

Buscar cómo funciona el algoritmo KMP

El algoritmo **KMP** consiste en verificar para cada posición del texto o cadena de caracteres en la que se hace una búsqueda exacta en la que el patrón puede concordar.

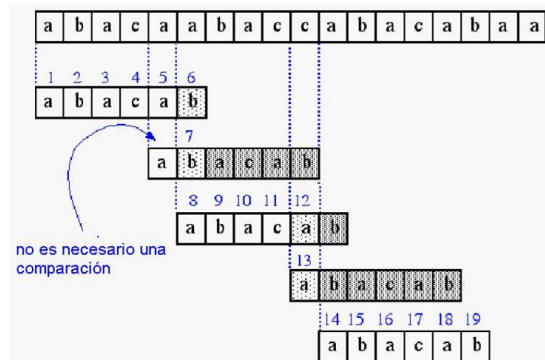
Su objetivo es:

- 🚦 Buscar la existencia de palabra dentro de una cadena de texto.
- 🚦 Consideremos que S sea el texto de longitud N y es la entrada del algoritmo.
- 🚦 Y P sea el patrón de longitud M la cual es la palabra a buscar en el texto.

El algoritmo KMP, trata de localizar la posición de comienzo de una cadena, dentro de otra. Antes que nada, con la cadena a localizar se pre calcula una tabla de saltos (conocida como tabla de fallos) que después al examinar entre si la cadena se utiliza para hacer saltos cuando se localiza un fallo.

Se considera que cuando el programa es realizado este se marca como la cadena de texto más larga y una sub cadena que generalmente es más corta, el programa consiste en que ir colocando la sub cadena desde el principio verificándonos que no tenga un fallo que quiere decir que mientras los caracteres de la cadena y de la sub cadenas sean iguales al programa el programa seguirá recorriendo, hasta que se encuentra con el fallo o la inconsistencia entre las dos y se detiene, después de eso hace que la sub cadena empiece nuevamente a recoger los espacios pero esta vez con un espacio más a la derecha, y así seguirá hasta que el programa detecte que las dos sean iguales.

Algoritmo KMP



• Complejidad: $O(n + m)$

8

Validar formato de correo electrónico mediante una expresión regular.

La expresión regular necesaria para validar un correo electrónico según la liga <https://www.lawebdelprogramador.com/codigo/Python/2040-Validar-cuenta-de-correo.html> es la siguiente expresión: `('^[a-z0-9_\\-\\.]+@[a-z0-9_\\-\\.]+\\.([a-z]){2,15}$', correo.lower())` en la cual se me hizo un poco más entendible ya que nos está separando los caracteres que a fuerzas debe de contener un correo electrónico.

Seudocódigo:

- ✚ Se crea variable "correo" acompañada de un input acompañado de un correo electrónico.
- ✚ Después se crea una condición con la expresión regular en la cual busca que caracteres son los que permite.
- ✚ Después se crea un print el cual nos imprimirá si el correo electrónico ingresado es correcto.
- ✚ Si detecta que no es correcto pasa por el else que este nos sirve para indicar que el correo ingresado no cumple con la condición así que nos va a imprimir que es incorrecto.
- ✚ Después viene lo del from time el cual nos está indicando el tiempo en que nuestro programa tarda en ejecutarse.
- ✚ Fin

Validar formato de un correo electrónico mediante método propio:

En este programa primero que nada tuvimos que buscar información sobre los métodos que existen para poder separar nuestro código en dos con el fin de que nos vaya verificando por separado la existencia del carácter como el "@" y el "." que en un correo electrónico son los caracteres más importantes. Para este ejemplo encontramos que se podía utilizar el método Split el cual su función es que es capaz de separar una cadena extensa en partes menos extensas.

Y como ya habíamos mencionado el from time es más que nada para ver cuánto tiempo tarda nuestro programa ejecutándose el cual cada vez que se corra el programa puede que vaya tardando más o se corra mucho más rápido y también esto es dependiendo de la máquina de donde estés corriendo el programa ya que no todas son iguales.






Cual se fue el mejor y diferencias:

Para mí el de expresión regular ya que en este programa es como que más simple y más entendible ya que nos indica que caracteres son permitidos dentro de un correo electrónico que viene siendo el @ y el., ya que si no tuviéramos eso no sería un correo electrónico a comparación del que fue realizado por nosotros es mucho más extenso y un poco más complicado ya que batallamos mucho para poder hacerlo.

¿Para qué se realizó?

Estos programas se realizaron para que nosotros supiéramos que hay muchas maneras de poder crear una validación de un correo electrónico y con esto ya vimos que puede haber muchos programas que hagan lo mismo, pero muchos tienen métodos para saber si la validación del correo electrónico es válida o no es válida y otros no necesariamente cuentan con eso, pero aun así tienen la misma función.

En la máquina que se probó este código fue en la siguiente:

-  Computadora marca dell
-  Procesador Intel® core i5
-  Memoria Ram de 8,00 GB
-  Sistema Operativo de 64 bits
-  Windows 10