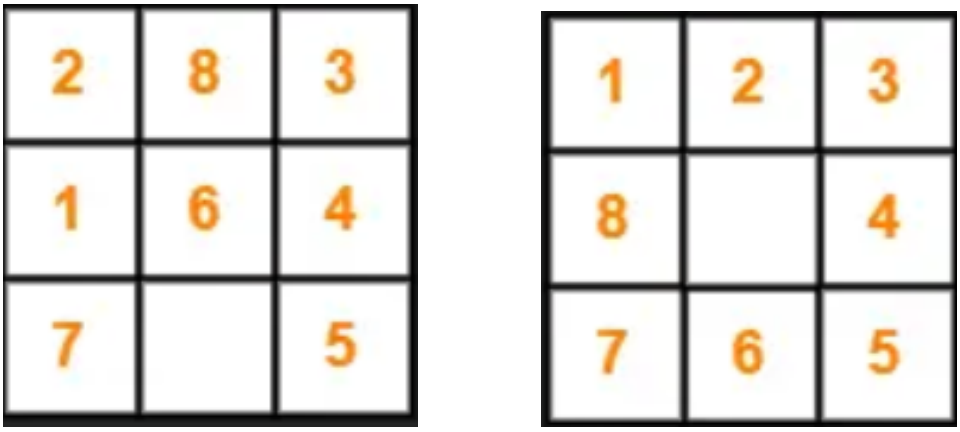
8-Puzzle Problem: Applying UCS, BFS, and A* Search

You are given the **Initial State** and **Goal State** of an 8-puzzle problem as follows:



(a) Initial State (b) Goal State

Figure 1: Initial and Goal States of the 8-Puzzle Problem

Tasks:

- **Solve the problem using Uniform Cost Search (UCS):**
  - Use $g(n)$ as the cost of reaching a node, where each move has a uniform cost of 1.
  - Provide the solution path, the total cost, and the search tree.
- **Solve the problem using Best First Search (BFS):**
  - Use the number of misplaced tiles as the heuristic $h(n)$, which counts the tiles not in their correct positions (excluding the blank tile).
  - Provide the solution path, heuristic values at each step, and the search tree.

- **Solve the problem using A* Search:**
  - Use the heuristic $h(n)$ as the number of misplaced tiles.
  - Use $f(n) = g(n) + h(n)$, where $g(n)$ is the cost from the start node, and $h(n)$ is the heuristic value.
  - Provide the solution path, $g(n)$, $h(n)$, and $f(n)$ values for each step, and the search tree.
- **Analyze Optimality and Completeness:**
  - Explain whether each algorithm guarantees an optimal solution and is complete (i.e., guarantees finding a solution if it exists).
- **Analyze the Impact of the Heuristic $h(n)$:**
  - Discuss the impact of the heuristic $h(n)$ on the performance of BFS and A* in finding an optimal solution.

Graph Search Problem: Applying UCS, BFS, and A* Search

You are given the graph with **Initial State (A)** and **Goal State(J)** for a city - travel problem as follows:

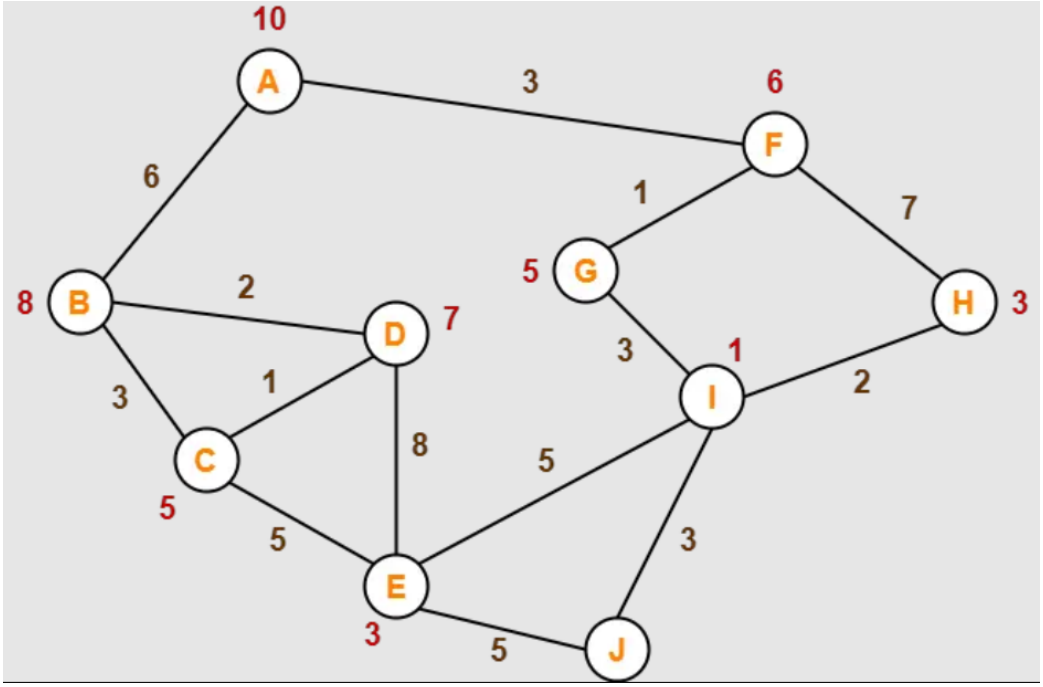


Figure 2: Graph of cities

Tasks:

- **Solve the problem using Uniform Cost Search (UCS):**
  - Use $g(n)$ as the cost of reaching a node, where each move has a uniform cost of 1.
  - Provide the solution path, the total cost, and the search tree.
- **Solve the problem using Best First Search (BFS):**

- Use the number of misplaced tiles as the heuristic $h(n)$, which counts the tiles not in their correct positions (excluding the blank tile).
- Provide the solution path, heuristic values at each step, and the search tree.

- **Solve the problem using A* Search:**
  - Use the heuristic $h(n)$ as the number of misplaced tiles.
  - Use $f(n) = g(n) + h(n)$, where $g(n)$ is the cost from the start node, and $h(n)$ is the heuristic value.
  - Provide the solution path, $g(n)$, $h(n)$, and $f(n)$ values for each step, and the search tree.

- **Analyze Optimality and Completeness:**
  - Explain whether each algorithm guarantees an optimal solution and is complete (i.e., guarantees finding a solution if it exists).

- **Analyze the Impact of the Heuristic $h(n)$:**
  - Discuss the impact of the heuristic $h(n)$ on the performance of BFS and A* in finding an optimal solution.