8-Puzzle

State: x00 x01 x02 x10 x11 x12 x20 x21 x22 xij in {0, …, 8}, i, j in {0, 1, 2}
xij is the number in row I and column j.
xij = 0 iff the square is empty.

Start state: 724506831
Goal state: 012345678

Successor function: move the blank space left, right, up or down.

State	Successor states
724506831	724056831, 724560831, 704526831, 724536801
724560831	724506831, 720564831, 724561830

Cost of each step is 1.

Belongs to the family of sliding-block puzzles.
NP-complete

- 8-puzzle: 9! states
- 15-puzzle: 4 by 4 board. 1.3 trillion states. solves in a few milliseconds by best search algorithm.
- 24-puzzle: 5 by 5 board. 10^25 states. solves in several hours by best search algorithm.

River Crossing Problem

State: P C1 C2 B. P, C1, C2, B in {0, 1}. 0 denotes left side of the river and 1 denotes the right side of the river.

Start state: 0000

Goal states: 1110 and 1111

State	Successor states
0000	1001, 0101, 0011, 0111
1001	0000
0101	0000
0011	0000
0111	0010, 0100, 0000

Cost of each move is 1

Total number of states: 2^4 = 16

This problem is small enough that we can write out the entire search graph.

4-Queens Problem

Incremental formulations

Formulation A:

State: k queens on the board. 0 <= k <= 4.
x1 y1 x2 y2 … xk yk. Queen I is in row xi and column yi. xi, yi in {0, 1, 2, 3}

Start state: no queens on the board.

Goal state: 4 queens on the board, none are attacking each other.

Successor function: add a queen to any empty square on the board.

In the worst case, how many paths do we need to search through? In other words, how many leaf nodes are there in the search tree?
16 * 15 * 14 * 13 = 43680. (More than 10^4)

Formulation B:

State: one queen per column in the leftmost k columns with no pair attacking each other.
r0 r1 … rk-1. Queen I is in column I and row ri. 0 <= i <= k – 1.

Start state: same as formulation A.

Goal state: same as formulation A.

Successor function: add a queen to the leftmost empty column such that it is not attacked by any other queen.

In the worst case, how many paths do we need to search through? In other words, how many leaf nodes are there in the search tree?

The search tree is small enough that we can write it out by hand. See a separate handout for the search tree. There are exactly 6 leaf nodes.

We can calculate a rough upper bound: The first queen has 4 possible locations. The second queen has at most 2 possible positions (at least 1 row and 1 diagonal blocked by previous queen). The third queen has at most 1 possible position (at least 2 rows and 1 diagonal blocked by the two previous queens.). The last queen also has at most 1 possible position. The number of leaf nodes is at most 4 * 2 * 1 * 1 = 8.

Complete-state formulation

State: 4 queens on the board, one per column. r_0 r_1 r_2 r_3, where r_i is the row position of the queen in column i. r_i is in {0, 1, 2, 3}.

Start state: any

Goal state(s): 4 queens on the board, none attacking each other.

Possible successor functions

1. Move a queen to another square in the same column.

2. Swap the row numbers of two queens.

Example:

Consider the state 1032. Consider the queen in row 0 column 1 and another queen in row 2 and column 3.

After swapping these two queens, we generate the successor state 1230. In the successor state, the first queen is moved to row 2 and column 1 and the second queen is moved to row 0 and column 3.