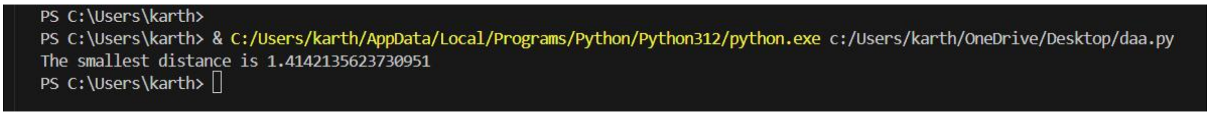
7. Closest pair of points using divide and conquer.

Code:

```
import math
def dist(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)
def brute_force(points):
    min_dist = float('inf')
    n = len(points)
    for i in range(n):
        for j in range(i + 1, n):
            if dist(points[i], points[j]) < min_dist:
                min_dist = dist(points[i], points[j])
    return min_dist
def strip_closest(strip, d):
    min_dist = d
    strip.sort(key=lambda point: point[1])
    for i in range(len(strip)):
        j = i + 1
        while j < len(strip) and (strip[j][1] - strip[i][1]) < min_dist:
            min_dist = min(min_dist, dist(strip[i], strip[j]))
            j += 1
    return min_dist
def closest_pair_rec(points):
    n = len(points)
    if n <= 3:
        return brute_force(points)
    mid = n // 2
    mid_point = points[mid]
    dl = closest_pair_rec(points[:mid])
    dr = closest_pair_rec(points[mid:])
    d = min(dl, dr)
    strip = [point for point in points if abs(point[0] - mid_point[0]) < d]
    return min(d, strip_closest(strip, d))
def closest_pair(points):
    points.sort(key=lambda point: point[0])
    return closest_pair_rec(points)
points = [(2, 3), (12, 30), (40, 50), (5, 1), (12, 10), (3, 4)]
min_distance = closest_pair(points)
print(f"The smallest distance is {min_distance}")
```

output:



```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/daa.py
The smallest distance is 1.4142135623730951
PS C:\Users\karth> 
```

Time complexity:

F(n)=o(nlogn)