “ number of misplaced tiles” and “total Manhattan distance” and compare their efficiency in terms of the number of moves to reach the goal. Is one better than the other? Your answer should be based on the average run-time difference between the two heuristics and its standard deviation over at least 100 randomly generated start/goal configuration pairs. Your test configurations should comprise of different puzzle sizes (n) ranging from 5 - 20. Explain your experimental procedure and provide the results.
- Submit your program as an executable file that can be run from the windows command line by providing the input as mentioned above in Q1. Submit the source code as well. Please submit everything in one zipped folder.