1. 143. You are given a cost matrix where each element cost[i][j] represents the cost of assigning worker i to task j. Develop a program that utilizes exhaustive search to solve the assignment problem. The program should Define a function total_cost(assignment, cost_matrix) that takes an assignment (list representing worker-task pairings) and the cost matrix as input. It iterates through the assignment and calculates the total cost by summing the corresponding costs from the cost matrix Implement a function assignment_problem(cost_matrix) that takes the cost matrix as input and performs the following Generate all possible permutations of worker indices (excluding repetitions).
Test Cases:
Input
   1. **Simple Case:** Cost Matrix:
      [[3, 10, 7],
      [8, 5, 12],
      [4, 6, 9]]

Code:
```
def total_cost(assignment, cost_matrix):
    total = 0
    for worker, task in enumerate(assignment):
        total += cost_matrix[worker][task]
    return total

from itertools import permutations

def assignment_problem(cost_matrix):
    num_workers = len(cost_matrix)
    all_permutations = permutations(range(num_workers))

    min_cost = float('inf')
    best_assignment = None

    for perm in all_permutations:
        current_cost = total_cost(perm, cost_matrix)
        if current_cost < min_cost:
            min_cost = current_cost
            best_assignment = perm

    return best_assignment, min_cost

# Example input
cost_matrix = [
    [3, 10, 7],
    [8, 5, 12],
    [4, 6, 9]
]

# Solve the assignment problem
best_assignment, min_cost = assignment_problem(cost_matrix)
print("Best Assignment:", best_assignment)
print("Minimum Cost:", min_cost)
```

output:
```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/problems.py
Best Assignment: (2, 1, 0)
Minimum Cost: 16
PS C:\Users\karth>
```

Time complexity:f(n)=0(nlogn)