
PPPD - Lab. 05

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 05, 2019/2020

Temat: Sad Bożydara

Treść zadania

Rolnik Bożydar ma kolistą sad, w którym drzewka rosną w niektórych punktach o współrzędnych całkowitych. Niestety nikczemnicy kradną mu jabłka, dlatego zatroskany Bożydar postanowił ogrodzić swoje drzewka płotem. Konieczne jest, żeby płot był ładny - ładny płot jest okręgiem o środku w punkcie o współrzędnych całkowitych. Bożydar chciałby wiedzieć, jak długi będzie ten płot i musimy mu w tym pomóc.

Masz daną funkcję:

```
def drzewo(x, y):
    if x >= 0 and x <= 10 and y >= 0 and y <= 10:
        return (x == y) and (x % 10 < 4 or x % 10 > 5)
    elif x >= -10 and x <= -1 and y >= 0 and y <= 10:
        return True
    else:
        return False
```

która odpowiada na pytanie, czy w punkcie o współrzędnych (x, y) znajduje się drzewo.

1. Napisz funkcję `wypisz_kolo(x, y, r)`, która wypisze podane koło w formacie `Srodek (x,y), promien r` (Na przykład: `Srodek (3,4), promien 5`).
2. Wczytaj od użytkownika informacje o położeniu sadu Bożydara: środek koła jako liczby całkowite x i y oraz rzeczywisty promień r . Wypisz ją używając funkcji z poprzedniego podpunktu.
3. Napisz funkcję `najmniejszy_plot(x_sad, y_sad, r_sad, x, y)` która jako pierwsze trzy parametry przyjmuje sad Bożydara i zwróci jako krotkę najmniejsze koło o środku w punkcie (x, y) zawierające wszystkie drzewa w sadzie (przypomnijmy: krotkę konstruujemy poprzez napis `(a,b,c)`). W szczególnym przypadku, kiedy wewnątrz podanego koła nie ma żadnego drzewka, zwróć koło o środku w punkcie $(0,0)$ i promieniu 0. Uwaga: nie wywołuj funkcji `drzewo` dla punktów leżących poza sadem.

Następnie wywołaj tę funkcję dla koła opisującego sad Bożydara oraz punktu (x, y) podanego przez użytkownika. Wypisz wynik używając funkcji `wypisz_kolo`.

Wskazówka: wystarczy znaleźć drzewo leżące najdalej od punktu (x, y) .

Przykład: dla sadu o środku w punkcie $(5,6)$ i promieniu 5 z następującym układem drzew:

```
11      .
10  .....
9   ..D.....
8   .....
7   .....
6  ....D.....
5   .....
4   .....
```

```
3 ..D.....
2 .....
1 .
y/x 012345678910
```

(gdzie D oznacza drzewo, współrzędna x rośnie w prawo i jest w zakresie od 0 do 10, a współrzędna y rośnie w górę i jest w zakresie od 1 do 11) oraz punktu $(x,y)=(3,3)$ odpowiedzią jest koło o środku w punkcie $(3,3)$ i promieniu 6 (najdalsze drzewko od punktu $(3,3)$ jest w odległości 6).

Wskazówka dotycząca krotek:

```
def dzielenie(a, b):
    return a//b, a%b

def main():
    # odbiór wyniku do dwóch zmiennych,
    # tzw. rozpakowywanie krotki
    iloraz, reszta = dzielenie(5,3)
```

UWAGA: rysunek jest poglądowy i nie musi przedstawiać rozkładu drzew zwracanego przez funkcję `drzewo()`.

4. Napisz funkcję `zapisz_do_pliku(sciezka, x, y, r)`, która w pliku wskazanym w parametrze `sciezka` zapisuje powyższy rysunek (czyli rozkład drzew na pełnym sadzie Bożydara wraz z podpisanymi osiami), wraz ze współzrędnymi. Oczywiście musimy powtórzyć bardzo podobne obliczenia jak w funkcji w poprzednim podpunkcie. Osie powinny wyglądać ładnie i się zgrywać z rysunkiem dla jednocyfrowych liczb nieujemnych.
5. Napisz funkcję `obwod_kola(x_min, x_max, y_min, y_max)`, która wyznaczy obwód podanego koła. Używając tej funkcji wyznacz i wypisz obwód sadu Bożydara oraz długość znalezionej płotu. Przybliżenie stałej π znajdziesz w `math.pi`.
6. Napisz funkcję `naprawde_najkrotszy_plot(x, y, r)` która przyjmie koło opisujące wymiary sadu i zwróci koło o najmniejszym obwodzie, którego środek znajduje się wewnątrz sadu Bożydara i które zawiera wszystkie drzewka. Wywołaj tę funkcję w mainie i wypisz znalezione koło.

Przykład: Zauważmy, że w powyższym przykładzie wywołanie funkcji `najmniejszy_plot` dla punktu $(3,6)$ zwróci koło o promieniu 3. Można sprawdzić, że dla każdego innego punktu otrzymamy koło o większym promieniu, a zatem odpowiedzią jest koło o środku w $(3,6)$ i promieniu 3.

Wskazówka: wystarczy dla każdego punktu (x,y) wewnątrz sadu wyznaczyć najmniejsze koło w zawierające wszystkie drzewka o środku w punkcie (x,y) , używając wcześniej napisanej funkcji `najmniejszy_plot`, i wybrać spośród nich koło o najmniejszym promieniu.

7. Napisz własną funkcję `drzewo2(x, y)`, która działa w sposób następujący: jeśli współrzędne wychodzą poza prostokąt $[-10, 10] \times [-10, 10]$, zwraca `False`. Jeśli jesteśmy w prostokącie $[0, 10] \times [-10, 10]$, to zwracamy `True` z prawdopodobieństwem 0.5. Jeśli jesteśmy w prostokącie $[-10, -1] \times [-10, 10]$, to zwracamy `True` z prawdopodobieństwem 0.1.

Uwaga: Program powinien być napisany w taki sposób, aby działał poprawnie po zmianie funkcji `drzewo()` na dowolną inną (tzn. jedyny dopuszczalny sposób sprawdzania, czy w jakimś punkcie znajduje się drzewo, to wywołanie funkcji `drzewo()`).

Przykłady interakcji użytkownika z programem:

```
Podaj wymiary sadu w kolejności x, y, promień:
8
7
7
Podane koło: Środek (8,7), promień 7
```

Podaj współrzędne środka plotu (x,y):

5

8

Najmniejszy plot zawieracy wszystkie drzewka: Środek (5,8), promień 5.38516

Obwód sadu: 43.96

Długość plotu: 33.8188

2

Naprawdę najmniejszy plot zawieracy wszystkie drzewka:

Środek (6,7), promień 5

Punktacja

Za poszczególne etapy można uzyskać następującą liczbę punktów:

- pkt 1,2,3 - 2p
- pkt 4 - 2p
- pkt 5. i 6. - 4p
- pkt 7 - 2p

Uwaga

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2 p.