Universidad del Valle de Guatemala
Guatemala agosto de 2020

Mini proyecto 1

Ejercicio 1 - El triángulo de Shierpinski:

El triángulo de Shierpinski es uno de los más famosos fractales. Un fractal se puede construir a través de saltos aleatorios entre funciones determinísticas (juegos de caos). Considere las siguientes tres funciones:

```
Point f1(Point p): return Point(p.x/2, p.y/2)
Point f2(Point p): return Point(p.x/2 + 0.5, p.y/2)
Point f3(Point p): return Point(p.x/2 + 0.25, p.y/2 + 0.5)
```

Estas funciones toman como parámetro el punto anterior en el fractal, y lo utilizan para generar el siguiente punto (paso determinístico).

La idea de los juegos del caos es simular una variable aleatoria **X**, para elegir la función a utilizar para calcular el siguiente punto. Noten que **X** debe ser discreta, por tanto:

P(X = x1) = **p1**, P(X = x2) = **p2**, P(X = x3) = **p3**, tal que: **p1 + p2 + p3 = 1**

Tasks:

1. Cree un programa que simule 100000 veces X para elegir entre **f1**, **f2** y **f3**, dibuje un triángulo de Sherpinski.
2. Determine experimentalmente **p1, p2, p3** que hacen su dibujo más denso.

Ejercicio 2 - El helecho de Barnsley:

En general los juegos del caos pueden definirse como un Framework (F, X, n), donde **F** es un conjunto de **k** funciones determinísticas, y **X** es una variable aleatoria discreta con distribución probabilística: **P** = {pi | P(X = i) = pi}, y **n** es la cantidad de puntos a dibujar. Noten que |P| = k.

Para este fractal en específico, considere:

F =

```
f1(x, y) → (x*0.85 + y*0.04 + 0.0, x*-0.04 + y*0.85 + 1.6)
f2(x, y) → (-0.15*x + 0.28*y + 0.0, x*0.26 + y*0.24 + 0.44)
f3(x, y) → (x*0.2 + y*-0.26 + 0.0, x*0.23 + y*0.22 + 1.6)
f4(x, y) → (x*0.0 + y*0.0, x*0.0 + y*0.16)
```

P = {0.85, 0.07, 0.07, 0.01}

n = 100000

Tasks:

1. Cree un programa que corra el anterior juego de caos y muestre el dibujo resultante.

Ejercicio 3 – Análisis de pseudorandoms:

Considere las siguientes dos funciones generadoras de pseudorandoms:

Generador 1:

$$x_n = 5^5 x_{n-1} \bmod(2^{35} - 1)$$

Generador 2:

$$x_n = 7^5 x_{n-1} \bmod(2^{31} - 1)$$

Y considere al generador de números aleatorios uniformes en (0, 1) default de su lenguaje de programación de elección, como un tercer generador de números aleatorios:

Generador 3: *(baseline)*

$$x_n = Math.random()$$

Tasks:

1. Construya un programa que compare estos tres generadores a través de un histograma de asteriscos (de 0 a 1 con saltos de 0.1). Use tres comparaciones, para 100, 5000 y 100000 repeticiones.
2. ¿Qué generador le parece mejor? (considere solamente Generador 1 y Generador 2)

Ejemplo: *(10000 corridas del Generador 2)*

```
0.0-0.1: ******************************************** (1001, 10.01%)
0.1-0.2: ********************************************* (1009, 10.09%)
0.2-0.3: ******************************************** (1004, 10.04%)
0.3-0.4: ******************************************** (985, 9.85%)
0.4-0.5: ******************************************* (981, 9.81%)
0.5-0.6: ******************************************* (970, 9.7%)
0.6-0.7: ******************************************** (992, 9.92%)
0.7-0.8: ********************************************** (1030, 10.3%)
0.8-0.9: ******************************************** (1002, 10.02%)
0.9-1.0: ********************************************* (1026, 10.26%)
```

Ejercicio 4 – integral unidimensional:

Considere la siguiente integral:

$$\theta = \int_{-\infty}^{\infty} e^{-x^2}\, dx = 1.772453850905516 02$$

Tasks:

1. Transforme la integral a una con límites de 0 a 1, muestre su procedimiento.

2. Aproxime la integral usando el método de Montecarlo, haciendo 100, 10000 y 1000000 iteraciones.

Ejercicio 5 – integral bidimensional:
Considere la siguiente integral:

$$\theta = \int_{0}^{\infty}\int_{0}^{x} e^{-(x+y)}\,dy\,dx = 0.5$$

Tasks:

1. Transforme la integral múltiple a una en la que ambos límites sean de 0 a 1, muestre su procedimiento..
2. Aproxime la integral usando el método de Montecarlo, haciendo 100, 10000 y 1000000 iteraciones.