Simulated Annealing is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. It starts with a random solution to a problem and iteratively makes small changes to the solution, selecting solutions based on a probability mechanism that favours more optimal outcomes. Initially, it accepts worse solutions with high probability to explore the solution space, but as "temperature" decreases, it becomes more selective. The gradual reduction of acceptance probability helps the algorithm avoid getting stuck in local optima and converge to a global optimum.

Basic Algo:

```
Simulated Annealing (initial_solution, cost_function, neighbor_function, initial_temperature,
cooling_rate, max_iterations):
  current_solution = initial_solution
  current_cost = cost_function(current_solution)
  temperature = initial_temperature
  FOR iteration = 1 TO max_iterations DO:
    neighbor_solution = neighbor_function(current_solution)
    neighbor_cost = cost_function(neighbor_solution)
    delta_cost = neighbor_cost - current_cost
    IF delta_cost < 0 OR random (0, 1) < exp (-delta_cost / temperature) THEN:
      current_solution = neighbor_solution
      current_cost = neighbor_cost
    END IF
    temperature = temperature - 1
  END FOR
  RETURN current_solution
END
```

Cost Function:

a. Misplaced Tile
b. Manhattan distance

Question 2A -

<u>Observation 1:</u> -

Initial State: [[0, 2, 3], [1, 4, 5], [8, 7, 6]]
Goal State: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]

<u>Simulated Annealing</u>: -

Heuristic: - 1

Status: Solution Found
Number of states Explored: 98350
Number of states to optimal path: 717
Optimal Path Cost: 716
Execution Time: 5.116452693939209

Heuristic: - 2

Status: Solution Found
Number of states Explored: 34562

Number of states to optimal path: 769
Optimal Path Cost: 768
Execution Time: 3.7897114753723145

<u>Observation 2</u>: -
Initial State: [[0, 4, 3], [2, 8, 5], [1, 7, 6]]
Goal State: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
<u>Simulated Annealing</u>: -

Heuristic: - 1
Status: Solution Found
Number of states Explored: 138636
Number of states to optimal path: 73 Optimal Path Cost: 72

Execution Time: 7.285011053085327
Heuristic: - 2
Status: Solution Found
Number of states Explored: 88674
Number of states to optimal path: 469
Optimal Path Cost: 468
Execution Time: 9.285709381103516

<u>Observation 3</u>: -
Initial State: [[2, 3, 5], [7, 1, 4], [8, 0, 6]]
Goal State: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
<u>Simulated Annealing</u>: -

Heuristic: -1

Status: Solution Found
Number of states Explored: 599773
Number of states to optimal path: 594
Optimal Path Cost: 593
Execution Time: 31.61857581138611
Heuristic: -2
Status: Solution Found
Number of states Explored: 88674
Number of states to optimal path: 469
Optimal Path Cost: 468
Execution Time: 9.285709381103516

Question 2B –
The effectiveness of simulated annealing often depends on the following factors:

1. **Initial Solution:** The choice of the initial solution can influence the convergence and quality of the final solution. A good initial solution may allow the algorithm to converge faster and find better solutions. However, a random initial solution is often used if the problem does not provide a good starting point.
2. **Objective Function**: The quality of the objective function (also known as the cost or energy function) is essential. It should accurately reflect the problem's characteristics and

constraints. Here we are using two different objective function (a. Misplaced Tile and b. Manhattan Distance)

3. **Temperature Schedule:** The cooling schedule, which determines how the temperature decreases over time, plays a significant role. A well-designed cooling schedule should allow for sufficient exploration in the early stages of the search at higher temperatures and exploitation in the later stages at lower temperatures. The rate of cooling can impact convergence.
4. **Acceptance Probability**: The heuristic used to calculate the acceptance probability when considering a worse solution is critical.
```
boltz = math.exp((-0.25 * deltaEnergy) / temperature)
if random.uniform(0.0, 1.0) < boltz:
    bestNodeHere = i
```
5. **Stopping Criteria:** Deciding when to terminate the algorithm is essential. Setting appropriate stopping criteria ensures that the algorithm runs for an adequate amount of time to explore the solution space effectively without wasting computational resources. Common stopping criteria include reaching a predefined temperature or running for a specified number of iterations.
6. **Randomization:** Simulated annealing uses randomness in its search process. The randomization factor, including the choice of initial solutions and the acceptance of worse solutions, can influence the algorithm's behaviour and the diversity of solutions explored.

Question 2C-
In general, when comparing these heuristics in simulated annealing:

- Using h2(n) (Total Manhattan Distance) is likely to lead to more efficient convergence towards the goal state because it considers the actual distances of tiles from their correct positions, making it a more informed and accurate heuristic.
- Using h1(n) (Number of Displaced Tiles) may still work but may require more iterations or longer execution times since it doesn't provide as much information about the state's quality.

Question 2D-
In general, between the two heuristics provided (h1 and h2) for a simulated annealing, the heuristic h2 (Total Manhattan Distance) is likely to perform better. This is because h2 provides a more accurate estimate of the distance of the current state from the goal state, and simulated annealing benefits from having a more informative heuristic to guide the search.
Sometimes performance is also varying depending on the problem instance, the implementation of the simulated annealing algorithm, and the parameters used such as the cooling schedule and initial temperature.

Question 3A-
<u>Observation 1</u>: -
Initial State: [[0, 2, 3], [1, 4, 5], [8, 7, 6]]
Goal State: [[1, 2, 3], [8, 0, 4], [7, 6, 5]]
<u>Simulated Annealing</u>: -

Heuristic: - 1

Status: Solution Found
Number of states Explored: 98350
Number of states to optimal path: 717

Optimal Path Cost: 716
Execution Time: 5.116452693939209
<u>**Heuristic: - 2**</u>
Status: Solution Found
Number of states Explored: 34562
Number of states to optimal path: 769
Optimal Path Cost: 768
Execution Time: 3.7897114753723145
<u>Hill Climb</u>: -
Heuristic 1:
Status: Solution found
Number of states Explored: 1
Number of states to optimal path: 7
Optimal Path Cost: 6
Execution Time: 0.0
Heuristic: - 2
Status: Solution Found
Number of states Explored: 7
Number of states to optimal path: 7
Optimal Path Cost: 6
Execution Time: 0.0

<u>Observation 2</u>: -
Initial State: [[0, 4, 3], [2, 8, 5], [1, 7, 6]]
Goal State: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
<u>Simulated Annealing</u>: -

Heuristic: - 1

Status: Solution Found
Number of states Explored: 138636
Number of states to optimal path: 73 Optimal Path Cost: 72

Execution Time: 7.285011053085327
Heuristic: - 2
Status: Solution Found
Number of states Explored: 88674
Number of states to optimal path: 469
Optimal Path Cost: 468
Execution Time: 9.285709381103516

<u>Hill Climb</u>: -

Heuristic: - 1

Status: Solution does not exist
Heuristic: - 2
Status: Solution Found
Number of states Explored: 13
Number of states to optimal path: 13
Optimal Path Cost: 12
Execution Time: 0.0028214454650878906

<u>Observation 3</u>: -
Initial State: [[2, 3, 5], [7, 1, 4], [8, 0, 6]]
Goal State: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
<u>Simulated Annealing</u>: -

Heuristic: -1

Status: Solution Found
Number of states Explored: 599773
Number of states to optimal path: 594
Optimal Path Cost: 593
Execution Time: 31.61857581138611
Heuristic: -2
Status: Solution Found
Number of states Explored: 88674
Number of states to optimal path: 469
Optimal Path Cost: 468
Execution Time: 9.285709381103516
<u>Hill Climb</u>: -

Heuristic: - 1

Status: Solution does not exist
Heuristic: - 2
Status: Solution Found
Number of states Explored: 13
Number of states to optimal path: 13
Optimal Path Cost: 12
Execution Time: 0.0

Question 3B-
<u>Simulated Annealing:</u>
Simulated annealing typically starts from an initial solution and then iteratively explores neighbouring solutions. The number of iterations represents how many times the algorithm will go through this process. A higher number of iterations can allow the algorithm to explore the search space more thoroughly, increasing the chances of finding a good solution. However, the algorithm also considers a cooling schedule that reduces the acceptance probability of worse solutions over time, which is a critical aspect of simulated annealing.
<u>Hill Climbing:</u>
1.The efficiency of Hill Climbing algorithm depends on the quality of heuristic.
2. From observation 2 it is seen that for the same Initial State and Goal State, h1(n) is showing Solution does not exist but if we use h2(n) it is showing Solution found. So, we can say that quality of heuristic h1(n) is not good as compared to h2(n) as it is showing no solution for the same Initial State and Goal State.

Question 3C-
Simulated Annealing have performed better for this particular problem than Hill Climbing algorithm. Following are the reasons for the performance of both algorithm: -

Simulated Annealing:

- Results tend to be of higher quality.
- Better at finding global optima or near-optimal solutions.
- Suitable for complex problems with many local optima.
- May require more computational resources and time.
- Effective in uncertain or rugged search spaces.

Hill Climbing:

- Results are often of lower quality.
- Quick convergence to a local optimum.
- Efficient in terms of computational resources.
- Suitable for simple problems with well-defined search spaces.
- Prone to getting stuck in local optima.

Question 3D-
Simulated Annealing have performed better for this particular problem than Hill Climbing algorithm. Simulated annealing is often better than hill climbing because it can find global optima, escape local optima and handle complex and uncertain optimization problems, whereas hill climbing tends to converge quickly but gets stuck in local optima.