
PPPD - Lab. 09

Copyright ©2022 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 09, grupa A, 2022/2023, autor: Michał Dębski

Temat: Linie ograniczające widoczność

Treść zadania

Celem zadania jest stworzenie obrazka składającego się z kilku linii między losowymi punktami, a następnie “narysowanie” go w formie tekstowej, zaznaczając znakiem ? wszystkie piksele niewidoczne ze środka obrazka.

Obrazek będziemy reprezentować jako macierz wartości logicznych, gdzie wartość **True** w *i*-tym wierszu i *j*-tej kolumnie oznacza, że piksel o współrzędnych (*i*, *j*) jest zapełniony, a wartość **False** - że jest pusty.

Piksele, które powinny zostać wypełnione przy rysowaniu linii pomiędzy punktami wyznaczamy według następującej zasady. Pierwszym pikselem zawsze jest punkt początkowy. Każdy następny piksel jest sąsiadem poprzedniego (tzn. różni się na dokładnie jednej współrzędnej o dokładnie 1) i jest wybrany tak, aby przybliżyć się do punktu końcowego oraz przekroczyć odcinek łączący punkt początkowy z końcowym lub przybliżyć się do niego.

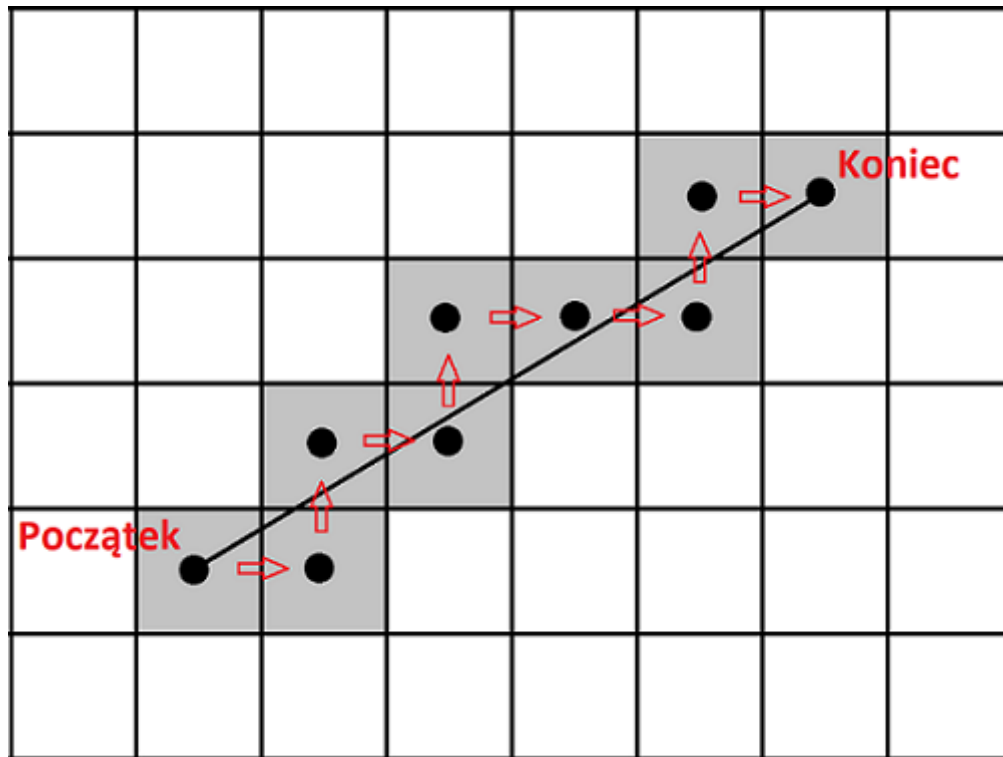
Jeśli punkt końcowy znajduje się na prawo w górę od początkowego, oznacza to, że w każdym kroku sprawdzamy, czy jesteśmy poniżej odcinka i jeśli tak, robimy krok w górę, a w przeciwnym przypadku robimy krok w prawo; patrz rysunek poniżej. Możesz wykorzystać kod funkcji znajdujące współrzędne punktów na linii w tym przypadku (jest on podany we wskazówce poniżej). Pozostałe trzy przypadki są symetryczne; z nimi musisz poradzić sobie samodzielnie.

Dla ustalonego obrazka (reprezentowanego jako macierz wartości logicznych) mówimy że punkt (*x2*, *y2*) jest widoczny z punktu (*x1*, *y1*), jeśli wszystkie piksele na linii od (*x1*, *y1*) do (*x2*, *y2*), nie licząc pierwszego i ostatniego, są puste.

Program powinien zawierać następujące elementy:

- Funkcja `narysuj_linie`, która jako argumenty przyjmie macierz wartości logicznych oraz cztery liczby całkowite *sx*, *sy*, *kx*, *ky* oznaczające współrzędne początku i końca linii, a następnie narysuje w macierzy linię między tymi punktami (tzn. ustawi wartości odpowiednich komórek na **True**)
- Funkcja `macierz_widoczności`, która jako argument przyjmuje macierz wartości logicznych i zwraca macierz o tym samym rozmiarze, wypełnioną znakami #, , ? oraz X. Znakiem X należy zaznaczyć środek, znak ? powinien oznaczać piksel, który jest niewidoczny z punktu środkowego, a wszystkie pozostałe piksele oznaczamy jako # jeśli są pełne oraz w przeciwnym przypadku.
- Funkcja `main`, która wczyta od użytkownika wymiary obrazka oraz liczbę linii do narysowania, a następnie narysuje zadaną liczbę linii posługując się funkcją `narysuj_linie`, przyjmując losowe współrzędne początków i końców, a na koniec wyznaczy i wypisze macierz widoczności. Możesz posłużyć się poniższymi funkcjami.

```
def wypisz_macierz_logiczna(M):  
    for j in range(len(M[0]) + 2):  
        print("-", end="")  
  
    print()
```



Rysunek 1: Kolejne punkty na linii

```

for i in range(len(M) - 1, -1, -1):
    print("|", end="")
    for j in range(len(M[i])):
        if M[i][j]:
            print("#", end="")
        else:
            print(" ", end="")
    print("|")

for j in range(len(M[0]) + 2):
    print("-", end="")
print()

def wypisz_macierz_znakow(M):
    for j in range(len(M[0]) + 2):
        print("-", end="")

    print()

    for i in range(len(M) - 1, -1, -1):
        print("|", end="")
        for j in range(len(M[i])):
            print(M[i][j], end="")
        print("|")

    for j in range(len(M[0]) + 2):

```

```
print("-", end="")
print()
```

Wskazówki

Możesz wykorzystać poniższą funkcję, która wyznacza współrzędne punktów na linii w przypadku, kiedy punkt końcowy ma obie współrzędne nie mniejsze niż punkt początkowy.

Funkcja jako argumenty przyjmuje dwie krotki zawierające współrzędne początku i końca linii i zwraca listę krotek zawierających współrzędne kolejnych punktów na linii.

```
def linia(start, koniec):
    start_x, start_y = start
    koniec_x, koniec_y = koniec

    if start_x > koniec_x or start_y > koniec_y:
        raise ValueError('Niepoprawne dane')

    px, py = start_x, start_y

    wynik = [None] * (koniec_x - start_x + koniec_y - start_y + 1)
    wynik[0] = (px, py)
    pozycja = 1

    while px != koniec_x or py != koniec_y:
        if (py - start_y) * (koniec_x - start_x) > (koniec_y - start_y) * (px - start_x)
            or py == koniec_y:
            px += 1
        else:
            py += 1
        wynik[pozycja] = (px, py)
        pozycja += 1

    return wynik

# przykład użycia:
# wywołanie linia((1, 3), (2, 5)) zwróci listę [(1, 3), (1, 4), (2, 4), (2, 5)]
```

Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Funkcja narysuj_linie - 3 punkty
- Funkcja macierz_widoczności - 4 punkty
- Funkcja main - 3 punkty

Uwagi

- 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 2p

Przykład interakcji użytkownika z programem

Podaj wysokość: 20

```

      ###
     ####
    #####
   #####
  #####
 #####
#####
#####
#####
#####
#####

```

[illegible][illegible]

```

##### ###
      ###   ##
##          #####    ##
####         #####    ###
     #####        ##
           #####       ##
            #####      ##
             #####     ##
              #####    ##
               #####   ##
                #####  ##
                 #####
                  #####
                   ##
                    ##
                     ##
                      ##
                       ##

```

[illegible]

Process finished with exit code 0