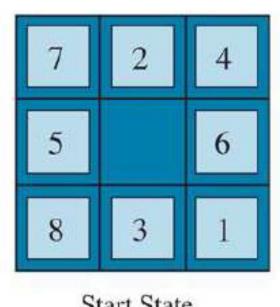
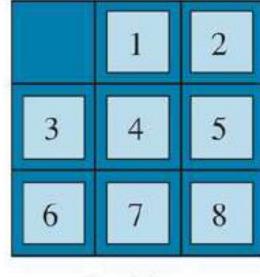
Your Task: 8-Puzzle

- From a random start state
- Check for solvability
- Generate **goal state**
- Using at least 2 different heuristic functions
- Provide estimate of algorithms complexity
- Implementation in Python or Java
- Measure memory effort (number of nodes expanded) and run time for each of 100 random states and each heuristics
- Provide mean and standard deviation of memory usage and execution time for each heuristics



Start State



Goal State

8-Puzzle Exercise Components

- 1. Check for **Solvability**
- 2. Implement two Heuristics: Hamming (misplaced tiles) and Manhattan
- 3. Compare two Heuristics (using 100 random searches each):
 - Memory Usage (number of expanded nodes in the search tree)
 - Computation Time
- 4. Comment Code, provide for each submodule at least:
 - What are Inputs and Outputs
 - What is the Function of the submodule
- 5. Don't just copy/paste code from Internet
 - Provide you own structure
 - Comment the code (see above)
 - Measure space (memory) and time complexity of each heuristics

8-Puzzle Task Documentation

Content

- 1. Short task description
- 2. Software architecture diagram
- 3. Short descriptions of modules and interfaces
- 4. Explain design decisions
- 5. Discussion and conclusions
 - Describe you experience
 - Provide a table with complexity comparisons of different heuristics
 - Possible improvements in future

Remember: Professional software developer work for others and an amateur for herself/himself.

Tasks and Schedule

- Provide **Design**:
 - Architecture: Modules and Interfaces (Functions, Inputs/Outputs)
 - Define Heuristics you are going to use
 - How would you test your code?
- Provide Code
- Deadline: 01.12.2024