```
from collections import deque

class PuzzleState:
    def __init__(self, board, zero_pos, moves):
        self.board = board
        self.zero_pos = zero_pos
        self.moves = moves

    def is_goal(self):
        return self.board == [1, 2, 3, 4, 5, 6, 7, 8, 0]

    def get_neighbors(self):
        neighbors = []
        x, y = self.zero_pos
        directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

        for dx, dy in directions:
            new_x, new_y = x + dx, y + dy
            if 0 <= new_x < 3 and 0 <= new_y < 3:
                new_board = self.board[:]
                new_board[x * 3 + y], new_board[new_x * 3 + new_y] = new_board[new_x * 3 + new_y], new_board[x * 3 + y]
                neighbors.append((new_board, (new_x, new_y)))

        return neighbors

def bfs(start_board):
    start_pos = start_board.index(0)
    start_state = PuzzleState(start_board, (start_pos // 3, start_pos % 3), 0)

    queue = deque([start_state])
    visited = set()
    visited.add(tuple(start_board))

    while queue:
        current_state = queue.popleft()

        print(f"Current State:\n{format_board(current_state.board)}")

        if current_state.is_goal():
            return current_state.moves

        for neighbor_board, neighbor_pos in current_state.get_neighbors():
            if tuple(neighbor_board) not in visited:
                visited.add(tuple(neighbor_board))
                queue.append(PuzzleState(neighbor_board, neighbor_pos, current_state.moves + 1))

    return -1

def format_board(board):
    """Formats the board into a string for easy visualization."""
    return '\n'.join([' '.join(map(str, board[i:i+3])) for i in range(0, 9, 3)])

start_board = [ 1, 2, 3, 4, 5, 6, 0, 7, 8]
result = bfs(start_board)
print("BFS moves to solve the puzzle:", result)
```

```
Current State:
1 2 3
4 5 6
0 7 8
Current State:
1 2 3
0 5 6
4 7 8
Current State:
1 2 3
4 5 6
7 0 8
Current State:
0 2 3
1 5 6
4 7 8
Current State:
1 2 3
5 0 6
4 7 8
```

```
Current State:
1 2 3
4 0 6
7 5 8
Current State:
1 2 3
4 5 6
7 8 0
BFS moves to solve the puzzle: 2
```

Start coding or generate with AI.