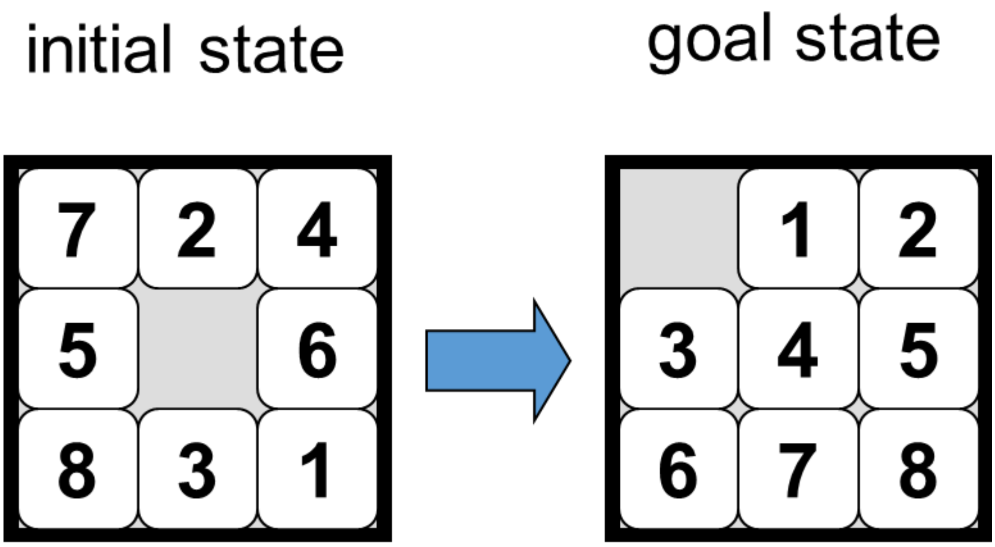
Lab 2

Puzzle Game

The (n^2-1)-puzzle is a sliding puzzle invented by Samuel Loyd in 1870's. In this game, (n^2-1) tiles are arranged on a (n x n) grid with one vacant space. The tiles are numbered from 1 to (n^2-1). Figure 1-left shows a possible configuration of the puzzle. The state of the puzzle can be changed by sliding one of the numbered tiles - adjacent to the vacant space - into the vacant space. The action is denoted by the direction, in which the numbered tile is moved. For each state, the set of possible actions is therefore a subset of {up; down; left; right}. The goal is to get the puzzle to the final state shown in Fig. 1-right by applying a sequence of actions. A puzzle configuration is considered as solvable, if there exists a sequence of actions which leads to the goal configuration. This holds true for exactly half of all possible puzzle configurations.



7	2	4
5		6
8	3	1

	1	2
3	4	5
6	7	8

Figure 1: A possible start state (left) and the goal state (right) of the 8-puzzle.

In general, for a given grid of width N, we can find out check if a n^2-1 puzzle is solvable or not by following below simple rules:

- If n is odd, then puzzle instance is solvable if number of inversions is even in the input state.
- If n is even, puzzle instance is solvable if
  - the blank is on an even row counting from the bottom (second-last, fourth-last, etc.) and number of inversions is odd.
  - the blank is on an odd row counting from the bottom (last, third-last, etc.) and number of inversions is even.
- For all other cases, the puzzle instance is not solvable.

Game Description

Given a 3×3 grid with 8 tiles and one space. This grid contains tiles numbered 1 through 8 along with one missing tile. The objective is to place the numbers on tiles using the space. The only moves allowed are those that slide a tile adjacent to the blank space into the blank space. We can slide four adjacent (left, right, above and below) tiles into the space, as shown in Figure 1.

a) Define states, actions and possible predicates for the 8-puzzle domain

b) Write the PDDL files (problem and domain files) for a planner of your choice to solve this problem. The <domain .pddl> should define the 8-puzzle actions and predicates. If you wish, you can hardcode the domain representation into your suggested planner but be sure to state this in your README. The <problem .pddl> contains the objects, init state, and goal state. The output of your planner should be a sequence of actions that solves the given problem.