ARTIFICIAL INTELLIGENCE

ALI SHAN

FA21-BCS-080

BCS-7A

HILL CLIMBING CODE

Java:

```java
import java.util.ArrayList;
public class SwapBasedOptimization {
public static void main(String[] args) {
    System.out.println("=============== Swap-Based Optimization ===============");
int best = h.computeCost(initial);
    ArrayList<Integer> currentSolution = new ArrayList<>(initial);
    ArrayList<Integer> bestSolution = new ArrayList<>(initial);
    int score;
    for (int i = 0; i < 1000; i++) {
      ArrayList<Integer> newSolution = h.swaps(currentSolution);
      score = h.computeCost(newSolution);
      if (score < best) {
        best = score;
        bestSolution = new ArrayList<>(newSolution);
        System.out.println("Improved Solution: " + bestSolution);
        System.out.println("Updated Cost: " + best);
      }
    }
```

```java
        System.out.println("============== Optimization Complete ==============");

        System.out.println("Final Best Solution: " + bestSolution);

        System.out.println("Final Best Cost: " + best);

    }

}
```

Python:

```python
print("============== Swap-Based Optimization ==============")

best = h.compute_cost(initial)

current_solution = initial.copy()

best_solution = initial.copy()

score = None


for i in range(1000):

    new_solution = h.swaps(current_solution)

    score = h.compute_cost(new_solution)


    if score < best:

        best = score

        best_solution = new_solution.copy()

        print(f"Improved Solution: {best_solution}")

        print(f"Updated Cost: {best}")


print("============== Optimization Complete ==============")

print(f"Final Best Solution: {best_solution}")

print(f"Final Best Cost: {best}")
```

OUTPUT:

=============== Swap-Based Optimization ==============

Improved Solution: [6, 4, 7, 3]

Updated Cost: 20

Improved Solution: [6, 7, 4, 3]

Updated Cost: 20

Improved Solution: [7, 6, 4, 3]

Updated Cost: 20

Improved Solution: [4, 6, 7, 3]

Updated Cost: 20

Improved Solution: [4, 7, 6, 3]

Updated Cost: 20

Improved Solution: [6, 7, 4, 3]

Updated Cost: 20

...

============== Optimization Complete ==============

Final Best Solution: [4, 6, 7, 3]

Final Best Cost: 20