INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Cómputo

Programa palíndromos

Teoría de la computación

Alumno:
Reyes Garnelo Uziel Bruno

Profesor: Genaro Juárez Martinez

Grupo: 5BM1

15 de junio de 2023

1. Introducción:

Los palíndromos binarios son secuencias de bits que se leen de la misma manera de izquierda a derecha y de derecha a izquierda. En otras palabras, son palabras binarias simétricas que conservan su significado sin importar el orden de lectura. Estos palíndromos tienen una estructura fascinante y presentan desafíos interesantes en el campo de la informática y las matemáticas.

Los palíndromos binarios son objeto de estudio en diversas áreas, como la teoría de la información y la criptografía. Su naturaleza simétrica los convierte en herramientas útiles para la detección de errores en la transmisión de datos, ya que cualquier alteración en la secuencia original resultará en la pérdida de la propiedad de palíndromo. Además, se utilizan en algoritmos de compresión de datos, donde la detección de patrones y simetrías puede ayudar a reducir el tamaño de la información almacenada. La presencia de palíndromos binarios en la teoría de números también ha sido objeto de investigación, revelando conexiones sorprendentes entre las propiedades de los números binarios y los palíndromos. En resumen, los palíndromos binarios son objetos intrigantes que desempeñan un papel importante en varios campos de estudio, destacando su relevancia en el ámbito de la información y las comunicaciones.

2. Presentación del problema:

Realizar un programa que construya palíndromos de un lenguaje binario. El lenguaje deberá solicitar únicamente la longitud del palíndromo a calcular, de esta manera el programa deberá construir el palíndromo de manera aleatoria. La longitud máxima que podría alcanzar un palíndromo será de 100,000 caracteres. La salida del programa se irá a un archivo de texto y ahí especificarán qué regla se seleccionó y la cadena resultante hasta llegar a la cadena final. El programa deberá ofrecer dos opciones: que el usuario defina la longitud del palíndromo o que lo genere todo de manera automática.

La gramática libre de contexto que construye palíndromos, se define con las siguientes reglas de producción.

- P -> e
- P -> 0
- P -> 1
- P -> 0P0
- P -> 1P1

3. Desarrollo:

3.1. Desarrollo del programa

El lenguaje seleccionado para resolver este problema es Python.

A continuación, mostraremos el código desarrollado.

```
1 import random
```

Usamos la librería 'random' para la generación aleatoria de números.

```
1  print('Programa Palindromos!')
2
3  # Preguntamos si queremos que la construccion sea automatica o manual
4  print('Deseas que la longitud del palindromo eliga de forma manual o automatica?'
       )
5  modo = int(input((" 0. Manual\n 1.Automatica\n")))
6  longitud = ''
7  if modo == 0:
8      longitud = int(input('Longitud del palindromo? --> ')) # Solicitamos la
           longitud del palindromo
9  else:
10     longitud = random.randint(1,100000) # Generamos la longitud de forma
           aleatoria
11
12 print('La longitud del palindromo sera de {}'.format(longitud))
```

En el bloque anterior, se pregunta al usuario si quiere que la longitud del palíndromo se determine de forma automática o manual. Si el usuario decide ingresarla el sistema pide un número n.

Si se debe generar automáticamente, se genera un número aleatorio del 1 al 100,000, que sera la longitud del palíndromo a generar.

```python
# Funcion que devuelve una regla de produccion de forma aleatoria en funcion del numero de terminales que posee
def regla_produccion(nivel:str):
    gramatica = {"0" : [[""]],
                 "1" : [["0"], ["1"]],
                 "2" : [["0", "P", "0"], ["1", "P", "1"]]
                 }
    return "".join(random.choice(gramatica[nivel]))
```

Primero, definimos una función auxiliar que toma las reglas de producción y las codifica en niveles.

- Nivel '0' : e
- Nivel '1' : 0 ó 1
- Nivel '2' : 0P0 ó 1P1

La función recibe un "nivel", y en base al nivel selecciona de forma aleatoria una regla correspondiente al nivel, la concatena en una cadena y la revuelve.

```python
# Funcion para generar palindromo
def generar_palindromo(longitud:int):
    #Guardaremos la construccion del palindromo en una cadena para despues escribir en un archivo
    construccion = "Longitud del palindromo a construir --> "+str(longitud)+"\n"
    archivo = open('./Palindromos/Construccion.txt', mode='w', encoding='utf-8') # Creamos y/o abrimos el archivo
    palindromo = "P" # Definimos nuestro palindromo inicial como P
    construccion += palindromo+"\n"
    while longitud > 1: # Si la longitud del palindromo es mayor a 1
        produccion = regla_produccion("2") # Solicitamos una regla de produccion con 2 terminales
        palindromo = palindromo.replace("P", produccion) # Reemplazamos la variable P por regla de produccion obtenida
        construccion += "Regla de produccion --> "+str(produccion)+"\n" # Informamos que regla de produccion se uso
        construccion += palindromo+"\n" # Agregamos la cadena resultante a nuestra construccion
        longitud -= 2 # Restamos a la longitud 2 (numero de terminales agregadas)

    if longitud == 1: # Si la longitud es 1
        produccion = regla_produccion("1") # Solicitamos una regla de produccion con 1 terminales
        palindromo = palindromo.replace("P", produccion) # Reemplazamos la variable P por regla de produccion obtenida
        construccion += "Regla de produccion --> "+str(produccion)+"\n" # Informamos que regla de produccion se uso
```

```
        construccion += palindromo # Agregamos la cadena resultante a nuestra construccion

    else:
        produccion = regla_produccion("0") # Solicitamos una regla de produccion con 0 terminales
        palindromo = palindromo.replace("P", produccion) # Reemplazamos la variable P por regla de produccion obtenida
        construccion += "Regla de produccion --> e\n" # Informamos que regla de produccion se uso
        construccion += palindromo # Agregamos la cadena resultante a nuestra construccion
    archivo.write(construccion) # Escribimos toda la construccion en el archivo
    archivo.close() # Cerramos el archivo
    return palindromo # Retornamos el palindromo generado
```

En el fragmento de código anterior, la función que genera el palíndromo recibiendo como argumento la longitud de la cadena a resolver. Expliquemos sus partes.

Se crea una cadena llamada 'construccion' donde se almacenará la evolucion de la cadena. Se inicializa con ' Longitud del palindromo a construir'+str (longitud) + " n

Posteriormente, se crea y/o abre un archivo llamado 'Construccion.txt'

Cualquier palíndromo con cualquier longitud lo podemos definir como 'P'. Así que lo definimos de esa manera y lo agregamos a nuestra variable 'construccion'.
La lógica para la construcción es simple. Partimos de una longitud, mientras esa longitud sea mayor a 1, solicitaremos una regla de produccion con 2 terminales, posteriormente reemplazamos 'P' por nuestra regla de producción, agregamos la regla usada en la variable construcción y la cadena resultante de haber usado esa regla de producción. Y un paso muy importante, reducimos en 2 (que es el numero de terminales que obtuvimos con nuestra regla de producción) a la longitud que nos queda. Este proceso se hace de forma iterativa hasta tener dos caminos posibles, que la longitud sea 1 ó 0.

Posteriormente, si la longitud es 1, solicitamos una regla con nivel 1 si es 0, solicitamos una regla con nivel 0, y se termina la producción del palindromo.

Finalmente, escribimos la construcción en el archivo creado, cerramos el archivo y devolvemos el palíndromo generado.

```
generar_palindromo(longitud)
```

Finalemente, llamamos a la función anterior dandole nuestra longitud como argumento.

Veamos los resultados obtenidos.

Probemos con tres diferentes longitudes.

Ejecución del programa:

```
(base) bruno-rg@bruno-rg ~/Documents/6to Semestre/TC main ± /bin/python3 "/home/bruno-rg/Documents/6to Semestre/TC/Palindromos/PalindromoBinario.py"
Programa Palindromos!
¡Deseas que la longitud del palindromo eliga de forma manual o automatica?
 0. Manual
 1.Automatica
0
¿Longitud del palindromo? --> 9
La longitud del palindromo será de 9
```

Archivo generado por el programa:

Construccion.txt M

Palindromos > Construccion.txt

```
1  Longitud del palindromo a construir ⟶ 9
2  P
3  Regla de produccion ⟶ 1P1
4  1P1
5  Regla de produccion ⟶ 1P1
6  11P11
7  Regla de produccion ⟶ 1P1
8  111P111
9  Regla de produccion ⟶ 0P0
10 1110P0111
11 Regla de produccion ⟶ 0
12 111000111
```

Ejecución del programa:

```
(base) bruno-rg@bruno-rg ~/Documents/6to Semestre/TC main ± /bin/python3 "/home/bruno-rg/Documents/6to Semestre/TC/Pal
indromos/PalindromoBinario.py"
Programa Palindromos!
¡Deseas que la longitud del palindromo eliga de forma manual o automatica?
 0. Manual
 1.Automatica
0
¿Longitud del palindromo? --> 10
La longitud del palindromo será de 10
```

Archivo generado por el programa:

Construccion.txt M

Palindromos > Construccion.txt

```
1  Longitud del palindromo a construir ⟶ 10
2  P
3  Regla de produccion ⟶ 0P0
4  0P0
5  Regla de produccion ⟶ 0P0
6  00P00
7  Regla de produccion ⟶ 1P1
8  001P100
9  Regla de produccion ⟶ 0P0
10 0010P0100
11 Regla de produccion ⟶ 0P0
12 00100P00100
13 Regla de produccion ⟶ e
14 0010000100
```

Ejecución del programa:

```
(base) bruno-rg@bruno-rg ~/Documents/6to Semestre/TC main ± /bin/python3 "/home/bruno-rg/Documents/6to Semestre/TC/Palindromos/PalindromoBinario.py"
Programa Palindromos!
¡Deseas que la longitud del palindromo eliga de forma manual o automatica?
 0. Manual
 1.Automatica
1
La longitud del palindromo será de 36116
```

Archivo generado por el programa:

Construccion.txt M

Palindromos > Construccion.txt

```
36083  Regla de produccion ⟶ 1P1
36084  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36085  Regla de produccion ⟶ 1P1
36086  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36087  Regla de produccion ⟶ 1P1
36088  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36089  Regla de produccion ⟶ 1P1
36090  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36091  Regla de produccion ⟶ 1P1
36092  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36093  Regla de produccion ⟶ 1P1
36094  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36095  Regla de produccion ⟶ 1P1
36096  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36097  Regla de produccion ⟶ 1P1
36098  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36099  Regla de produccion ⟶ 0P0
36100  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36101  Regla de produccion ⟶ 0P0
36102  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36103  Regla de produccion ⟶ 1P1
36104  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36105  Regla de produccion ⟶ 0P0
36106  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36107  Regla de produccion ⟶ 1P1
36108  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36109  Regla de produccion ⟶ 0P0
36110  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36111  Regla de produccion ⟶ 0P0
36112  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36113  Regla de produccion ⟶ 0P0
36114  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36115  Regla de produccion ⟶ 0P0
36116  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36117  Regla de produccion ⟶ 1P1
36118  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
36119  Regla de produccion ⟶ e
36120  10111100010101101011000011011111111000010010000101011000100110111110000001011001011010110101111110111001110001010011011001
```

Como podemos observar, el programa funciona correctamente para longitudes pares, impares y bastante grandes.