

## **Exp-2.9**

### **Title:**

Find the closest pair of points in a set using brute force.

### **Aim:**

To design and implement a Python program that identifies the pair of points with the minimum distance between them using the brute force approach.

### **Procedure:**

1. Read the input list of 2D points as tuples (x, y).
2. Initialize variables to store the minimum distance and the closest pair.
3. Iterate through all pairs (i, j) with  $i < j$  of points.
4. Calculate the Euclidean distance between each pair.
5. Update the minimum distance and closest pair if a smaller distance is found.
6. Print the closest pair and their minimum distance.

### **Algorithm:**

1. Start
2. Initialize min\_dist to a very large number, and closest\_pair as empty.
3. For each point p1 at index i in points:
  - For each point p2 at index  $j > i$ :
    - Compute distance between p1 and p2.
    - If distance < min\_dist: update min\_dist and closest\_pair.
4. Return closest\_pair and min\_dist.
5. End.

**Input:**

4  
1 2  
4 5  
7 8  
3 1

**Output:**

Closest pair: (1, 2) - (3, 1)

Minimum distance: 1.4142135623730951

**Program:**

```
import math

def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

def closest_pair_brute_force(points):
    min_distance = float('inf')
    closest_pair = (None, None)
    for i in range(len(points)):
        for j in range(i + 1, len(points)):
            dist = euclidean_distance(points[i], points[j])
            if dist < min_distance:
                min_distance = dist
                closest_pair = (points[i], points[j])
    return closest_pair, min_distance

n = int(input("Enter the number of points: "))
points = []
print("Enter each point as two space-separated numbers (e.g., '3 4'):")
```

```

for _ in range(n):
    x, y = map(float, input().split())
    points.append((x, y))

pair, distance = closest_pair_brute_force(points)

print(f"\nClosest pair: {pair[0]} - {pair[1]}")
print(f"Minimum distance: {distance}")

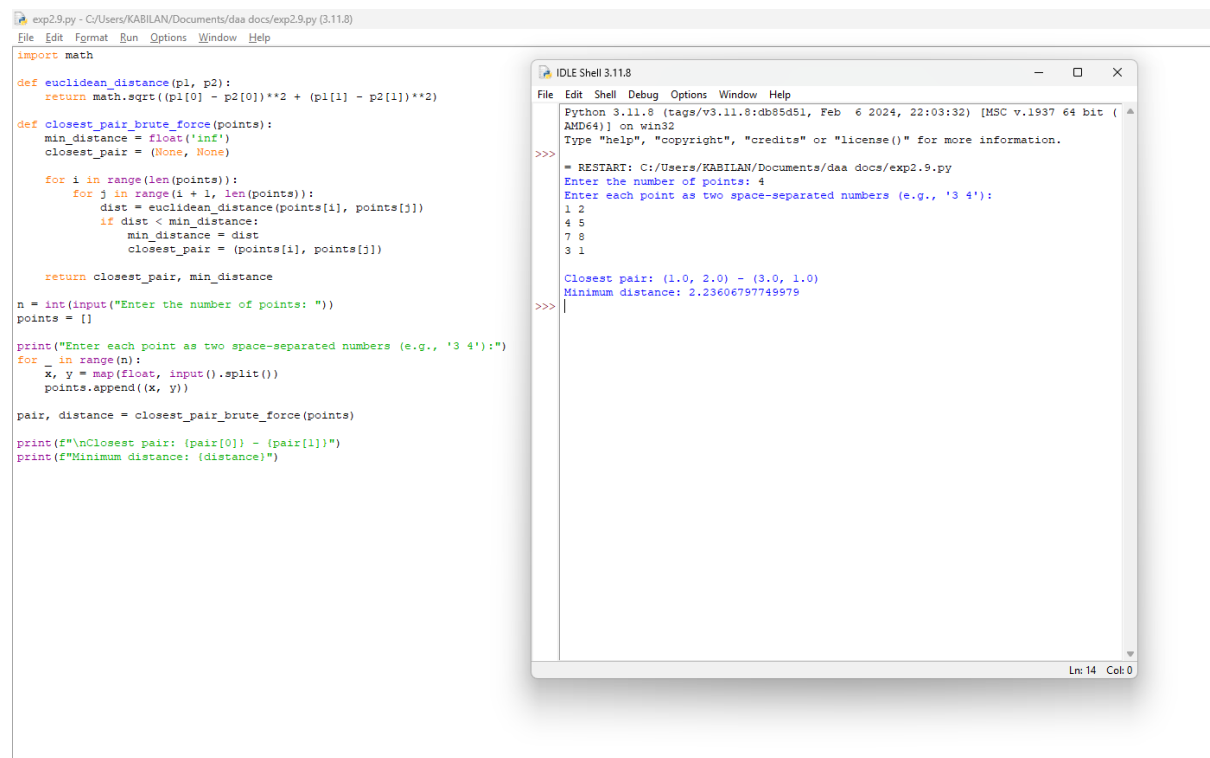
```

## Performance Analysis:

**Time Complexity:**  $O(n^2)$

**Space Complexity:**  $O(n)$

## Program Output:



The image shows a screenshot of a Python IDE with two windows. The left window displays the source code for a program that finds the closest pair of points using a brute force algorithm. The right window shows the execution output, where the user has entered 4 points: (1, 2), (4, 5), (7, 8), and (3, 1). The program outputs the closest pair as (1.0, 2.0) - (3.0, 1.0) and the minimum distance as 2.23606797749979.

```

exp2.9.py - C:/Users/KABILAN/Documents/daa docs/exp2.9.py (3.11.8)
File Edit Format Run Options Window Help

import math
def euclidean_distance(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

def closest_pair_brute_force(points):
    min_distance = float('inf')
    closest_pair = (None, None)

    for i in range(len(points)):
        for j in range(i + 1, len(points)):
            dist = euclidean_distance(points[i], points[j])
            if dist < min_distance:
                min_distance = dist
                closest_pair = (points[i], points[j])

    return closest_pair, min_distance

n = int(input("Enter the number of points: "))
points = []

print("Enter each point as two space-separated numbers (e.g., '3 4'):")
for _ in range(n):
    x, y = map(float, input().split())
    points.append((x, y))

pair, distance = closest_pair_brute_force(points)

print(f"\nClosest pair: {pair[0]} - {pair[1]}")
print(f"Minimum distance: {distance}")

```

```

IDLE Shell 3.11.8
File Edit Shell Debug Options Window Help

Python 3.11.8 (tags/v3.11.8:db85d51, Feb  6 2024, 22:03:32) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/KABILAN/Documents/daa docs/exp2.9.py
Enter the number of points: 4
Enter each point as two space-separated numbers (e.g., '3 4'):
1 2
4 5
7 8
3 1

Closest pair: (1.0, 2.0) - (3.0, 1.0)
Minimum distance: 2.23606797749979
>>>

```

## Result:

Thus the given program Closest Pair Brute Force is executed and got output successfully.