

Práctica 2. “Algoritmos probabilísticos y aleatorios”

Benítez Morales Manuel Emilio

Téllez Pérez Juan Manuel

I. ALGORITMO DE MONTECARLO

Este algoritmo se basa en el tiempo de ejecución que se deja, es decir, en tre más veces se ejecute, mejor será el resultado obtenido.

A. Cálculo de π

Ejecutando el código proporcionado del cálculo de π

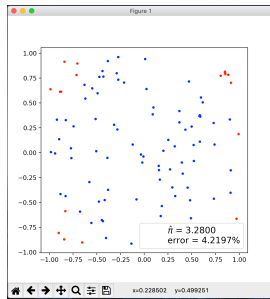


Fig. 1. Ejecución con $N = 100$

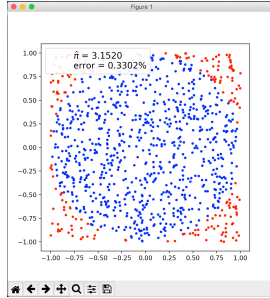


Fig. 2. Ejecución con $N = 1,000$

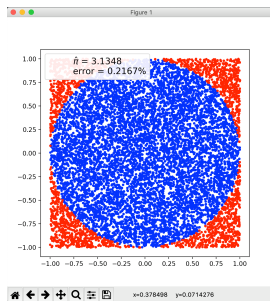


Fig. 3. Ejecución con $N = 10,000$

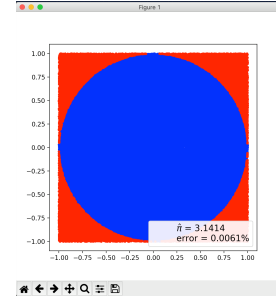


Fig. 4. Ejecución con $N = 100,000$

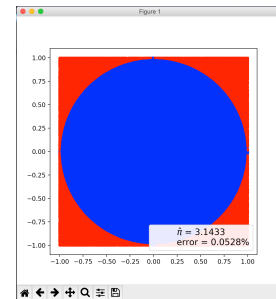


Fig. 5. Ejecución con $N = 1,000,000$

B. Aproximación de integrales

Por medio de un lenguaje diferente, se hizo la aproximación de las integrales:

$$f_1(x) = \int_0^1 (1 - x^2)^{3/2} \cdot dx \quad (1)$$

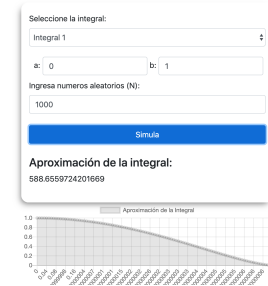


Fig. 6. Aproximación de Integral 1

$$f_2(x) = \int_{-1}^1 e^{x+x^2} \cdot dx \quad (2)$$

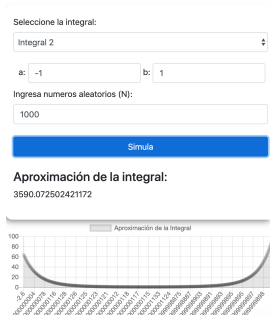


Fig. 7. Aproximación de Integral 2

$$f_3(x) = \int_0^2 (1+x^2)^2 \cdot dx \quad (3)$$

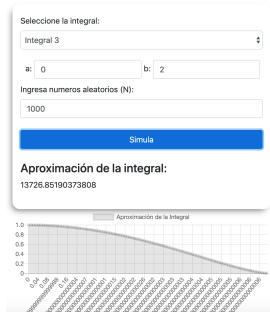


Fig. 8. Aproximación de Integral 3

$$f_4(x) = \int_0^{2\pi} \frac{1}{\cos(x) + 2} \cdot dx \quad (4)$$

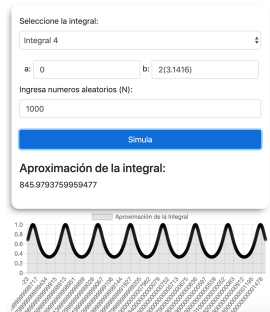


Fig. 9. Aproximación de Integral 4

$$f_5(x) = \int_0^2 \log(x) \cdot dx \quad (5)$$

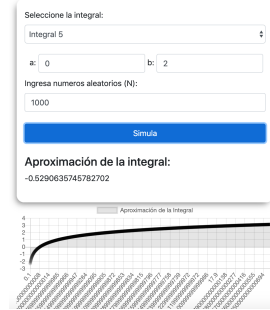


Fig. 10. Aproximación de Integral 5

II. ALGORITMO DE LAS VEGAS

A. Modificación de Quick-Sort

De acuerdo con las ejecuciones realizadas, se puede observar que con la modificación para el algoritmo de las Vegas, se vuelve un poco más lento el tiempo de ejecución, tal como se observa en la siguiente tabla de resultados.

Al ser un algoritmo de aleatoriedad, se tiene la desventaja que para casos no tan grandes, el algoritmo Las Vegas no es tan óptimo para solucionar un ordenamiento, pero a medida que vayamos incrementando la cantidad de números en el arreglo, se podrá visualizar que tendrá un mayor rendimiento.

n	QS normal(ms)	QS las vegas(ms)
1 000	2.208232879638672	3.2279491424560547
2 000	5.138874053955078	8.321762084960938
3 000	9.265899658203125	11.788129806518555
4 000	12.62283325195312	18.229007720947266
5 000	17.19307899475097	27.52995491027832
6 000	24.30105209350586	28.820037841796875
7 000	30.980825424194336	37.85085678100586
8 000	37.58096694946289	44.69609260559082
9 000	40.9998893737793	51.50413513183594
10 000	53.26223373413086	62.72602081298828

TABLE I

CASOS PARA EL PROBLEMA DE LAS 8 REINAS

B. Problema de las 8 reinas

La ejecución se realizó con las reinas definidas especificadas posteriormente, el porcentaje de error en las Vegas no mejora conforme más veces se ejecuta, a diferencia del algoritmo Montecarlo, por lo que no se espera un resultado de solución al problema todo el tiempo, en este caso, se consideró la opción de "No se sabe".

Coordenadas Definidas para las reinas en cada caso:

1) 3 definidas:

- a) [1, 2]
- b) [3, 6]
- c) [5, 3]

2) 4 definidas:

- a) [1, 2]
- b) [3, 6]
- c) [5, 3]
- d) [8, 5]

3) 5 definidas:

- a) [4, 8]
- b) [6, 1]
- c) [2, 4]
- d) [8, 5]
- e) [3, 6]

4) 8 definidas:

- a) [4, 8]
- b) [6, 1]
- c) [2, 4]
- d) [8, 5]
- e) [3, 6]
- f) [1, 2]
- g) [7, 7]
- h) [5, 3]

III. CONCLUSIÓN

Los algoritmos aleatorios son soluciones óptimas para muchos problemas, sin embargo hay que saber cómo y donde implementarlos, puesto que no siempre serán efectivos totalmente.

Así mismo, es importante diferenciar cuando un algoritmo aleatorio depende del tiempo que se ejecute o no, y principalmente en esta práctica se aprendió a diferenciar cada algoritmo y comprender su función y comportamiento aplicado a diferentes problemas.

TABLE II
CASOS PARA EL PROBLEMA DE LAS 8 REINAS

soluciones aleatorias	Reinas aleatorias			
	8	5	4	3
10 000	100%	0%	0%	0%
50 000	100%	0%	0%	0%
100 000	100%	0%	0%	0%
1 000 000	100%	0%	0%	0%