



Ingeniería en Sistemas Computacionales
PROGRAMACION ORIENTADA A OBJETOS
Oscar David Galvan Alvarez
TUE0096

PRACTICA#3 DECIMAL/BINARIO, SIMULACION MONTECARLO

SIMULACION MONTECARLO

```
##PASO1:
##-grafico x,y variable
##-circunferencia con variable radio
import matplotlib.pyplot as plt
import numpy as np
from random import randint, uniform
from math import sqrt
import time

cuadro=5000
radio=cuadro/2
assert radio<= (cuadro/2)
centro=cuadro/2

##### trazar circulo
theta= np.linspace(0,2* np.pi,100)
circ_x=centro + radio* np.cos(theta)
circ_y=centro + radio* np.sin(theta)

inicio=time.time()
n=30000
puntos_x=[round(uniform(0,cuadro),2) for i in range(0,n)]
puntos_y=[round(uniform(0,cuadro),2) for i in range(0,n)]

print(puntos_x)

x_in, y_in=[],[]
x_out,y_out=[],[]
for i in range(0,n):
    distancia=sqrt((puntos_x[i]-centro)**2+(puntos_y[i]-centro)**2)
    if distancia <= radio:
        x_in.append(puntos_x[i])
        y_in.append(puntos_y[i])
    else:
        x_out.append(puntos_x[i])
        y_out.append(puntos_y[i])
final=time.time()
print(round((final-inicio),4), 'segundos')

PI=4*(len(x_in)/n)

cuadro_x,cuadro_y=[0,0,cuadro,cuadro,0],[0,cuadro,cuadro,0,0]
plt.plot(cuadro_x,cuadro_y)
plt.plot(circ_x,circ_y, label=("circunferencia"))
plt.scatter(x_in,y_in,color="red", s=1)
plt.scatter(x_out,y_out,color="yellow",s=2)
plt.text(centro/2, centro,f'PI={PI}',size=14)
plt.gca().set_aspect('equal')
plt.xlabel('ejex')
plt.ylabel('ejey')
plt.title('Valor de pi')
plt.show()
```

En el código de Montecarlo solo explicare el como funciona ya que este se elaboro en clase y se mostró su funcionamiento.

¿Cuáles son los puntos importantes del código?

Creo que sería bueno dividirlo en 5 partes para analizar cada una de ellas:

```
import matplotlib.pyplot as plt
import numpy as np
from random import randint, uniform
from math import sqrt
import time
```

Primero importaremos las librerías necesarias, así como también las funciones correspondientes ya que sin las mismas nuestro código no funcionaría.

Ahora bien, iniciemos con nuestro código:

```
cuadro=5000
radio=cuadro/2
assert radio<= (cuadro/2)
centro=cuadro/2

##### trazar circulo
theta= np.linspace(0,2* np.pi,100)
circ_x=centro + radio* np.cos(theta)
circ_y=centro + radio* np.sin(theta)
```

En esta parte del código haremos las fórmulas correspondientes a nuestro cuadrado, como también las formulas para nuestro circulo. En el caso del cuadrado obtendremos su radio el cual simplemente es la mitad de cualquiera de sus lados, así como también el centro.

Ahora bien, estas no son formulas como tal de un cuadrado ya que sabemos que son de un círculo y así es utilizaremos esos rangos para obtener nuestras coordenadas de nuestro circulo tanto en x como en y usando la formula del ángulo obtenido theta llamada la librería numpy para después realizar el cálculo de las coordenadas.

Ahora obtendremos las coordenadas de los puntos que rellenaran nuestro circulo y nuestro cuadrado:

```
inicio=time.time()
n=30000
puntos_x=[round(uniform(0,cuadro),2) for i in range(0,n)]
puntos_y=[round(uniform(0,cuadro),2) for i in range(0,n)]
```

Ahora bien, le daremos valores a nuestros puntos en este caso

redondeando un numero random uniformes entre 0 y la distancia de nuestro cuadro dando saltos de 2 en 2. Haciendo esto en un ciclo que itere en un rango de 0 a n que es nuestra cantidad de puntos. Ahora bien, porque se ocupa puntos_x y puntos_y y es muy sencillo debido a que graficaremos en un plano cartesiano ocupamos coordenadas en x y y así obteniéndolas de forma aleatoria.

Después de ello se crearán listas vacías para almacenar las coordenadas de

```
x_in, y_in=[],[]
x_out,y_out=[],[]
for i in range(0,n):
    distancia=sqrt((puntos_x[i]-centro)**2+(puntos_y[i]-centro)**2)
    if distancia <= radio:
        x_in.append(puntos_x[i])
        y_in.append(puntos_y[i])
    else:
        x_out.append(puntos_x[i])
        y_out.append(puntos_y[i])
```

nuestros puntos con un ciclo for anidado con un if/else que lo que hace es que almacena según la distancia que esta

del centro del círculo si están dentro del límite del círculo o están del lado del cuadrado.

Ahora la parte mas importante de nuestro código el graficar cada una de las partes

```
PI=4*(len(x_in)/n)

cuadro_x,cuadro_y=[0,0,cuadro,cuadro,0],[0,cuadro,cuadro,0,0]
plt.plot(cuadro_x,cuadro_y)
plt.plot(circ_x,circ_y,label="circunferencia")
plt.scatter(x_in,y_in,color="red",s=1)
plt.scatter(x_out,y_out,color="yellow",s=2)
plt.text(centro/2,centro,f'PI={PI}',size=14)
plt.gca().set_aspect('equal')
plt.xlabel('ejex')
plt.ylabel('ejey')
plt.title('Valor de pi')
plt.show()
```

que usamos en el plano cartesiano.

Primero dando valores de el cuadro en x y el cuadro en y indicando los rangos en los que se graficara con un plt.plot.

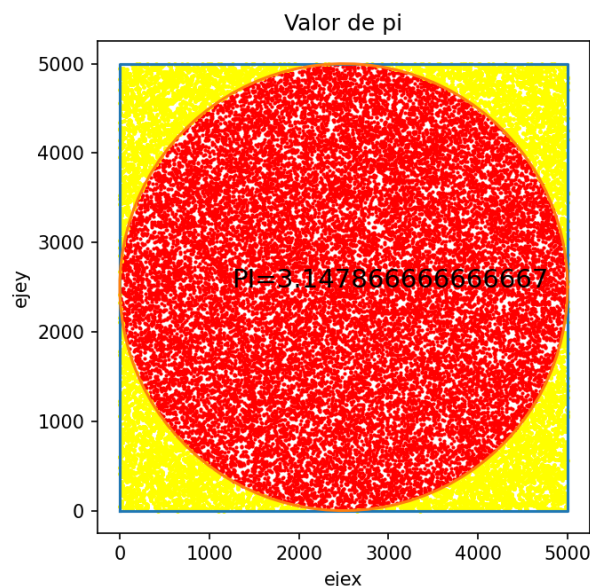
Después graficaremos el circulo con las coordenadas que ya obtuvimos de X y Y con el título de circunferencia también con un plt.plot.

También se graficarán los puntos con plt.scatter analizando las listas x_in, y_in y las listas x_out, y_out indicando el color y el tamaño.

Y en el centro de nuestro circulo se imprimirá el valor calculado de pi que obtuvimos con el largo de la lista entre el numero de puntos por 4.

Y para la relación de aspecto utilizaremos el plt.gca que nos ayudara a que no se deforme nuestra grafica.

Y por último utilizaremos ylabel y xlabel para darle el titulo de los ejes, y graficando todo con plt.show() que nos mostrara en el plano todo lo que hemos realizado. Obteniendo lo siguiente como resultado:



BINARIO/DECIMAL

```
print("PROGRAMA QUE TRANSFORMA DE BINARIO A DECIMAL Y VICEVERSA")
print("Seleccione si es binario o decimal:")
#####modificar el conversor#####
conversor=input()
#####

if conversor=="decimal":
    print("Teclee el numero decimal:")
    #####modificar el numero#####
    numero=int(input())
    #####
    print("El numero",numero)
    deci=[]
    while numero>0:
        residuo=numero%2
        numero=numero//2
        deci.append(residuo)
    else:
        num_bin=deci[::-1]
        print("En binario es:",num_bin)

else:
    if conversor=="binario":
        print("Teclee el numero binario")
        #####modificar el numero#####
        Bin=int(input())
        #####modificar el numero#####
        print("El numero binario:",Bin)
        bina=list(str(Bin))
        largo=len(bina)
        reverso=bina[::-1]
        lista_bin=[]
        for i in range(largo):
            num_bin=(2**i)*(int(reverso[i]))
            lista_bin.append(num_bin)
        print("En numero decimal es:",sum(lista_bin))
    else:
        print("Teclee de nuevo la opcion")
```

En la práctica de decimal a binario lo que hice fue modificar la estructura original del examen usando de decimal a binario un while y de binario a decimal un ciclo for ya que es más fácil al momento de recorrer una lista.

Primero lo que se pide es que el usuario introduzca si quiere decimal o binario haciendo la evaluación con el condicional if/else. Si es decimal entra en donde se ubica el ciclo while/else pero antes le pregunta al usuario el numero a convertir y después imprimiendo el número, luego inicializa una lista vacía donde se almacenara la conversión a binario repitiendo el ciclo siendo que si el numero

```
#####modificar el conversor#####
conversor=input()
#####

if conversor=="decimal":
    print("Teclee el numero decimal:")
    #####modificar el numero#####
    numero=int(input())
    #####
    print("El numero",numero)
    deci=[]
    while numero>0:
        residuo=numero%2
        numero=numero//2
        deci.append(residuo)
    else:
        num_bin=deci[::-1]
        print("En binario es:",num_bin)
```

es mayor a 0, obtiene el residuo de la división después divide el numero entre 2 ya que la base de los números binarios es base 2 y lo almacena en la misma variable para actualizarla y así vaya decreciendo con cada división, y luego el residuo se almacena en la lista vacía `deci=[]` una vez que numero ya no sea mayor a 0 se sale del ciclo y en lugar de imprimir la lista como esta se voltea ya que al momento de resolverlo el numero binario sale volteado y se tiene que leer de adelante hacia atrás. Obteniendo el siguiente resultado.

```
PROGRAMA QUE TRANSFORMA DE BINARIO A DECIMAL Y VICEVERSA
Seleccione si es binario o decimal:
decimal
Teclee el numero decimal:
100
El numero 100
En binario es: [1, 1, 0, 0, 1, 0, 0]
```

Esto se puede ya que se almacena el numero en la lista de uno en uno con cada repetición del ciclo.

El segundo apartado entra en la excepción y con `if/else` anidados pregunta nuevamente si se escribió binario si no fue así se pasa a la excepción y pide que se escriba de nuevo para lo que se tiene que volver a correr el programa. Si cumple la condición pasa a leer el numero binario. Y aquí fue donde modifique el código original ya que lo que hice

```
else:
    if conversor=="binario":
        print("Teclee el numero binario")
        #####modificar el numero#####
        Bin=int(input())
        #####modificar el numero#####
        print("El numero binario:",Bin)
        bina=list(str(Bin))
        largo=len(bina)
        reverso=bina[::-1]
        lista_bin=[]
        for i in range(largo):
            num_bin=(2**i)*(int(reverso[i]))
            lista_bin.append(num_bin)
        print("En numero decimal es:",sum(lista_bin))
    else:
        print("Teclee de nuevo la opcion")
```

fue primero introducir el numero en binario después de esto se convierte en una lista y los datos se convierten en string o carácter para que se separe cada número. Después de este primer paso se analiza el largo de la lista y después se voltea con la variable `reverso` ya que para obtener el numero decimal se tiene que elevar 2 a la potencia `n` de adelante hacia atrás como en el siguiente ejemplo:

$$\begin{array}{cccc}
 8 & 4 & 2 & 1 \\
 2^3 & 2^2 & 2^1 & 2^0 \\
 1 & 1 & 0 & 0
 \end{array}
 \cdot
 \begin{array}{cc}
 1 & 1
 \end{array}$$

$$8 + 4 + 0 + 0$$

Entrando ahora si a el ciclo `for` que itera en el rango del largo de la lista, haciendo la operación 2 se eleva a la potencia `i` la cual es cada iteración por el numero de lista que va aumentando con cada iteración y es convertido en entero con `int`,

todo esto se almacena en la `lista_bin` ya como números enteros, para luego al imprimir el numero suma todos sus valores dando un resultado en decimal como el siguiente ejemplo:

```
PROGRAMA QUE TRANSFORMA DE BINARIO A DECIMAL Y VICEVERSA
Seleccione si es binario o decimal:
binario
Teclee el numero binario
1100100
El numero binario: 1100100
En numero decimal es: 100
```

Así obteniendo el resultado tanto del numero decimal como el numero binario.