

Problema Internship - Stefan Razvan-Mihai

Cod problema:

```
def count_rectangles(points):
    diagonale = 0 # variabila unde numaram cate diagonale valide am gasit
    puncteDict = dict() # un dictionar in care punem punctele ca si chei
    # si valorile pe true (folosit pt a accesa in O(1) punctele din lista)
    for i in range(len(points)):
        puncteDict[points[i]] = True #punctele care exista au valoarea true

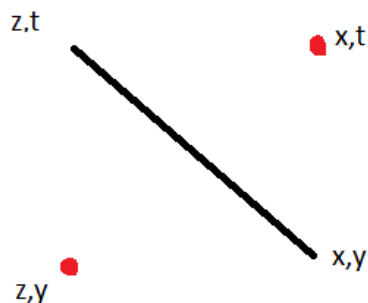
    for i in range(len(points)): #consider i colt dreapta jos
        for j in range(i+1, len(points)): #consider j colt stanga sus, j
# pleaca de la i+1 ca sa nu avem acelasi punct si nici sa nu avem aceleasi 2
# puncte doar ca inversate
            if (points[i][0]!=points[j][0] and points[i][1]!=points[j][1]
# verificam ca linia formata de cele 2 pct sa fie diagonala

# (daca e false conditia aceasta, inseamna ca linia e paralela cu OX sau
# OY)
                and puncteDict.get((points[i][0],points[j][1]))==True
#daca punctul din dreapta sus exista
                and puncteDict.get((points[j][0],points[i][1]))==True): #si
#punctul din stanga jos exista
                    #and points[i]!=points[j] #COMENTEZ aceasta conditie pentru
# ca imi asum ca un punct nu apare de 2 ori in lista
                    diagonale+=1

    return diagonale//2 #din moment ce in variabila diagonale avem nr de
# diagonale valide, iar un dreptunghi are 2 diagonale, impartim la 2
```

Explicatie algoritm:

Algoritmul prezentat parcurge toate perechile distincte de puncte si verifica daca linia formata de acestea poate fi diagonala (daca cele 2 puncte au una dintre cele 2 coordonate egale, inseamna ca linia este paralela cu OX sau OY, deci nu poate fi diagonala pentru ca se cer doar dreptunghiurile paralele cu OX,OY). Daca linia poate fi diagonala, se verifica daca exista punctele necesare formarii unui dreptunghi cu acea diagonala, adica verificam existenta celor 2 puncte desenate cu rosu din figura de mai jos, care vor avea coordonatele dupa cum se vede.



La final, impartim numarul diagonalelor la 2, pentru ca fiecare dreptunghi are 2 diagonale, deci obtinem astfel numarul de dreptunghiuri.

Pentru a verifica existenta punctelor cu rosu in $O(1)$, am construit un dictionar cu elemente de tipul {punct : True}. Fara acest dictionar, ar fi fost nevoie de o noua parcurgere a tuturor punctelor, in cautarea punctului a carui existenta se incearca a fi verificata, fiind ineficient d.p.d.v al timpului de executie.

Solutia are complexitatea in timp $O(n^2)$