**프로그래밍 실습 #9**

201501489 최영진

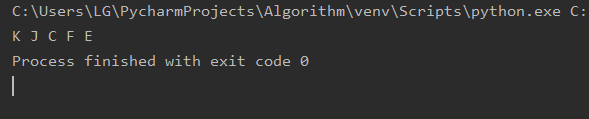
1. 다음 알고리즘을 사용하여 짐꾸리기 프로그램을 파이썬으로 구현하라. 연습

문제 5.3의 점 집합을 사용하여 프로그램을 실행시켜 보라.

1. 코드

import geo as g  
  
def theta(p1, p2):  
 dx = p2.x - p1.x  
 ax = abs(dx)  
 dy = p2.y - p1.y  
 ay = abs(dy)  
 if ax + ay == 0:  
 t = 0  
 else:  
 t = dy / (ax + ay)  
 if dx < 0:  
 t = 2-t  
 elif dy < 0:  
 t = 4 + t  
 return t \* 90  
  
def packageWrapping(p, n) :  
 minIndex = 0  
 for i in range(n):  
 if p[i].y < p[minIndex].y:  
 minIndex = i  
 p[n] = p[minIndex]  
 th = 0.0  
 for m in range(n):  
 p[m], p[minIndex] = p[minIndex], p[m]  
 minIndex = n  
 v = th  
 th = 360.0  
 for i in range(m+1, n+1):  
 if theta(p[m], p[i]) > v:  
 if theta(p[m], p[i]) < th:  
 minIndex = i  
 th = theta(p[m], p[minIndex])  
 if minIndex == n:  
 return m  
  
N = 12  
p = []  
for i in range(N):  
 p.append(g.point(g.x\_value[i], g.y\_value[i], g.c\_value[i]))  
p.append(g.point(None, None, None))  
M=packageWrapping(p, N)  
for i in range(M+1):  
 print(p[i].c, end=' ')

1. 실행화면



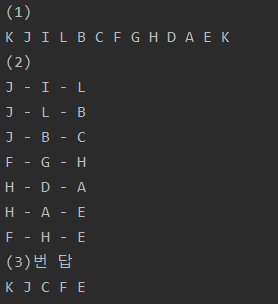
2. 다음 알고리즘을 사용하여 그라함 스캔 프로그램을 파이썬으로 구현하라.

구현된 프로그램을 사용하여 연습문제 5.3을 풀어보라.

1. 코드

import geo as g  
  
def ccw(p0, p1, p2):  
 dx1 = p1.x - p0.x  
 dy1 = p1.y - p0.y  
 dx2 = p2.x - p0.x  
 dy2 = p2.y - p0.y  
 if dx1 \* dy2 > dy1 \* dx2 : return +1  
 if dx1 \* dy2 < dy1 \* dx2 : return -1  
 if dx1 == 0 and dy1 == 0: return 0  
 if (dx1 \* dx2 < 0) or (dy1 \* dy2 < 0): return -1  
 if (dx1 \* dx1 + dy1 \* dy1) < (dx2 \* dx2 + dy2 \* dy2): return +1  
 return 0  
  
def theta(p1, p2):  
 dx = p2.x - p1.x  
 ax = abs(dx)  
 dy = p2.y - p1.y  
 ay = abs(dy)  
 if ax + ay == 0:  
 t = 0  
 else:  
 t = dy / (ax + ay)  
 if dx < 0:  
 t = 2 - t  
 elif dy < 0:  
 t = 4 + t  
 return t \* 90  
  
def selectionSort(p, n):  
 for i in range(1, n):  
 minIndex = i  
 for j in range(i+1, n+1):  
 if theta(p[1], p[j]) < theta(p[1], p[minIndex]):  
 minIndex = j  
 p[minIndex], p[i] = p[i], p[minIndex]  
  
def grahamScan(p, n):  
 minIndex = 1  
 for i in range(2, n+1):  
 if p[i].y < p[minIndex].y:  
 minIndex = i  
 for i in range(1, n+1):  
 if p[i].y == p[minIndex].y:  
 if p[i].x > p[minIndex].x:  
 minIndex = i  
 p[1], p[minIndex] = p[minIndex], p[1]  
 selectionSort(p, n)  
 p[0] = p[n]  
 print('(1)')  
 for i in range(1, n+1):  
 print(p[i].c, end=' ')  
 if i == n:  
 print(p[1].c)  
 m = 3  
 print('(2)')  
 for i in range(4, n+1):  
 while ccw(p[m], p[m-1], p[i]) >= 0:  
 print(p[m-1].c, '-', p[m].c, '-', p[i].c)  
 m -= 1  
 m += 1  
 p[i], p[m] = p[m], p[i]  
 return m  
  
N = 12  
p = []  
p.append(g.point(0, 0, None))  
for i in range(N):  
 p.append(g.point(g.x\_value[i], g.y\_value[i], g.c\_value[i]))  
  
M = grahamScan(p, N)  
print('(3)번 답')  
for i in range(1, M+1):  
 print(p[i].c, end = ' ')

1. 실행화면



3. 다음 알고리즘을 사용하여 최근점 점쌍 찾기 프로그램을 파이썬으로 구현

하라. 구현된 프로그램을 사용하여 연습문제 5.4를 풀어보라.

1. 코드

import geo as g  
import math  
  
class node:  
 def \_\_init\_\_(self):  
 self.p = g.point(max, max, '')  
 self.next = None  
  
def comp(t):  
 if t\_pass == 1:  
 return t.p.x  
 else:  
 return t.p.y  
  
def merge(a, b):  
 c = z  
 while True:  
 if (comp(a) < comp(b)):  
 c.next = a  
 c = a  
 a = a.next  
 else:  
 c.next = b  
 c = b  
 b = b.next  
  
 if c == z:  
 break;  
 c = z.next  
 z.next = z  
 return c  
  
def check(p1, p2):  
 global min, cp1, cp2  
 if (p1.y != z.p.y) and (p2.y != z.p.y):  
 dist = math.sqrt((p1.x - p2.x)\*(p1.x-p2.x) + (p1.y - p2.y) \* (p1.y - p2.y))  
 if dist < min:  
 min = dist  
 cp1 = p1  
 cp2 = p2  
  
def sort(c, N):  
 if c.next == z:  
 return c  
 a = c  
 for i in range(2, (N//2)+1): # 나누기 연산 후 소수점 이하의 수를 버리고, 정수 부분의 수만 구함. 일반 나누기  
 c = c.next  
 b = c.next  
 c.next = z  
 c = merge(sort(a, N//2), sort(b, N-(N//2)))  
  
 if t\_pass == 2:  
 middle = b.p.x  
 p1 = z.p  
 p2 = z.p  
 p3 = z.p  
 p4 = z.p  
 a = c  
 while a != z:  
 if math.fabs(a.p.x - middle) < min:  
 check(a.p, p1)  
 check(a.p, p2)  
 check(a.p, p3)  
 check(a.p, p4)  
 p1 = p2  
 p2 = p3  
 p3 = p4  
 p4 = a.p  
 print('min = %.3f,\t cp1 = %s,\t cp2 = %s,\t p1 = %s,\t p2 = %s,\t p3 = %s,\t p4 = %s\t'  
 % (min, cp1.c, cp2.c, p1.c, p2.c, p3.c, p4.c))  
 print()  
 a = a.next  
 return c  
  
def readlist():  
 p = node()  
 h = p  
 for i in range(N):  
 t = node()  
 t.p.x = g.x\_value[i]  
 t.p.y = g.y\_value[i]  
 t.p.c = g.c\_value[i]  
 p.next = t  
 p = p.next  
 p.next = z  
 return h  
  
N = 8  
max = 1000  
cp1 = g.point(max, max, '')  
cp2 = g.point(max, max, '')  
min = max  
  
z = node()  
z.p.x = max  
z.p.y = max  
z.next = z  
h = node()  
h.next = readlist()  
  
t\_pass = 1  
h.next = sort(h.next, N)  
t\_pass = 2  
h.next = sort(h.next, N)

1. 실행화면

