

Metody Monte Carlo

Laboratorium 1

Zadanie 1

Kod

```
import matplotlib.pyplot as plt

def main():
    # Input values
    m = 2 ** 13 - 1
    a1 = 17
    a2 = 85
    c = 0
    iterations = 2500

    values_a1 = [1]
    for index in range(0, iterations):
        values_a1.append(generator(values_a1[index], m, a1, c))

    values_a2 = [1]
    for index in range(0, iterations):
        values_a2.append(generator(values_a2[index], m, a2, c))

    show(list(chunks(values_a1)), a1)
    show(list(chunks_skip(values_a1)), a1)

    show(list(chunks(values_a2)), a2)
    show(list(chunks_skip(values_a2)), a2)

def generator(input_value, m, a, c):
    return (a * input_value + c) % m

def show(pairs, a):
    x = [x[0] for x in pairs]
    y = [x[-1] for x in pairs]

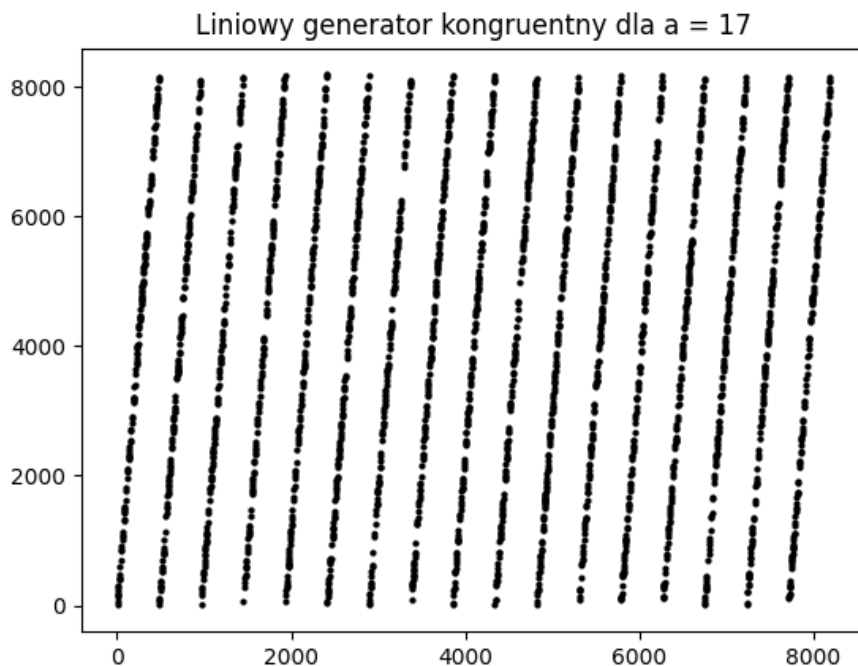
    print(x)
    print(y)
    plt.plot(x, y, 'ko', markersize=2)
    plt.title(f'Liniowy generator kongruentny dla a = {a}')
    plt.show()

def chunks(lst, n=2):
    for i in range(0, len(lst)):
        yield lst[i:(i+n)]

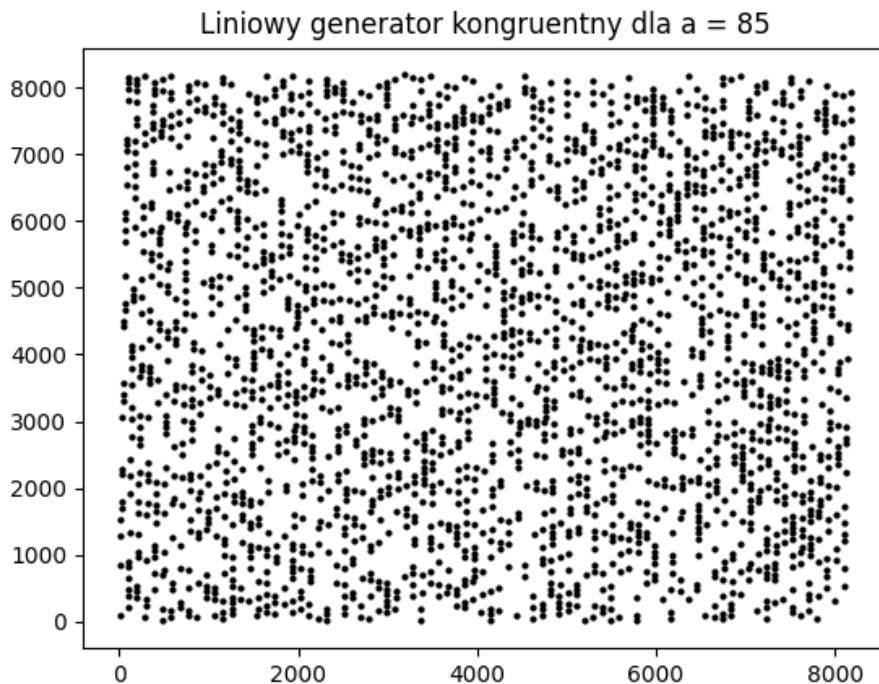
def chunks_skip(lst, n=2):
    for i in range(0, len(lst) - n):
        yield [lst[i], lst[i + n]]

if __name__ == '__main__':
    main()
```

Wykres XY dla próby $a = 17$ i par (x_i, x_{i+1})

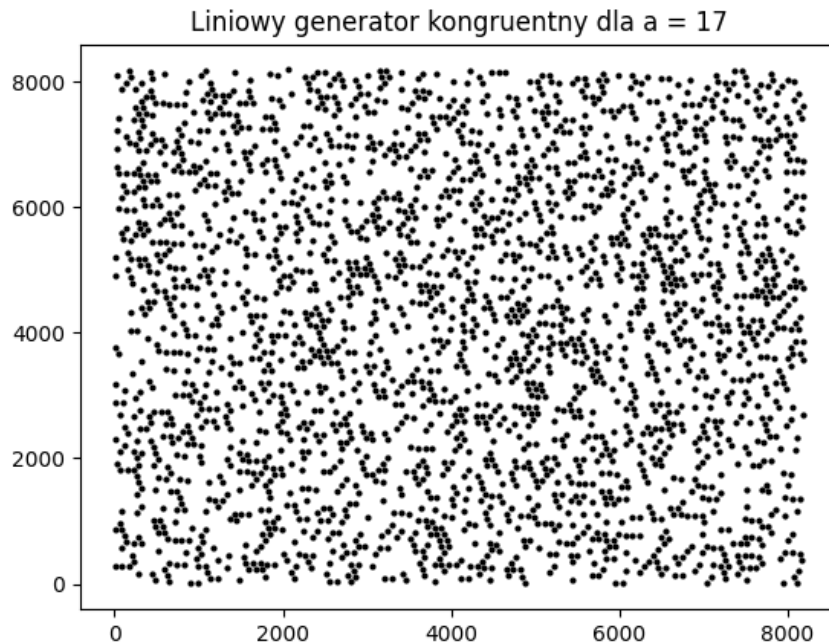


Wykres XY dla próby $a = 85$ i par (x_i, x_{i+1})

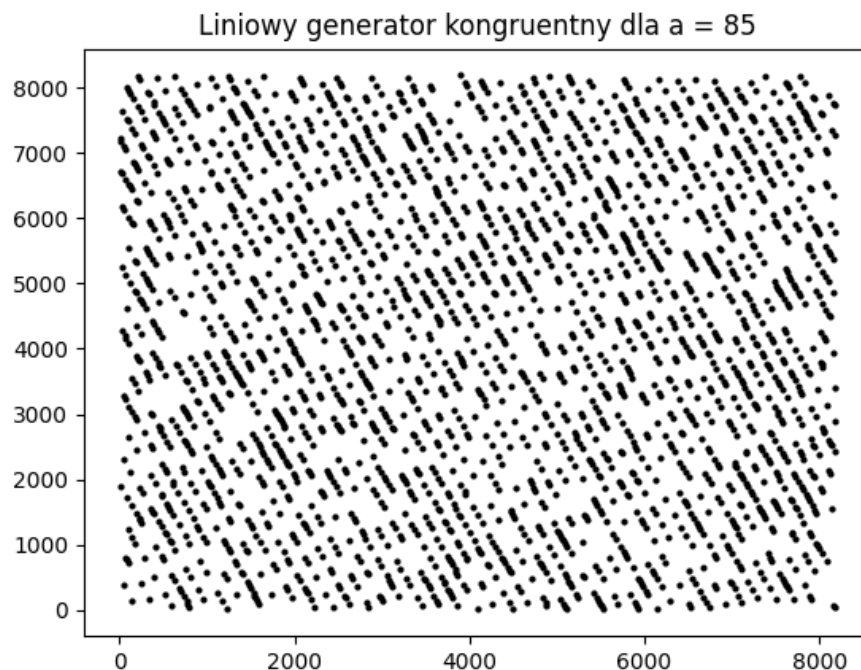


Wykres dla $a = 85$ jest lepszy ponieważ punkty są wyraźnie bardziej rozproszone od siebie (średnia odległość między dwoma punktami jest większa niż dla $a = 17$)

Wykres XY dla próby $a = 17$ i par (x_i, x_{i+2})



Wykres XY dla próby $a = 85$ i par (x_i, x_{i+2})



Dla $a = 85$ wyniki są wyraźnie gorsze niż dla przypadku $a = 17$ gdy próbujemy punkty z par co drugich liczb. Oznacza to, że nie zawsze większy współczynnik a daje lepsze rezultaty, szczególnie w kontekście tego algorytmu. Ten algorytm wymaga odpowiedniego kalibrowania parametru a oraz sposobu doboru próbek aby wykazywał jak najmniejszą korelację.