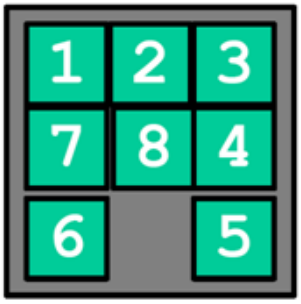
8-Puzzle game (Hill Climbing Search)

In an 8-puzzle game, we need to rearrange some tiles to reach a predefined goal state. Consider the following 8-puzzle board.

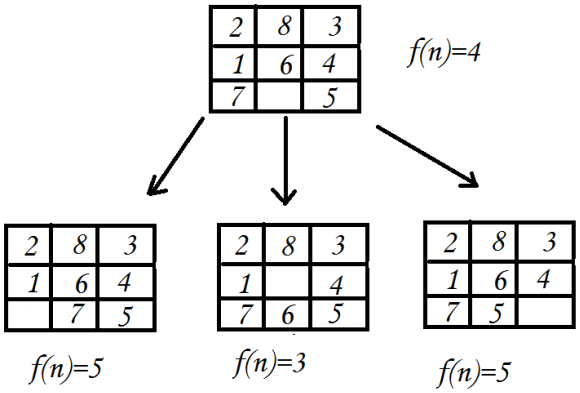
This is the goal state where each tile is in correct place. In this game, you will be given a board where the tiles aren't in the correct places. You need to move the tiles using the gap to reach the goal state.

Suppose f (n) can be defined as: ***the number of misplaced tiles***



In the above figure, tiles 6, 7 and 8 are misplaced. So f (n) = 3 for this case.

For solving this problem with hill climbing search, we need to set a value for the heuristic. Suppose the ***heuristic function h (n) is the lowest possible f (n) from a given state***. First, we need to know all the possible moves from the current state. Then we have to calculate f (n) (number of misplaced tiles) for each possible move. Finally we need to choose the path with lowest possible f (n) (which is our h (n) or heuristic).

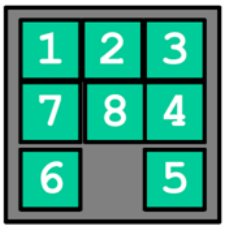


Consider the figure above. Here, 3 moves are possible from the current state. For each state we have calculated f (n). From the current state, it is optimal to move to the state with f (n) = 3 as it is closer to the goal state. So we have our h (n) = 3.

However, do you really think we can guarantee that it will reach the goal state? What will happen if we reach to a state where all the f (n) values are equal? This condition can be called a local maxima and this is the problem of hill climbing search. Therefore, we may get stuck in local maxima. In this scenario, you need to backtrack to a previous state to perform the search again to get rid of the path having local maxima.

Input:

You will be given the initial state of the board. The input will be given by row by row of the 3x3 grid of the 8-puzzle game. A digit from 1 to 9 will denote a tiles number. A 0 will denote the gap of the board.



For the above board, the input will be:

1 2 3
7 8 4
6 0 5

Task 1: Print the total cost (number of steps) needed to reach the goal state (if possible). Report if you reach a local maxima and get stuck. Print the board state in this scenario.

Task 2: You need get rid of any local maxima and reach the goal state anyway. Print the total cost needed to reach the goal state. Mention where you have backtracked to avoid local maxima (if any).