Name(s):_____

CSE 3521 Artificial Intelligence SP'21

**Homework Assignment #3** (26 points)

---

This assignment requires coding in Javascript. Use the template files provided in `HW3_template.zip` to get started.

1. Implement a successor function and goal test function for the 8-puzzle problem, as described in the slides. Refer to the provided template file `eight_puzzle_student.js` for more detailed instructions. (Also see the `two_jugs.js` and `vaccuum.js` files for examples of these functions for some other problems.) (4 pts)

2. Come up with several board configurations and use `eight_puzzle.htm` to test your functions from (1) against them. (Be sure to include the goal state!) Do the results from your functions match your expectations? Explain. (1 pt)

3. Implement the breadth-first search algorithm. Refer to the provided template file `bfs.js` for more detailed instructions. (6 pts)

   Your search functions **must** be generic (i.e., they don't depend on the problem you are solving). You should be able to use the provided example problems in `two_jugs.htm` and `vaccuum.htm` as additional tests for your code.

4. Implement the depth-limited search algorithm. Refer to the provided template file `dls.js` for more detailed instructions. (6 pts)

5. Implement the iterative-deepening search algorithm. Refer to the provided template file `ids.js` for more detailed instructions. (2 pts)

6. Implement the A* search algorithm. Refer to the provided template file `astar.js` for more detailed instructions. (4 pts)

(continued)

7. Come up with several board configurations and test your 4 search functions on them (you may re-use the boards from (2) ). Run your depth-limited search twice, first using as the depth limit the length of the path returned by either

your BFS or IDS. Second, use twice that value. Do the returned solutions (or lack thereof) match your expectations? Explain. (1 pt)

8. Choose a non-trivial board configuration and report the number of states evaluated and expanded for each search function. Run depth-limited search with two different depth values as in (7). Test A* search using both the Misplaced Tile Count and Manhattan Distance heuristics (both have been provided for you in `eight_puzzle_student.js`). Also, test A* using a "stupid" heuristic that returns only 0.

   Do these values match your expectations? Discuss.
   (2 pts)

9. (Optional) Include an estimate of the time you (total if working in a pair) spent working on this assignment. (This will used to help evaluate how to adjust assignments in future iterations of the course.)

**Create (and submit in class) a report** including answers to the asked questions and a printout of your code. Also, **create a ZIP archive of your code files** and submit it in the Homework 3 dropbox on Carmen.

Note: This assignment may be completed as a group of up to two people. Each group should submit a single report (with both names on it) and only one group member should submit the combined code to Carmen.

---

**Tips:**

If you need to print out debug statements, you may use the `console.log()` function to print out to the browser's debug console. To access this log, use Ctrl-Shift-J in Chrome or Shift-F5 (Console tab) in Firefox.
Safari Option-Cmd-C

Alternately, you can use the `helper_log_write()` function (from `search_helper.js`) to output to the log region on the web page.

An example search function has been provided in `rnds.js` which shows how the `is_goal_state()` and `find_successors()` functions should be called and how the solution path returned by search functions should be formatted.