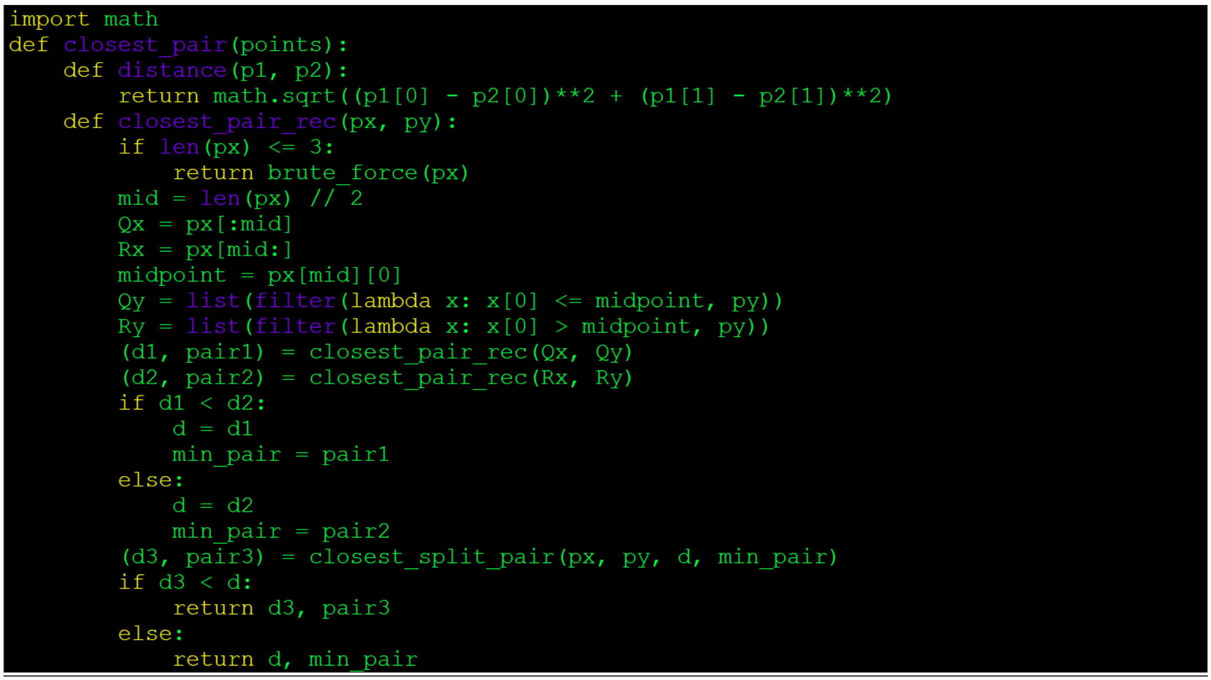
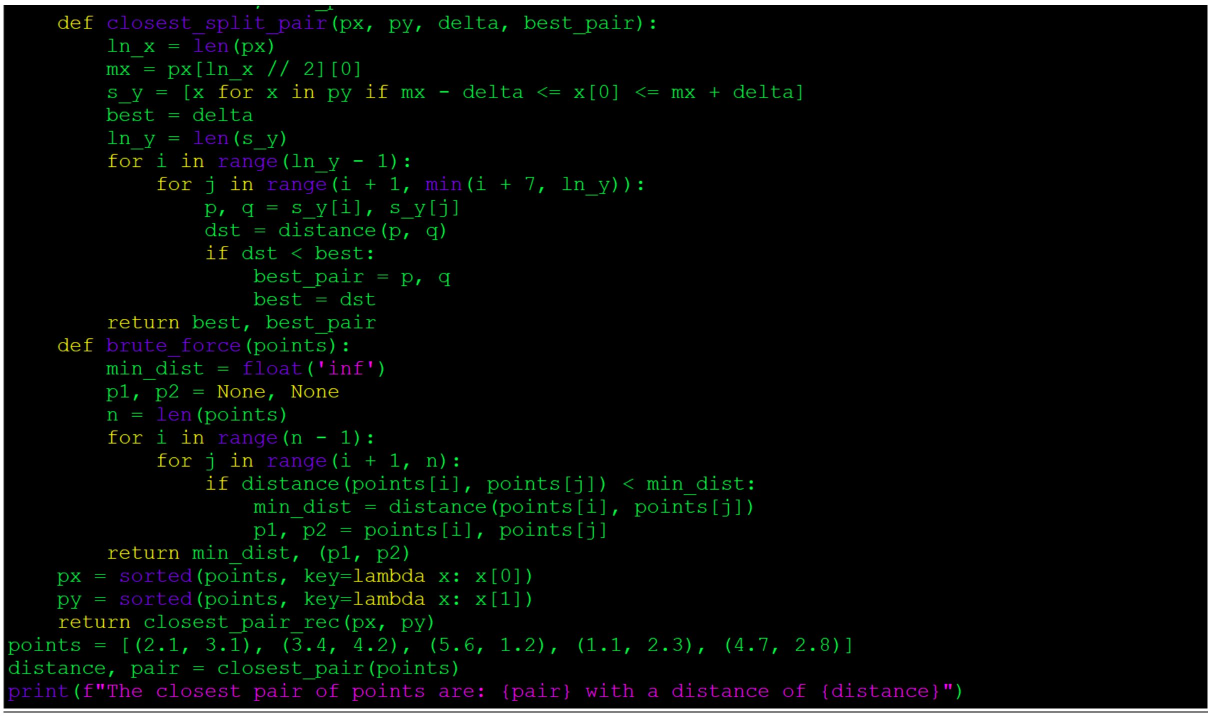
85. Closest pair of points using divide and conquer

Code:

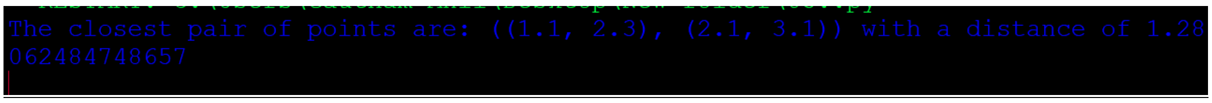


```
import math
def closest_pair(points):
    def distance(p1, p2):
        return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)
    def closest_pair_rec(px, py):
        if len(px) <= 3:
            return brute_force(px)
        mid = len(px) // 2
        Qx = px[:mid]
        Rx = px[mid:]
        midpoint = px[mid][0]
        Qy = list(filter(lambda x: x[0] <= midpoint, py))
        Ry = list(filter(lambda x: x[0] > midpoint, py))
        (d1, pair1) = closest_pair_rec(Qx, Qy)
        (d2, pair2) = closest_pair_rec(Rx, Ry)
        if d1 < d2:
            d = d1
            min_pair = pair1
        else:
            d = d2
            min_pair = pair2
        (d3, pair3) = closest_split_pair(px, py, d, min_pair)
        if d3 < d:
            return d3, pair3
        else:
            return d, min_pair
```



```
    def closest_split_pair(px, py, delta, best_pair):
        ln_x = len(px)
        mx = px[ln_x // 2][0]
        s_y = [x for x in py if mx - delta <= x[0] <= mx + delta]
        best = delta
        ln_y = len(s_y)
        for i in range(ln_y - 1):
            for j in range(i + 1, min(i + 7, ln_y)):
                p, q = s_y[i], s_y[j]
                dst = distance(p, q)
                if dst < best:
                    best_pair = p, q
                    best = dst
        return best, best_pair
    def brute_force(points):
        min_dist = float('inf')
        p1, p2 = None, None
        n = len(points)
        for i in range(n - 1):
            for j in range(i + 1, n):
                if distance(points[i], points[j]) < min_dist:
                    min_dist = distance(points[i], points[j])
                    p1, p2 = points[i], points[j]
        return min_dist, (p1, p2)
    px = sorted(points, key=lambda x: x[0])
    py = sorted(points, key=lambda x: x[1])
    return closest_pair_rec(px, py)
points = [(2.1, 3.1), (3.4, 4.2), (5.6, 1.2), (1.1, 2.3), (4.7, 2.8)]
distance, pair = closest_pair(points)
print(f"The closest pair of points are: {pair} with a distance of {distance}")
```

Output:



```
The closest pair of points are: ((1.1, 2.3), (2.1, 3.1)) with a distance of 1.28
062484748657
```

Time Complexity:

- T(n)= O(nlogn)