AI Assignment-2

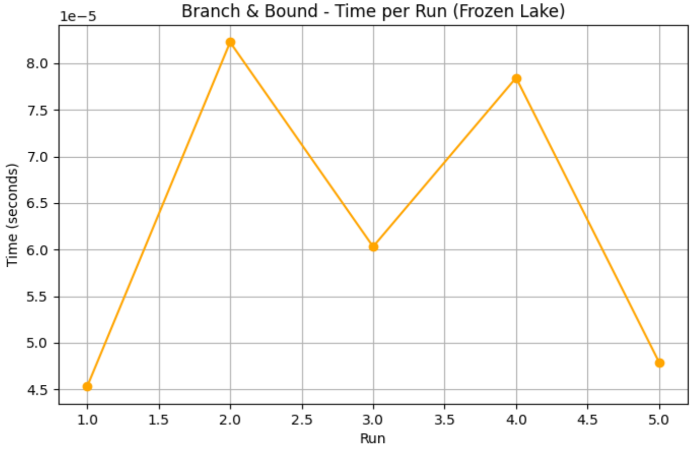
Implementations

- Branch and Bound, Iterative Deepening A* algorithms on the Frozen Lake environment.
- Hill Climbing and Simulated Annealing algorithms on the Traveling Salesman Problem.

Team Members

- Dharmendra Chauhan (CS24M115)
- Kodela Phanindra(CS24M121)

Branch and Bound algorithm on the Frozen Lake environment



Average time = 6.292e-5 sec
Average reward = 1
Average steps = 7

Heuristic function used:

<u>None</u>

Because it uses pure cost-based search(uniform cost) with a priority queue(i.e., it explores all paths but prioritizes shorter ones). It doesn’t estimate the cost to goal.

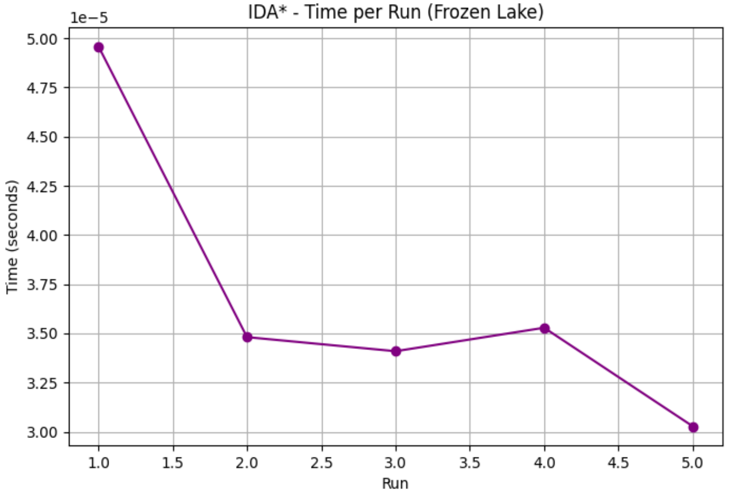
<u>Characteristics:</u>

1. Explores paths in increasing cost order, avoiding longer paths unless needed.
2. In this case, the optimal solution is very shallow in the tree (cost 6), and the total state space is small, so B&B is very efficient.
3. Uses a priority queue, which helps it avoid bad branches early.

<u>Observation:</u>

For simple and deterministic maps, this algorithm guarantees optimality by cost expansion.

Iterative Deepening A algorithm on the Frozen Lake environment*



Average time = 3.678e-5 sec
Average reward = 1
Average steps = 7

Characteristics:

1. Uses a heuristic to guide its search, which is perfectly admissible and consistent in grid-based environments.
2. Although it performs multiple iterations, the heuristic is so good and the solution is shallow enough that the overhead is minimal.
3. Recursive depth-limited search incurs some additional function calls, but it’s fast due to the small state space.

Heuristic function used:

Manhattan Distance in 4x4 grid

```
def heuristic(state):
    x1, y1 = state % 4, state // 4
    x2, y2 = goal_state % 4, goal_state // 4
    return abs(x1 - x2) + abs(y1 - y2)
```

Why IDA* faster on avg.

1. On a very small graph like this, the heuristic helps prune early, so even though IDA* restarts depth-first searches at increasing thresholds, it spends less time per node.
2. B&B still generates all neighbors and pushes them into a heap, even though the graph is simple, which might have a tiny overhead.

Observation:

For simple, deterministic maps, IDA* benefits from a heuristic and limits memory usage (depth-first nature).

Hill Climbing algorithm on the Traveling Salesman Problem environment

Heuristic Function used:

Total path cost (distance) of a tour

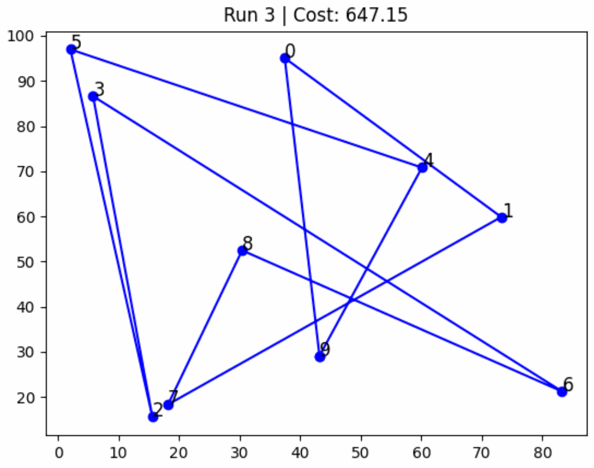
```
def path_cost(path):
    return sum( dist_matrix[path[i]][path[i+1]] for i in range(len(path)-1))
```

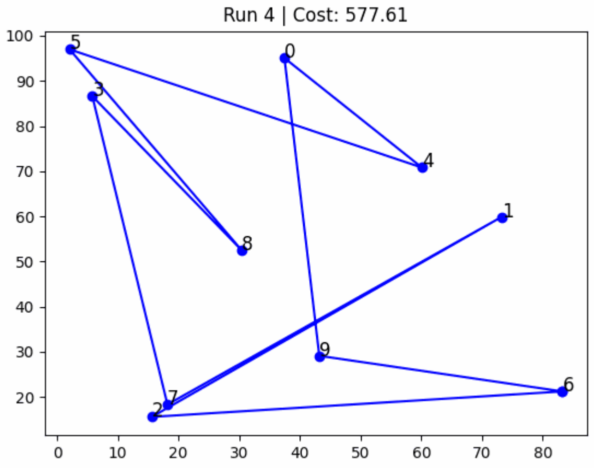
Characteristics:

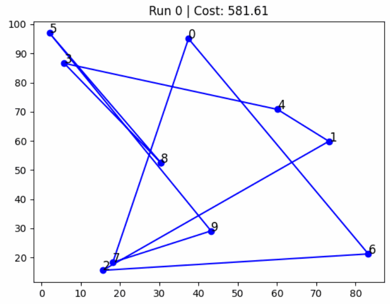
1. Exploitation-focused: always picks the best neighbor (greedy).
2. If it finds a local minimum, it gets stuck – there's no escape.

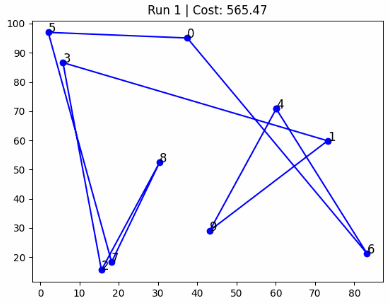
Observation:

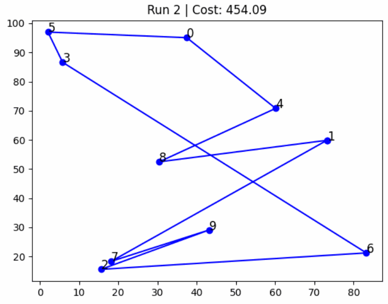
It fails once (Run 2), ending up in a worse minimum due to its inability to escape local traps.











Simulated Annealing algorithm on the Traveling Salesman Problem environment

Heuristic Function used:

Total path cost (distance) of a tour

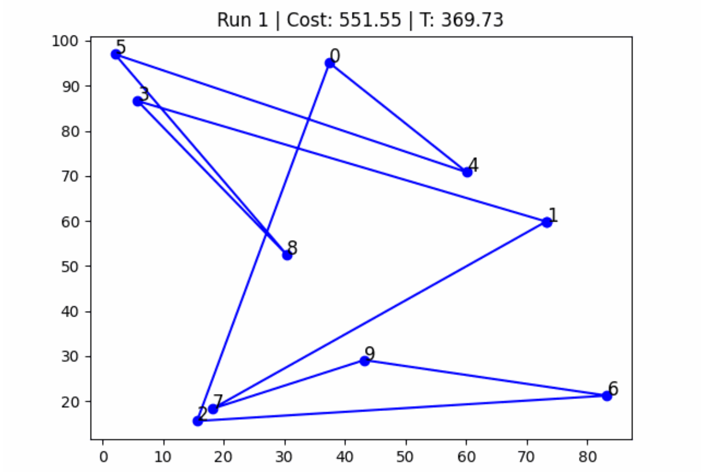
```
def path_cost(path):
    return sum( dist_matrix[path[i]][path[i+1]] for i in range(len(path)-1))
```

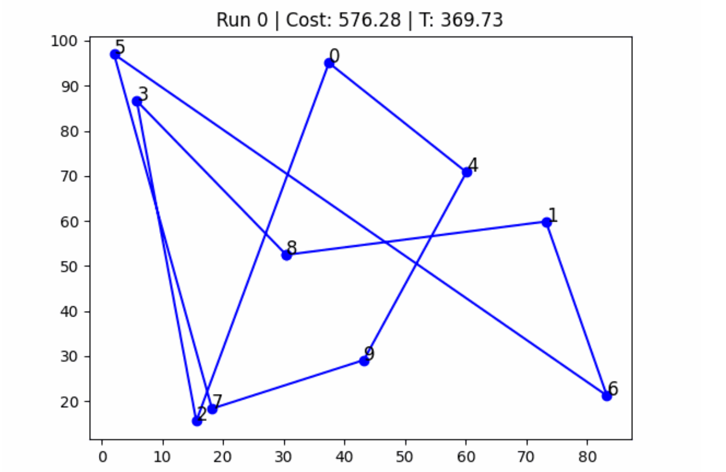
Characteristics:

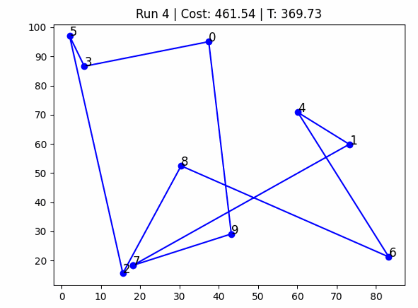
1. Balances exploration and exploitation using a temperature parameter.
2. Early on, it accepts worse solutions with a high probability -> exploration.
3. As temperature cools, behavior becomes greedy -> exploitation, which leads to more consistent convergence to global optima, despite running more iteration.

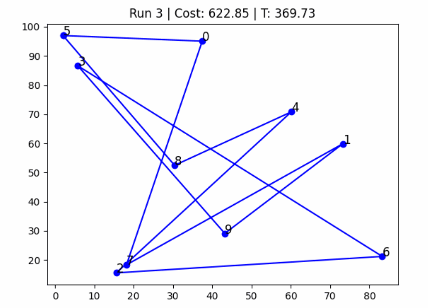
Observation:

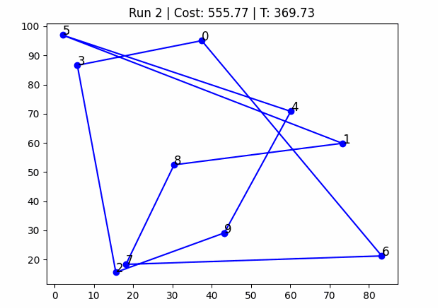
It always find the same best cost which suggests it’s robust local minima.











Performance analysis b/w HC and SA on TSP

Situation	Preferred Algorithm	Why
Need quick approximations	Hill Climbing	Fast, low overhead
Need reliable, better-quality results	Simulated Annealing	Can escape local minima
Very large search space	Simulated Annealing	Better global exploration
Small problem + time-sensitive	Hill Climbing	Less computationally expensive

<u>Efficiency Tradeoff</u>

1. Hill Climbing is faster with fewer iterations.
2. Simulated Annealing takes more time and effort, but it’s worth it due to the higher reliability and better average cost.

<u>Key Observation:</u>

1. If the number of cities are increased, Hill Climbing will likely struggle more and Simulated Annealing will shine even more clearly.
2. The complexity of the landscape increases, and greedy strategies usually fall short.