

## CSE 4082 – Project 1

(Due 03.01.2021 at 23:59, electronic submission only, to cse.cse482@gmail.com )

Diagonal 15-Puzzle is a modified version of the 15-Puzzle in which the blank square may slide about a diagonal axis in addition to horizontal and vertical axes. However, the cost of each diagonal slide is 3 whereas the cost of each horizontal or vertical slide is 1. The final state of the Diagonal 15-Puzzle is as follows:

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

In this project, you are required to implement the following search methods for solving the Diagonal 15-Puzzle:

- Uniform Cost Search
- Iterative Lengthening Search
- A\* Heuristic Search with the admissible heuristic  $h_1$  discussed in class ( $h_1(n)$  = The number of misplaced tiles at node  $n$ )
- A\* Heuristic Search with the admissible heuristic  $h_2$  discussed in class ( $h_2(n)$  = The sum of the city-block distances of each misplaced tile from its current location to its goal location)
- (Bonus - 10 points) A\* Heuristic Search with the admissible heuristic  $h_3$  which provides better solutions than  $h_2$ .

In order to compare performances of these methods, your program should be able to generate random puzzle instances such that the optimum solution is at the depth  $d$  of the tree. In order to generate random puzzle instances, you may start from the final state and randomly go backwards. Note that you should prevent cycles when generating random instances.

You have to provide the following outputs:

The total number of expanded nodes				
d: depth of the solution	UCS	ILS	A*-H1	A*-H2
2				
4				
6				
8				
10				
12				

16				
20				
24				
28				

The maximum number of nodes stored in memory				
d: depth of the solution	UCS	ILS	A*-H1	A*-H2
2				
4				
6				
8				
10				
12				
16				
20				
24				
28				

For each entry in the above tables, you have to generate 10 random instances and report the arithmetic average. You may not be able to fill every entry in the table since it may take too long. In this case leave the corresponding entries empty.

In addition to the tables above, for each of the three instances provided below and for each algorithm, you should submit:

- i. The cost of the solution found.
- ii. The solution path itself (from the given initial state to the final state).
- iii. The number of expanded nodes.

a.

	1	3	4
12	13	2	5
11	14	15	6
10	9	8	7

b.

1	3	5	4
2	13	14	15
11	12	9	6
	10	8	7

1 8 3

c.

1	13	3	4
12	11	2	5
9	8	15	7
10	6	14	

Notes:

- a. The project should be done in groups of two! If you want to work in groups of three, then you have to implement the following algorithms additionally:
  - i. Depth First Search
  - ii. Breadth First Search
  - iii. Iterative Deepening Searchand report the results.
- b. Using source code (even partially) found from internet or any other resources is not allowed!
- c. You should also submit a design document describing the classes (fields and methods) used in the project. This document should also contain the required tables and the other outputs.
- d. Your code will not be executed if you do not submit the required outputs! Thus, if you do not present your outputs (two tables and outputs for the given three instances) in the document, you will get the minimum grade!
- e. Do not submit separate output file, embed the outputs in the design document!
- f. Do not submit any executable file (your e-mail may return back!) Submit only source code and design document (with outputs).