84. Closest pair of points using divide and conquer

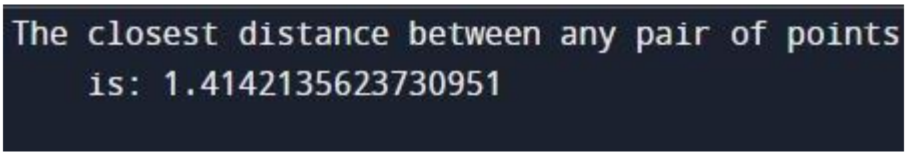
AIM: To find the Closest pair of points using divide and conquer

PROGRAM:

```
import math


def closest_pair(points):
    points.sort(key=lambda x: x[0])
    def closest_pair_rec(points, n):
        if n <= 3:
            return brute_force(points)
        mid = n // 2
        mid_point = points[mid]
        left_points = points[:mid]
        right_points = points[mid:]
        d_left = closest_pair_rec(left_points, mid)
        d_right = closest_pair_rec(right_points, n - mid)
        d = min(d_left, d_right)
        strip = []
        for point in points:
            if abs(point[0] - mid_point[0]) < d:
                strip.append(point)
        strip_closest = strip_closest_pair(strip, len(strip), d)
        return min(d, strip_closest)
    def brute_force(points):
        min_dist = float('inf')
        for i in range(len(points)):
            for j in range(i + 1, len(points)):
                dist = distance(points[i], points[j])
                if dist < min_dist:
                    min_dist = dist
        return min_dist
    def strip_closest_pair(strip, size, d):
        min_dist = d
        strip.sort(key=lambda x: x[1])
        for i in range(size):
            for j in range(i + 1, size):
                if strip[j][1] - strip[i][1] >= min_dist:
                    break
                dist = distance(strip[i], strip[j])
                if dist < min_dist:
                    min_dist = dist
        return min_dist
    def distance(p1, p2):
        return math.sqrt((p1[0] - p2[0])*2 + (p1[1] - p2[1])*2)
    return closest_pair_rec(points, len(points))
points = [(2, 3), (12, 30), (40, 50), (5, 1), (12, 10), (3, 4)]
closest_distance = closest_pair(points)
print(f"The closest distance between any pair of points is: {closest_distance}")
```

OUTPUT:



```
The closest distance between any pair of points
    is: 1.4142135623730951
```

TIME COMPLEXITY: O (n log n)