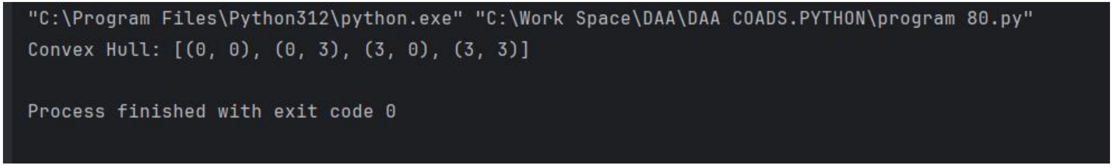
80.Exhaustive Search

Program:

```
def orientation(p, q, r):
    val = (q[1] - p[1]) * (r[0] - q[0]) - (q[0] - p[0]) * (r[1] - q[1])
    if val == 0:
        return 0
    elif val > 0:
        return 1
    else:
        return -1

def convex_hull_brute_force(points):
    n = len(points)
    if n < 3:
        return points

    hull = []

    for i in range(n):
        for j in range(i + 1, n):
            left = False
            right = False
            for k in range(n):
                if k == i or k == j:
                    continue
                orient = orientation(points[i], points[j], points[k])
                if orient > 0:
                    left = True
                elif orient < 0:
                    right = True
                if left and right:
                    break
            if not (left and right):
                if points[i] not in hull:
                    hull.append(points[i])
                if points[j] not in hull:
                    hull.append(points[j])

    # Sorting the points of the convex hull
    hull.sort(key=lambda p: (p[0], p[1]))
    return hull

# Example usage:
points = [(0, 3), (2, 2), (1, 1), (2, 1), (3, 0), (0, 0), (3, 3)]
hull = convex_hull_brute_force(points)
print("Convex Hull:", hull)
```

Output:



```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA\DAA COADS.PYTHON\program 80.py"
Convex Hull: [(0, 0), (0, 3), (3, 0), (3, 3)]

Process finished with exit code 0
```

Time complexity:
O(n^3)