Tarea 1.

Fecha de entrega: Jueves 19 de febrero

1. Sea $x_n \sim U\{1, 2, .., n-1\}$ y $U \sim U(0, 1)$ demuestre que

$$\frac{x_n}{n} \xrightarrow{d} U.$$

2. Sean $x_i \sim U\{0, 1, ..., 9\}$ con $i \in \{1, 2, ...\}$ independenites y definimos $Y_n = 0.x_1x_2...x_n$. ¿Es cierto que $Y_n \xrightarrow{d} U$?, justifique su respuesta.

3. Implemente el algoritmo de generador congruencial lineal dado por la recursión

$$x_{n+1} = (ax_n + b) \bmod m$$

   - Usando $x_0 = 1$ analice que pasa en los siguientes casos :

$$\begin{aligned} a &= 5 & b &= 0 & m &= 11 \\ a &= 5 & b &= 0 & m &= 1000 \\ a &= 781 & b &= 387 & m &= 1000 \end{aligned}$$

   ¿Son de rango máximo?. Una forma de ver que tan aleatorio es el generador es haciendo un gráfico de dispersión (x_i, x_{i+1}), ¿Qué puede decir de los gráficos para los 3 casos?

   - Usando $x_0 = 1$ analice que pasa en los siguientes casos :

$$\begin{aligned} a &= 2^{34} + 1 & b &= 1 & m &= 2^{35} \\ a &= 2^{18} + 1 & b &= 1 & m &= 2^{35} \end{aligned}$$

   ¿Cuál es el rango en cada caso?, ¿Cuál de los dos casos usaría y por qué?

4. Implemente y analice el algoritmo de Blum-Blum-Shub, usando $m = pq$ con

$$\begin{aligned} p &= 1267650600228229401496703981519 \\ q &= 1267650600228229401496704318359 \end{aligned}$$

5. Implemente y analice el algoritmo Wichman-Hill

$$\begin{aligned} x_t &= 171x_{t-1} \bmod 30269 \\ y_t &= 172y_{t-1} \bmod 30307 \\ z_t &= 170z_{t-1} \bmod 30323 \end{aligned}$$

$$U_t = \frac{x_t}{30269} + \frac{y_t}{30307} + \frac{z_t}{30323} \bmod 1$$

6. (4 puntos) Una de las versiones de Mersenne más populares es la conocida como MT19937, ya que esta versión es la basada en el número de Mersenne $2^{19937}-1$. Existen dos variantes importantes del algoritmo que son las de 32 y 64 bits, que deberán ser implementadas en esta tarea.

   Implemente y analice MT19937 de 32 bits con parámetros

$$
\begin{aligned}
(w, n, m, r) &= (32, 624, 397, 31) \\
a &= 9908B0DF_{16} \\
u &= 11 \\
s &= 7 \\
t &= 15 \\
l &= 18 \\
b &= 9D2C5680_{16} \\
c &= EFC60000_{16}
\end{aligned}
$$

   Si está interesado en una discusión más profunda respecto a la selección de parámetros puede consultar: Jagannatam, A. (2008). Mersenne Twister A Pseudo Random Number Generator and its Variants. George Mason University, Department of Electrical and Computer Engineering.

7. Implemente y analice un generador recursivo múltiple. Recuerde que las mejores opciones para tener la menor cantidad de operaciones es cuando se utilizan polinomios primitivos; para algunos ejemplos en base 2 puede consultar:
   https://maths-people.anu.edu.au/˜brent/trinom.html .
   Mencione el rango del generador que uso (debe ser por lo menos de rango $2^{100}-1$).

8. Implemente la prueba ”Birthday spacing test” y apliquela a los algoritmos que utilizó en el ejercicio 5 y para numpy random unfiorm, usando un año de 2^{32} días y tomando 2^{12} cumpleaños. Como sugerencia empiece probando runif().

9. Implemente la prueba ”3D sphere test” y aplíquela a los algoritmos que utilizó en el ejercicio 8 y para numpy random uniform. Para la parte de la prueba de bondad de ajuste use una muestra de al menos 100 datos. Como sugerencia empiece probando numpy random uniform.

10. (2 puntos) Uno de los resultados más populares en el área de matrices aleatorias y de probabilidad libre es la distribución de Marchenko-Pastur.

    Si X denota una matriz de $m \times n$ con entradas v.a.i.i.d. con media 0 y varianza σ^2 finita y definimos

$$Y_n = \frac{1}{n} X X^T$$

    y sean λ_1,...,λ_m los valores propios de Y_n. Considere la función de distribución para los valores propios:

$$F_m(x) = \frac{1}{m} |\{\lambda_i \text{ con } i \text{ en } \{1, , , m\} | \lambda_i \leq x\}|$$

    El teorema de Marchenko-Pastur dice que si $m, n \to \infty$ tales que el radio $\frac{m}{n} \to \lambda \in (0, \infty)$ entonces $F_m \to F$ donde

$$F(x) = \begin{cases} (1 - \frac{1}{\lambda})\mathbb{1}_{0 \leq x} + \int_{-\infty}^{x} d\nu(x), & \text{si } \lambda > 1 \\ \int_{-\infty}^{x} d\nu(x), & \text{si } 0 \leq \lambda \leq 1, \end{cases}$$

$$d\nu(x) = \frac{1}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{\lambda x} \mathbf{1}_{x \in [\lambda_-, \lambda_+]}\, dx$$

con

$$\lambda_\pm = \sigma^2(1 \pm \sqrt{\lambda})^2.$$

Implemente y analice el algoritmo de transformada inversa para F para los casos con $\lambda > 1$ y $\lambda < 1$.