

Instituto Politécnico Nacional

Escuela Superior de Cómputo



Programa 7: Palindromes

Autor: Colín Ramiro Joel

Materia: Teoría de la Computación

Grupo: 4CM2

Profesor: Juarez Martínez Genaro

Fecha de entrega: **29 de Diciembre 2021**

Introducción

En un contexto fuera de las ciencias de la computación, un palíndromo es una palabra o frase que se lee igual en un sentido que en otro (derecha a izquierda y viceversa). Habitualmente, las frases palindrómicas se resienten en su significado cuanto más largas son.

Ya en un contexto computacional, no se tiene que tomar necesariamente a los palíndromos como palabras reales en un idioma cualquiera. Se puede también pensar en un palíndromo como cualquier secuencia de letras que se lee igual hacia adelante y hacia atrás, como lo puede ser **xyzyzyx** o un conjunto de números como **1001001**. Como bien ya se sabe a una secuencia de letras se le conoce como una cadena. Así entonces se puede considerar que cualquier cadena que contenga una sola letra es, por defecto, un palíndromo. Ahora bien, una cadena puede no contener ninguna letra; a una cadena de cero letras la llamamos una cadena vacía. Esta cadena vacía también es un palíndromo, ya que se "lee" del mismo modo hacia adelante y hacia atrás como bien se definió al inicio de esta sección. Así que ahora consideremos que cualquier cadena que contenga como mínimo una letra es un palíndromo.

Instrucciones

Realizar un programa que construya palíndromes 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 selecciono 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 palíndromes, se define con las siguientes reglas de producción.

1. $P \rightarrow e$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

En el reporte debe de estar también el código de la implementación en latex, no en imágenes.

Desarrollo

Para la realización de este programa se implementó un menú principal en el cual el usuario seleccionará el modo en el que se introducirá la longitud del palíndromo. Se puede ya sea el modo manual, el automático y además, se puede seleccionar para salir del programa.

Se implementó una función llamada **palindrome**, la cual dependiendo de la selección del usuario, preguntará cuál es la longitud deseada para el palíndromo si se selecciona el modo manual. Por otro lado, si se selecciona el modo automático generará un número aleatorio entre 1 y 100,000 el cual será la longitud del palíndromo.

Conforme se va generando el palíndromo, se van aplicando las reglas definidas en el código, estas reglas son las que se especificaron previamente en la sección e Instrucciones. La construcción del palíndromo se especifica en el archivo de texto **"SalidaP7.txt"**

Capturas del Funcionamiento

En esta sección se encuentran las capturas de pantalla del funcionamiento del programa, tanto de la consola, como de los archivos generados. Las capturas se ordenaron conforme a la opción seleccionada en el programa:

1. Modo Manual

```
C:\Users\Joel\OneDrive\Documents\Joel\4to\TC\Programas\Programa7-Palindromes>python Programa7_TC.py
**Menú Principal Programa 7**
(Cómo será la longitud)

1.- Longitud Manual
2.- Longitud Aleatoria
3.- Salir del Programa
4

Se ha seleccionado el modo manual...
Digite la longitud: 20
La longitud del Palindrome es: 20
Construyendo Palindrome...
P
1P1
11P11
011P110
00011P11001
0100011P1100101
1010011P110010101
1101010011P1100101011
La construcción de los palindromos se encuentra en el archivo 'SalidaP7.txt.'
El palindrome final generado es: 11010100111100101011
```

```
SalidaP7: Bloc de notas
Archivo Edición Formato Ver Ayuda
**Este es el archivo de txt donde se encuentra la construcción del Palindrom**

La longitud del Palindrome es: 20
Construyendo Palindrome...
P
Se aplico la regla de producción 5: (P -> 1P1) 1P1
Se aplico la regla de producción 5: (P -> 1P1) 11P11
Se aplico la regla de producción 4: (P -> 0P0) 011P110
Se aplico la regla de producción 4: (P -> 0P0) 0011P1100
Se aplico la regla de producción 5: (P -> 1P1) 10011P11001
Se aplico la regla de producción 4: (P -> 0P0) 010011P110010
Se aplico la regla de producción 5: (P -> 1P1) 1010011P1100101
Se aplico la regla de producción 4: (P -> 0P0) 01010011P11001010
Se aplico la regla de producción 5: (P -> 1P1) 101010011P110010101
Se aplico la regla de producción 5: (P -> 1P1) 1101010011P1100101011
Se aplico la regla de producción 1: (P -> e) ['1101010011P1100101011']

El palindrome final generado es: 11010100111100101011
```

2. Modo Automático


```

palindrome = "P"
print("Construyendo Palindrome...\n" + palindrome + "\n")
archivo.write("Construyendo Palindrome...\n" + palindrome + "\n")

for i in range(0, longPal2 + 1):
    if (i == longPal2):
        finPalin = palindrome.split()
        if(impar == False): #Regla de Producci n (1) P -> e
            archivo.write("Se aplica la regla de producci n 1: (P
                           -> e) ")
            archivo.write(str(finPalin) + "\n")
        if(impar == True):
            regla_2_3 = randint(2,3)
            if(regla_2_3 == 2): #Regla de Producci n (2) P -> 0
                archivo.write("Se aplica la regla de producci n 2: (
                               P -> 0) ")
                palindrome = "0" + palindrome
                print(palindrome)
                archivo.write(str(finPalin) + "\n")
            if(regla_2_3 == 3): #Regla de Producci n (3) P -> 1
                archivo.write("Se aplico la reglade producci n 3: (
                               P -> 1) ")
                palindrome = "1" + palindrome
                print(palindrome)
                archivo.write(str(finPalin) + "\n")
        else:
            regla_4_5 = randint(4,5)
            if (regla_4_5 == 4): #Regla de Producci n (4) P -> 0P0
                archivo.write("Se aplico la regla de producci n 4: (P
                               -> 0P0) ")
                palindrome = "0" + palindrome + "0"
                print(palindrome)
                archivo.write(str(palindrome) + "\n")
            if (regla_4_5 == 5): #Regla de Producci n (5) P -> 1P1
                archivo.write("Se aplico la regla de producci n 5: (P
                               -> 1P1) ")
                palindrome = "1" + palindrome + "1"
                print(palindrome)
                archivo.write(str(palindrome) + "\n")

    print("\nLa construccion de los palindromos se encuentra en
          el archivo 'SalidaP7.txt.'")
    print("\nEl palindrome final generado es: " + palindrome[0:
        longPal2] + palindrome[longPal2+1
        :]+ "\n")
    archivo.write("\nEl palindrome final generado es: " +
        palindrome[0:longPal2] +
        palindrome[longPal2+1 :]+ "\n")

opc = 0
salir = 3
while opc != salir:
    print("***Men Principal Programa 7***")
    print(" Como ser la longitud?")
    opc = int(input(''))
    1.- Longitud Manual
    2.- Longitud Aleatoria

```

```

3.- Salir del Programa
'''
if (opc == 1) or (opc == 2):
    palindrome(opc)
elif opc == 3:
    print("Saliendo del Programa. Hasta Luego!!!")
else:
    print("Opci n inv lida , Vuelva a intentar")

```

Conclusiones

En esta sección se encuentran las conclusiones de la práctica. Me apoye de las láminas presentadas en clase, así como de otros recursos encontrados en diferentes páginas web relacionadas con el tema, para la realización del programa así como para reforzar los conocimientos esperados. En cuanto al programa puedo concluir que a mi parecer fue el más sencillo de implementar en comparación de los otros, ya que como también se puede observar en el código, es el más pequeño por así decirlo en cuanto a líneas de código y también en cuanto al tiempo efectivo de realización.

Referencias

1. J. Andres Montoya. (2016). Introduccion a la teoria de la computacion: via palindromos. Diciembre 21, 2021, de Universidad Industrial de Santander Sitio web: <http://matematicas.uis.edu.co/jmontoya/sites/default/files/algoritmos-automatas-tiempo-real.pdf>
2. Terry Anderson, Narad Rampersad, Nicolae Santean, and Jeffrey Shallit. (2019). Finite Automata, Palindromes, Powers, and Patterns. Diciembre 21, 2021, de University of Waterloo Sitio web: <https://cs.smu.ca/nic.santean/art/pal.pdf>
3. Línea de Código. (2007). Palíndromos con Java. Diciembre 21, 2021, de Línea de Código Sitio web: <https://lineadecodigo.com/java/saber-si-una-palabra-es-un-palindromo-con-java/>