**ARTIFICIAL INTELLIGENCE**

**HUSSAIN ALI**

**FA21-BCS-066**

**BCS-7A**

**HILL CLIMBING CODE IN JAVA AND ITS PYTHON CONVERSION**

**Java:**

import java.util.ArrayList;

public class Optimization {

public static void main(String[] args) {

System.out.println("=============== Iterative Cost Optimization ===============");

int bestCost = h.computeCost(initial);

int currentCost;

ArrayList<Integer> currentSolution = new ArrayList<>(initial);

ArrayList<Integer> bestSolution = new ArrayList<>(initial);

ArrayList<ArrayList<Integer>> allSolutions = new ArrayList<>();

for (int i = 0; i < initial.size(); i++) {

int removedElement = currentSolution.remove(i);

currentSolution.add(removedElement);

currentCost = h.computeCost(currentSolution);

if (currentCost < bestCost) {

bestCost = currentCost;

bestSolution = new ArrayList<>(currentSolution);

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

System.out.println("Updated Cost: " + bestCost);

}

allSolutions.add(new ArrayList<>(currentSolution));

}

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

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

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

}

}

**Python:**

print("=============== Iterative Cost Optimization ===============")

best\_cost = h.compute\_cost(initial)

current\_solution = initial.copy()

best\_solution = initial.copy()

all\_solutions = []

for i in range(len(initial)):

removed\_element = current\_solution.pop(i)

current\_solution.append(removed\_element)

current\_cost = h.compute\_cost(current\_solution)

if current\_cost < best\_cost:

best\_cost = current\_cost

best\_solution = current\_solution.copy()

print(f"Updated Best Solution: {best\_solution}")

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

all\_solutions.append(current\_solution.copy())

Print("=============== Optimization Complete ===============")

print(f"Final Best Solution: {best\_solution}")

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

**OUTPUT:**

=============== Iterative Cost Optimization ===============

Updated Best Solution: [2, 4, 7, 1]

Updated Cost: 14

Updated Best Solution: [4, 7, 1, 2]

Updated Cost: 13

Updated Best Solution: [7, 1, 2, 4]

Updated Cost: 12

Updated Best Solution: [1, 2, 4, 7]

Updated Cost: 10

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

Final Best Solution: [1, 2, 4, 7]

Final Best Cost: 10