1. (a) Write down "big-O" expressions for the time and space complexity of depth-first and breadth-first search.

   (b) Under what circumstances would a depth-first search be preferred to a breadth-first one, and vice-versa?

   (c) Using reasonable estimates for the time and space costs of expanding a node, explain whether the use of a breadth-first approach would be justified in searching a tree with an estimated branching factor $b = 10$, and a solution known to be at depth $d = 8$.

   (d) What does this suggest to you about the use of A^* search for wide and deep trees? Explain.

2. (a) A^* search uses an evaluation function $f(n)$ to order the nodes on its search agenda. $f(n) = g(n) + h(n)$ where:

   - $g(n)$ is the cost of getting from the initial state to state n along a particular path chosen by the search, and;
   - $h(n)$ is a *heuristic estimate* of the remaining cost to get from n to the goal.

   What search strategy (or strategies) do we get if we set

   i. $g(n) = 0$?

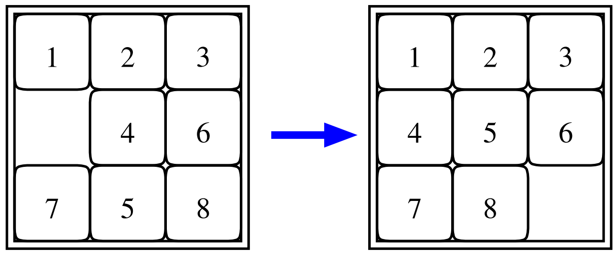
   ii. $h(n) = 0$?

   iii. both $g(n) = 0$ and $h(n) = 0$?

   (b) If $h(n)$ can be guaranteed to be equal to $h^*(n)$, the A^* process will converge to the goal immediately (i.e., as fast as possible). Prove this.

   (c) Suppose we have two candidates for $h(n)$, one never less than the true minimum distance to the goal and the other never greater. Which one would you choose for the search? Why?

3. In the popular "8-tiles" puzzle eight small square tiles (and an empty space) are arranged on a 3×3 grid. The aim is to organise the tiles in a certain way, but the only permitted moves involve sliding a tile into the empty space. A particular puzzle starts in state A, and the aim of the game is to get to state B in the smallest number of moves.



State A State B

(a) Suggest a suitable heuristic for this puzzle and justify your choice.

(b) Perform an A^* search on this problem, starting at state A and progressing to state B. Show all intermediate states generated, and the search agenda at each iteration of the search.

4. Consider this search graph. Why is it admissible? Why is it inconsistent? Show that A^* graph search fails for this problem (i.e., returns a sub-optimal solution), whereas A^* tree search succeeds.