



Stephen Jakson Cisneros Zamarripa

16480229

Proyecto 2

Lenguajes Y Autómatas

Monterrey, Nuevo León a 7 de Octubre del 2018

Programa KMP

```
class KMP:

    from time import time

    def parcial(auto, patron):

        """ Calcular tabla de coincidencias parciales: String ->
            [Int]"""

        ret = [0]

        for i in range(1, len(patron)):

            j = ret[i - 1]

            while j > 0 and patron[j] != patron[i]:

                j = ret[j - 1]

            ret.append(j + 1 if patron[j] == patron[i] else j)

        return ret
```

```
def busqueda(auto, T, P):
```

```
    """
```

```
        KMP algoritmo principal de búsqueda: Cadena ->
            Cadena -> [Int]
```

```
        Devuelve todas las posiciones coincidentes de la cadena
            de patrón P en S
```

```
    """
```

```
parcial, ret, j = auto.parcial(P), [], 0
```

```
    for i in range(len(T)):
        while j > 0 and T[i] != P[j]:
            j = parcial[j - 1]
        if T[i] == P[j]: j += 1
        if j == len(P):
            ret.append(i - (j - 1))
            j = 0
```

```
    return ret
```

```
    start = time ()
```

```
    end = time () - start
```

Programa para validar un correo electrónico

```
import re
```

```
correo='sj7@hotmail.com'
```

```
if re.match('^[a-z0-9_\-\.\.]+\@[a-z0-9_\-\.\.]+\.[a-z]{2,15}$',correo.lower()):
```

```
    print "Correo Correcto"
```

```
    else:
```

```
        print "Correo Incorrecto"
```

Programa para validar un correo electrónico mediante una expresión regular

```
import re

correo = 'cz7@hotmail.com'

emailRegex = r'^([\w]+)@([\w]+)' + '.com'

match = re.search(emailRegex, correo)

    if match:

        print(correo + ' es una dirección válida.')

    else:

        print(correo + ' no es una dirección válida.')
```

Reporte

Introducción

en estos programas se verán 2 cosas, la primera es que el programa 1 se trata de ver cómo funciona el algoritmo KMP y en el otro programa se trata de validar el formato de un correo electrónico y también validarlo mediante una expresión regular.

KMP busca la aparición de una palabra P dentro de una “cadena de texto” principal C, empleando la simple observación de que cuando no sucede una coincidencia, la palabra misma contiene suficiente información para determinar cuándo puede ocurrir la siguiente coincidencia, reexaminando caracteres previamente coincidentes.

¿Qué se hizo?

Se realizaron 3 programas, el 1 es acerca del algoritmo KMP, el 2 es validar el formato de un correo electrónico y el 3 validar el formato de un correo electrónico mediante una expresión regular.

¿En dónde se realizó?

Estos programas se realizaron en lenguaje PYTHON

¿Cómo se realizó?

En el programa 2 se creó una función que realiza una validación a una cadena de texto para saber si tiene el formato correcto para ser un correo electrónico.

En el programa 3 se creó una función de expresión regular necesaria para validar un correo electrónico en Python.

¿Para qué se realizó?

Se realizó para ver cómo funciona el algoritmo KMP y para saber cómo se valida un formato de correo electrónico para saber si es válido o no.