

به نام خدا

استاد: دکتر مهدی کریمی نسب
درس مبانی نظریه گراف

نام: فاطمه زهرا بخشنده
شماره دانشجویی: 98522157

تمرین امتیازی پیدا کردن Convex Hull:

8 نقطه زیر را به عنوان ورودی برنامه می دهیم:

```
inputs = [  
    (2, 2),  
    (0, 5),  
    (1, 1),  
    (5, 5),  
    (0, 0),  
    (1, 2),  
    (6, 1),  
    (3, 3)  
]
```

کد پایتون اجرا شده و رئوس روی پوسته محدب را به ترتیب می نویسد. همچنین معادلات خطوط گذرنده از یال های پوسته محدب را نیز به ترتیب خروجی می دهد:

```
points on convex hull sorted:  
(0, 5)  
(5, 5)  
(6, 1)  
(0, 0)  
  
lines on convex hull:  
y = 5.0  
y = -4.0 x + 25.0  
y = 0.17 x + 0.0  
x = 0.0
```

کد پایتون نیز به شرح زیر است:

```
def find_line(p1, p2):

    a = p2.y - p1.y
    b = p1.x - p2.x
    c = a*(p1.x) + b*(p1.y)

    if a == 0:
        print("y = ", float(c) / b, "\n")
    elif b == 0:
        print("x = ", float(c) / a, "\n")
    else:
        m = round(float(-a) / b, 2)
        t = round(float(c) / b, 2)

        if t < 0:
            print("y =", m, "x -", -t, "\n")
        else:
            print("y =", m, "x +", t, "\n")

def distance(p1, p2):
    return (p1.x - p2.x) ** 2 + (p1.y - p2.y) ** 2

def check_orientation(p, q, r):
    val = (q.y - p.y) * (r.x - q.x) - (q.x - p.x) * (r.y - q.y)
    return 0 if val == 0 else 1 if val > 0 else 2

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

p0 = Point(0, 0)

def compare_points(p1, p2):
    o = check_orientation(p0, p1, p2)
    if o == 0:
        return -1 if distance(p0, p2) >= distance(p0, p1) else 1
    else:
        return -1 if o == 2 else 1

def find_convex_hull(points):
```

```

n = len(points)

ymin = points[0].y
min = 0
for i in range(1, n):
    y = points[i].y

    if ((y < ymin) or
        (ymin == y and points[i].x < points[min].x)):
        ymin = points[i].y
        min = i

points[0], points[min] = points[min], points[0]

p0 = points[0]
from functools import cmp_to_key
points = sorted(points, key=cmp_to_key(compare_points))

m = 1
for i in range(1, n):
    while ((i < n - 1) and
           (check_orientation(p0, points[i], points[i + 1]) == 0)):
        i += 1

    points[m] = points[i]
    m += 1

if m < 3:
    return

stack = []
stack.append(points[0])
stack.append(points[1])
stack.append(points[2])

for i in range(3, m):
    while ((len(stack) > 1) and
           (check_orientation(stack[-2], stack[-1], points[i]) != 2)):
        stack.pop()
    stack.append(points[i])

convex = []
while stack:
    p = stack[-1]
    convex.append(p)

```

```

        print("(" + str(p.x) + ", " + str(p.y) + ")\n")
        stack.pop()

    return convex

if __name__ == "__main__":

    inputs = [
        (2, 2),
        (0, 5),
        (1, 1),
        (5, 5),
        (0, 0),
        (1, 2),
        (6, 1),
        (3, 3)
    ]

    points = [Point(x, y) for (x, y) in inputs]

    print('\npoints on convex hull sorted:\n')
    convex_hull = find_convex_hull(points)

    print('\nlines on convex hull:\n')
    for i in range(len(convex_hull) - 1):
        find_line(convex_hull[i], convex_hull[i + 1])
    find_line(convex_hull[-1], convex_hull[0])

```