

### LISP Assignment 3

#### **Sudoku, Lisp and Searching**

Questions 1, 2, and 3 require you to write two search algorithms in Lisp to solve Sudoku puzzles and answer questions regarding the search results. We are providing all the required code, other than the search functions in the file sudoku.lisp. The code permits running the system with either a puzzle entered directly into the program file, or a puzzle loaded from a file. We have also provided five sample puzzles for purposes of testing your algorithms, which are (puzzle1.lisp, puzzle2.lisp, puzzle3.lisp, puzzle4.lisp, and puzzle5.lisp). Question 3 requires you to answer a series of questions based upon your solutions to Questions 1 and 2.

**Submit your program as Sudoku\_RollNo.lisp.**

**Last date for submitting the assignment is 23/11/2018 (Friday)**

Each of your search functions needs to include the following:

- A test to verify that the input board is indeed a legal board, if not the function should halt and output “Invalid board given”.
- A variable, that will be accessible in the general Solve functions, named numstates-expanded. You must also write the code to update this variable with the number of states expanded during the search.
- A variable, again accessible in the general Solve functions, named numstates-onstack. You must write the code to update this variable with the maximum number of states on the stack.
- Output the test “No Solution Found” if there is no solution to the puzzle.

(Hint: There is a solution, provided in the file, for all five puzzles. However, we may test your code for a puzzle that has no solution).

You should not to modify any of the code outside of the two search functions. Nor are you to modify the input variables to the functions.

- (20 points) Implement a depth first search. The skeleton of the function DFS-Solve is in the sudoku.lisp file. You are to write the function body.
- (20 Points) Implement a breadth first search. The skeleton of the function BFS-Solve is in the sudoku.lisp file. You are to write the function body.
- (10 Points) You will need to run the solver with each puzzle for both of the search methods. In order to obtain the correct information you need to run the solve-from-file function.
  - (4 Points) Provide a completed table with the following format.

| Search | Puzzle | #States Expanded | #States on Stack | Real Time (secs.). | Run Time (secs.) | Space (bytes) |
|--------|--------|------------------|------------------|--------------------|------------------|---------------|
| DFS    | 1      |                  |                  |                    |                  |               |
|        | 2      |                  |                  |                    |                  |               |
|        | 3      |                  |                  |                    |                  |               |
|        | 4      |                  |                  |                    |                  |               |
|        | 5      |                  |                  |                    |                  |               |
| BFS    | 1      |                  |                  |                    |                  |               |
|        | 2      |                  |                  |                    |                  |               |
|        | 3      |                  |                  |                    |                  |               |
|        | 4      |                  |                  |                    |                  |               |
|        | 5      |                  |                  |                    |                  |               |

b. (6 Points) Given your output in the table from part (a) of this question, explain the differences that you see between the two algorithms based upon the five sets of data that you collected. In addition to describing the differences you see in your data; please describe the underlying cause of the differences between the algorithms (e.g. provide an answer with additional depth, not simply the number of nodes expanded for DFS was X and for BFS was Y).