Introduction to AI NCTU Spr2020 Programming Assignment #1 Due 4/10/2020

This programming exercise concerns the move of a "knight" in a 8x8 chessboard. A knight's moves are limited to (±1,±2) or (±2, ±1) from its current position. Here you need to write a program that finds the optimal (minimum number of steps) path between two given locations on the chessboard.

You should implement and experiment with BFS, DFS, IDS, A*, and IDA*. For A* and IDA*, first use the heuristic function given below. Use graph-search for BFS, DFS, and IDS. You can use a lookup table to implement the explored set.

Write your program as a function. The first input should be an integer representing the type of algorithm. The next four inputs represent locations: starting x, starting y, goal x, and goal y. All the locations range from 0 to 7. Use the return value to represent the number of expanded nodes.

Your program should print out a list of 0-based coordinates (row, column) representing the 'stops' of the knight on the solution path, including A and B. For example, for the path from (0,0) to (2,2), an optimal path is

(0,0)(2,1)(0,2)(1,0)(2,2)

You submission is a report file in PDF format. The report (maximum 5 pages single-spaced) should describe your experiments and results, especially the comparison between the different algorithms. In your report, also include a section describing your observations, interpretations, things you have learned, remaining questions, and ideas of future investigation. Include your program code as an appendix (not counting toward the 5-page limit), starting from a separate page. You can use C/C++, Java, Python, or MATLAB to write your program. In general, the TAs will not actually compile or run your programs. The code listing is used to understand your thoughts during your implementation, and to find problems if your results look strange. Therefore, the code listing should be well-organized and contain comments that help the readers understand your code; this will also affect your grade.

The submission is to be through E3. Late submission is accepted for up to a week, with a 5% deduction per day.

Try to think of some other experiments you can do. For example, what happens if you double the size of the board? How do the different searching strategies scale?

Heuristic function: A simple heuristic function to use is floor((|d_x|+|d_y|)/3), where (dx,dy) is the vector from the current location to the goal, and floor(v) is the largest integer equal to or less than v.

Extra credit: You can try to devise your own heuristic function that is better than this one, and compare the performance.