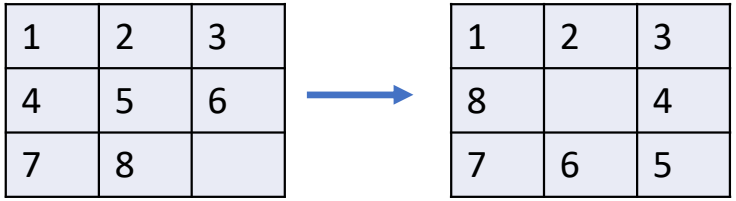
8-Puzzle

- Goal
  - Move tiles via empty space to get from initial state to goal state



- Environment
  - One empty space and 8 numbered, movable tiles in 3x3 frame

8-Puzzle

- Define the problem
  - Initial state:

1	2	3
4	5	6
7	8	

  - Goal test:

1	2	3
8		4
7	6	5

  - Actions: Move empty space Left, Right, Up or Down
  - Transition model: Given a state and action, this returns the resulting state; for example, if we apply left to the start state, the resulting state has 8 and blank switched.
  - Path cost: number of steps in the path

8 Queens

- Goal
  - Place 8 queens on a chessboard such that no queen attacks any other
- Environment
  - 8 queens on an 8x8 chessboard
- States
  - Any arrangement of 0 to 8 queens on the board
- Actions
  - Add/move a queen to any empty square
- Transition model
  - Returns the board with a queen added/moved to the specified square
- Goal Test
  - 8 queens on the board, none attacked
- Path cost
  - Number of moves

Vacuum world

- Goal
  - Suck up the dirt in all locations
- Environment
  - Vacuum in 2 possible space/locations
- States
  - The state is determined by both the agent location and the dirt locations. The agent is in one of two locations, each of which might or might not contain dirt.
- Actions
  - Left, Right, Suck
- Transition model
  - Left move vacuum left, Right move vacuum right, Suck will clean up the dirt. Moving Left in the leftmost location, moving Right in the rightmost square, and sucking in a clear square have no effect.
- Goal Test
  - Check all the locations are clean
- Path cost
  - Number of steps in the path

Touring problem

- Goal
  - Visit every city at least once starting and ending in “Location X”
- Environment
  - Route map of locations and transitions links between locations.
- Initial State: In(Location X), Visited({Location X})
- Goal State: In(Location X), Visited({all locations})
- Actions
  - Move to adjacent city
- Transition model
  - Result In(Location X), Visited({Location X}) Go (Location Y) => In (Location Y), Visited({Location X, Y})
- Goal Test
  - Check all the locations have been visited and current location is Location X
- Path cost
  - Number of steps in the path