Để tạo số giả ngẫu nhiên, chúng ta có thể sử dụng một số thuật toán khác nhau. Trong ví dụ này, chúng ta sẽ sử dụng thuật toán Linear Congruential Generator (LCG).

Thuật toán LCG sử dụng một phương trình đệ quy để tạo ra các số giả ngẫu nhiên.

Nó được định nghĩa như sau:

Xn+1 = (aXn + c) mod m

Trong đó:- Xn là số nguyên dương tại vòng lặp thứ n

-a, c, và m là các số nguyên dương

- mod là toán tử lấy phần dư.

Chuỗi số giả ngẫu nhiên được tạo ra bằng cách sử dụng giá trị đầu vào X0 và phương trình đệ quy trên.

Để kiểm định tính ngẫu nhiên của chuỗi số đầu ra, chúng ta có thể sử dụng một số phương pháp. Trong ví dụ này, chúng ta sẽ sử dụng phương pháp Kiểm định Kolmogorov-Smirnov (KS).

Phương pháp KS kiểm tra tính đồng đều của phân bố xác suất của một mẫu dữ liệu so với một phân bố xác suất cho trước. Để thực hiện kiểm định KS, chúng ta so sánh chuỗi số đầu ra với một phân bố xác suất đồng đều, và tính toán khoảng cách giữa các phân phối tích lũy của chuỗi số đầu ra và phân bố xác suất đồng đều.

Nếu khoảng cách này nhỏ hơn một ngưỡng xác định, chúng ta có thể kết luận rằng chuỗi số đầu ra có tính ngẫu nhiên.

File tạo chuỗi số:

class RandomNumberGenerator:

def \_\_init\_\_(self, seed, a=1103515245, c=12345, m=2\*\*31-1):

self.seed = seed

self.a = a

self.c = c

self.m = m

def generate(self, num\_random\_numbers):

random\_numbers = []

x = self.seed

for i in range(num\_random\_numbers):

x = (self.a\*x + self.c) % self.m

random\_numbers.append(x/self.m)

return random\_numbers

File kiểm thử

import numpy as np

from scipy.stats import ks\_2samp

# Tạo chuỗi số

rng = RandomNumberGenerator(seed=12345)

random\_numbers = rng.generate(10000)