Report for Project 2

I. Findings

My Java code offers two algorithms: Steepest Ascent Hill Climbing and Genetic Algorithm to solve the Eight Queens puzzle. In the Steepest Ascent Hill Climbing method, a random initial board configuration is generated, and the algorithm iteratively explores neighboring states by moving a single queen at a time. It selects the move that minimizes the number of conflicts the most, and if it reaches a local minimum, it restarts with a new random initial state. The Genetic Algorithm starts with a population of random board configurations and evolves the population over multiple generations. It selects the fittest individuals to be parents, applies recombination and mutation to produce new individuals, and repeats this process until a solution is found or a maximum number of generations is reached. Both methods provide search cost and time taken metrics, allowing comparison between their performances in solving the puzzle.

II. Table Analysis:

Algorithm	Number of Instances Generated (instances)	Average Search Cost (search costs)	Average Time Taken (milliseconds)	Average Percentage of Problem Solved (%)
Steepest Ascent Hill Climbing	200	3.83	0.17	12
Genetic Algorithm	200	1842.11	6.03	29

The Steepest Ascent Hill Climbing algorithm has a relatively low average percentage of puzzles solved (12%). This can be attributed to the algorithm's local search nature, which means it can easily get stuck in local minima. Local minima occur when there are no neighboring states with fewer conflicts, and the algorithm cannot make progress toward a global solution. The algorithm explores the immediate neighbors of the current state and moves to the one with the minimum number of conflicts. However, this can lead to limited exploration of the solution space, preventing it from reaching a global solution.

The Genetic Algorithm shows better performance compared to Steepest Ascent Hill Climbing, with a higher average percentage of puzzles solved (29%). The genetic algorithm's ability to maintain diversity within the population allows it to explore a broader range of solutions, increasing the chances of finding a global solution to the problem. The selection, crossover, and mutation operations enable the algorithm to gradually evolve towards better solutions over multiple generations. However, the search cost for this algorithm is relatively high because the genetic algorithm often requires a larger number of generations to converge to a solution, which might have contributed to the higher search cost.

III. Output Examples:

```
Please select an option:
[1] Steepest Ascent Hill Climbing
[2] Genetic Algorithm
[3] Both
[4] Exit
Enter your choice: 1

Steepest Ascent Hill Climbing:
Enter the number of 8-Queen instances to generate: 10
```

```
Instance 7:
Initial Puzzle:
. Q . . . . . .
. . . . Q . . .
. . . . Q . . .
Q . . . . . . .
. . Q . . . . .
. Q . . . . . .
. . . . Q . . .
. Q . . . . . .
```

```
Solution using Steepest-Ascent Hill Climbing:
. . Q . . . . .
. . . . . Q . .
. . . . . . . Q
Q . . . . . .
. . . Q . . . .
. . . . . Q .
. . . . Q . . .
. Q . . . . . .
Search Cost: 5
Time Taken: 0 milliseconds
```

```
Please select an option:
[1] Steepest Ascent Hill Climbing
[2] Genetic Algorithm
[3] Both
[4] Exit
Enter your choice: 2

Genetic Algorithm:
Enter the number of 8-Queen instances to generate: 2
```

```
Instance 1:
Initial Puzzle:
. . . . . Q . .
. . Q . . . . .
. Q . . . . . .
. . . . . . Q .
. . . . . Q . .
Q . . . . . . .
. . . . Q . . .
. . . . Q . . .
```

```
Genetic Algorithm reached the maximum number of generations without finding a solution.
```

```
Instance 2:
Initial Puzzle:
. . . . Q . . .
. . Q . . . . .
. Q . . . . . .
. . Q . . . . .
. . . . . . Q .
Q . . . . . . .
Q . . . . . . .
. . . . . . . Q
```

```
Solution using Genetic Algorithm:
. Q . . . . . .
. . . . . Q . .
Q . . . . . . .
. . . . . . Q .
. . . Q . . . .
. . . . . . . Q
. . Q . . . . .
. . . . Q . . .
Generation: 41
Search Cost: 205050
Time Taken: 5 milliseconds
```