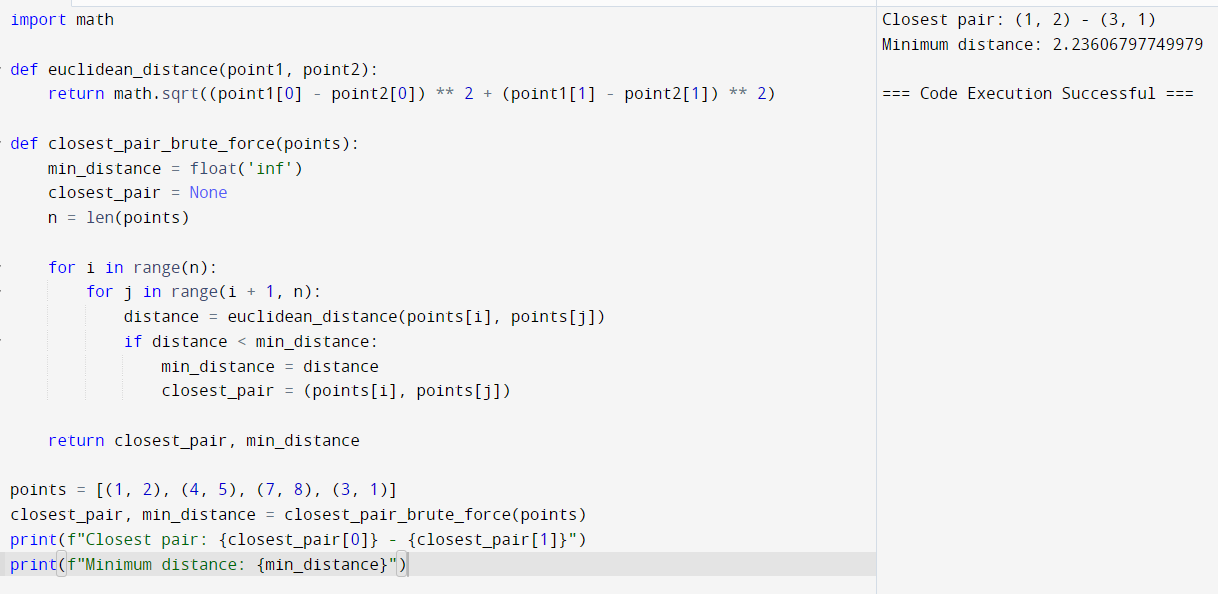
1. Write a program that finds the closest pair of points in a set of 2D points using the brute

force approach.



2. Write a program to find the closest pair of points in a given set using the brute force

approach. Analyze the time complexity of your implementation. Define a function to

calculate the Euclidean distance between two points. Implement a function to find the

closest pair of points using the brute force method. Test your program with a sample set

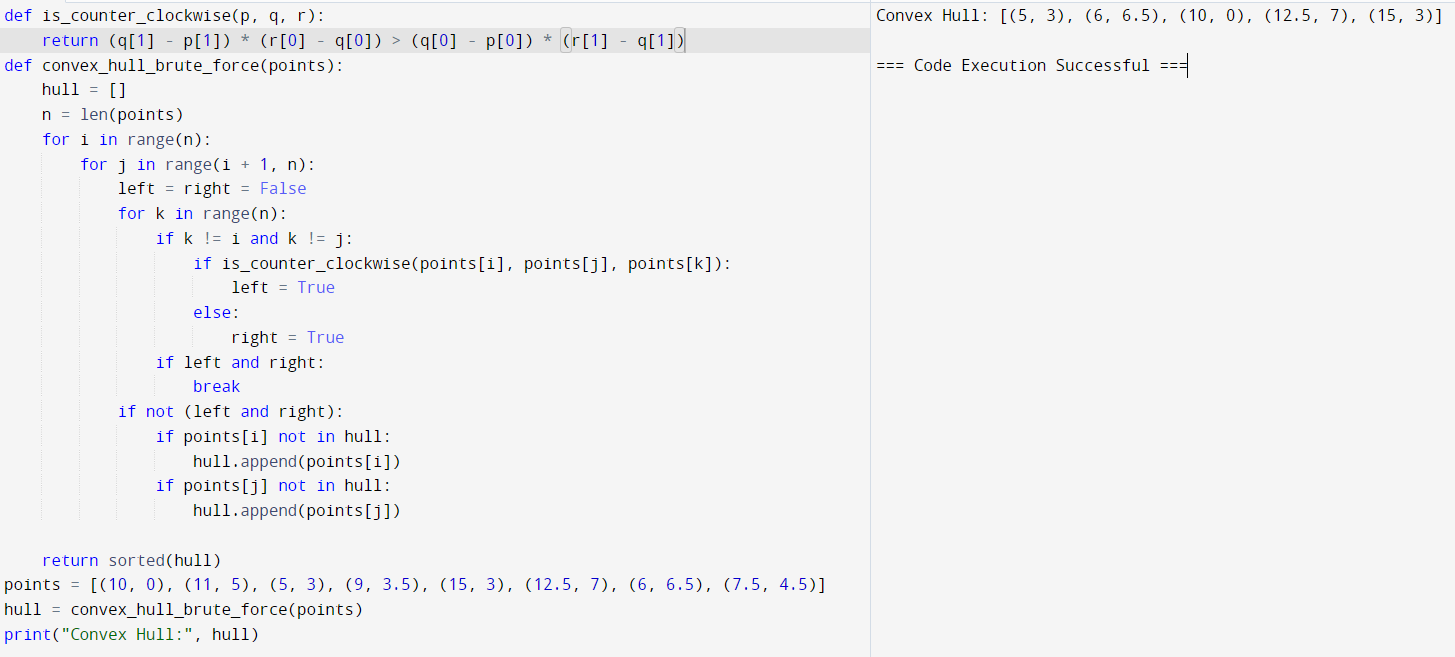
of points and verify the correctness of your results. Analyze the time complexity of your

implementation. Write a brute-force algorithm to solve the convex hull problem for the

following set S of points? P1 (10,0)P2 (11,5)P3 (5, 3)P4 (9, 3.5)P5 (15, 3)P6 (12.5, 7)P7

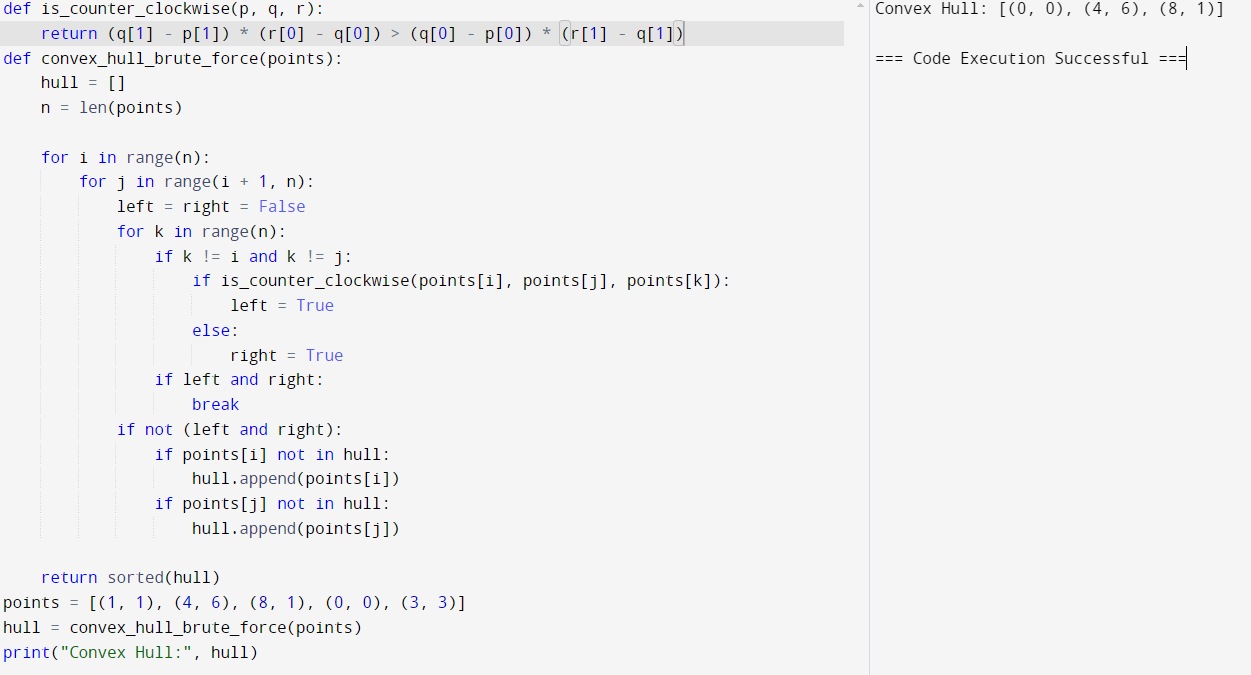
(6, 6.5)P8 (7.5, 4.5).How do you modify your brute force algorithm to handle multiple

points that are lying on the sameline?



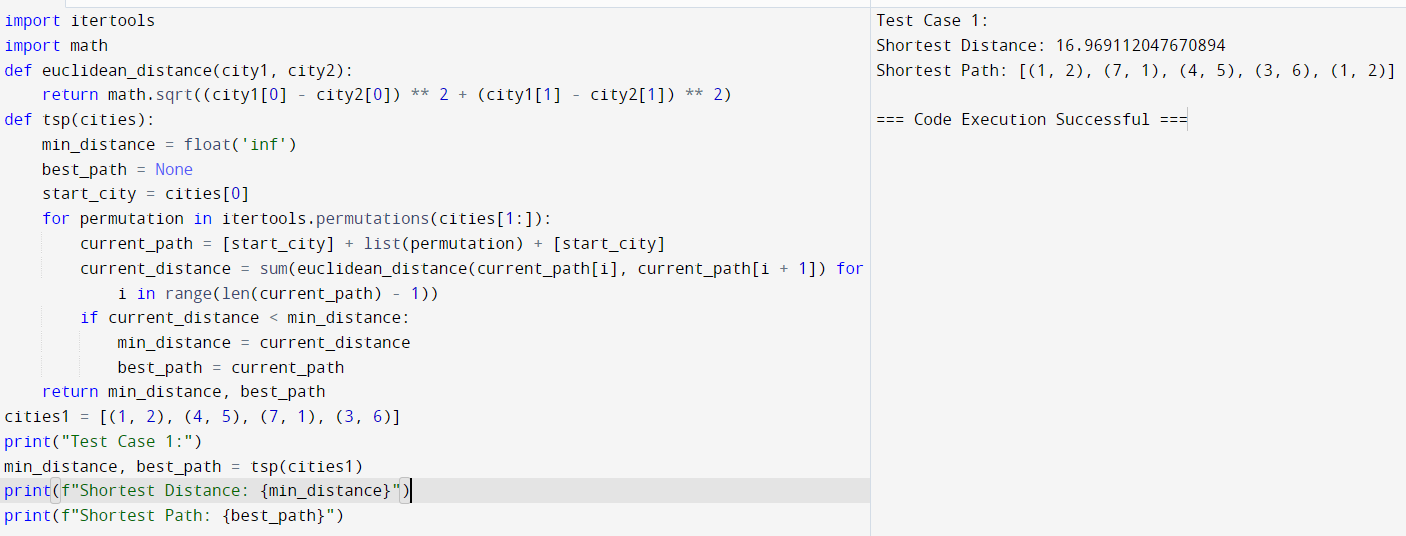
3. Write a program that finds the convex hull of a set of 2D points using the brute force

approach.



4. You are given a list of cities represented by their coordinates. Develop a program that

utilizes exhaustive search to solve the TSP.



5. 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,

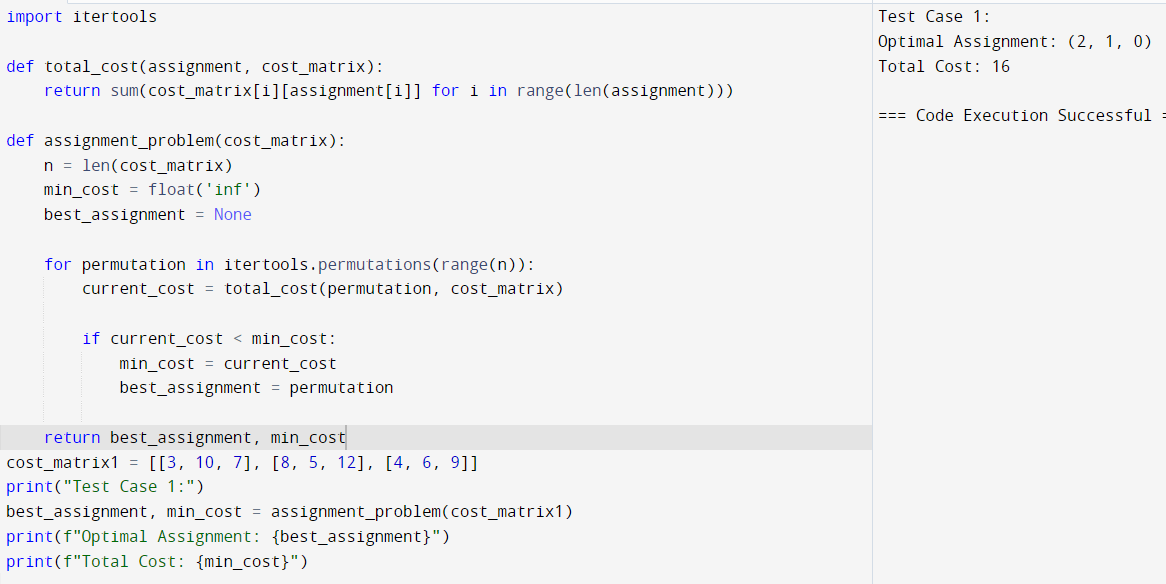
cost\_matrix) that takes an assignment (list representing worker-task pairings) and the

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).



6. You are given a list of items with their weights and values. Develop a program that

utilizes exhaustive search to solve the 0-1 Knapsack Problem.

