Universidad Autónoma del Estado de México

Ingeniería en Computación

Materia: Algoritmos Geneticos
Laboratorio 02

Axel Valenzuela Juárez
Profesor: Dr. Asdrúbal López Chau
20 de Febrero del 2020

1. Problema: Encontrar contrasenia por medio de furza bruta

El problema consiste en encontrar una contrasenia donde las posibles combinaciones estaban dadas por únicamente vocales y una longitud de 4 , esto por medio de encontrar todas las posibles combinaciones hasta dar con la correcta, en este caso sería rápido pero con más combinaciones sería un método muy tardado y hoy en día casi inútil a la hora de poner en práctica. Siendo este un gran ejemplo para demostrar la gran utilidad de los algoritmos genéticos en este caso a la hora de encontrar una contrasenia, Claro siendo este primer taller la demostración del método largo y muy poco practico que resulta ser utilizar fuerza bruta.

Â¿Qué es un ataque de fuerza bruta?

Un ataque de fuerza bruta es un método de prueba y error utilizado para obtener información como contrasenias u otros códigos de acceso. El atacante prueba una variedad de posibles combinaciones de caracteres con la ayuda del software apropiado, con el objetivo de encontrar la secuencia de caracteres deseada para obtener acceso ilegal a datos sensibles, parcialmente encriptados.

1.1. Desarrollo del Problema

```
17
18
19 arr=[1,2,3,4] #arreglo con la contraseña a encontrar
20 arr2=[]
21 i=0 #auxiliar para guardar las iteraciones
22 vocal = dict() #declaracion de diccionario para convertir los numeros a vocales
23
```

Figura 1: Creación de arreglo.

El primer paso fue entender muy bien el problema, se quería encontrar una contrasenia con posibles combinaciones de vocales y de longitud 4. Cuando se comprendió el problema empecé programando un arreglo para guardar la contrasenia a encontrar. Fig 1

El siguiente paso que decidí hacer fue crear 4 ciclos for uno para cada carácter de la contrasenia, en cada uno de ellos los rangos serian de 1 a 6 ya que en total son 5 vocales.

```
31 for a  in range(1,6):#se hace un for para cada vocal en la cual se encontrara la contraseña
32     for b in range(1,6):
33         for c in range(1,6):
34             for d in range(1,6):
35                 i=i+1 #se guarda el numero de iteraciones y se van sumando
36                 arr2=[a,b,c,d]#los valores de los for se guardan en un arreglo para ser comparados despues con la co
37                 if(arr==arr2):#se compara el arreglo guardado con la contrseña, en caso de ser iguales se procede a
38                    print("Encontrado en :",i)#se imprime el numero de iteraciones
39                    print("contraseña con numeros: " ,a,b,c,d )#se imprime la contraseña con los numeros
40                    print("Contraseña desifrada con vocales: ",vocal.get(a),vocal.get(b),vocal.get(c),vocal.get(d))#s
41                    #se usa el metodo get() para obtener un elemento del diccionario segun su indice
42
```

Figura 2: Multiples for.

Utilice un contador para saber cuentas iteraciones fueron necesarias hasta encontrar la contrasenia, este fue introducido después del último for. Fig: 2

Cada valor en los for fueron introducidos en un arreglo para posteriormente compararlo con el arreglo de la contrasenia por medio de un if, de no ser iguales el programa seguiría comprobando carácter por carácter hasta encontrar la contrasenia, en caso de ser idénticos el programa imprimirá un mensaje con el número de iteración y la contrasenia dada en números donde 1=a,2=e,3=i,4=o,5=u, ya que esto no es muy practico y se necesitaba saber cual es la contrasenia en vocales, decidí utilizar un diccionario en los valores de las

vocales esto con la finalidad de no utilizar múltiples if para simplemente cambiar de números a letras.

Para definir un nuevo diccionario utilice la función dict() y se la asigne a una variable en este caso llamada vocal.

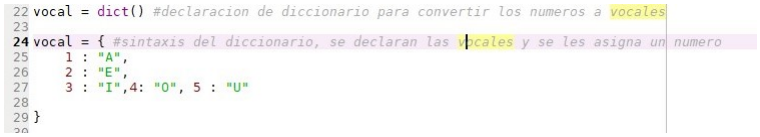

Figura 3: Diccionario.

Una vez creado el diccionario, le asigne valores de acuerdo con la sintaxis, dándole a el primer elemento un valor de A, al segundo E y así sucesivamente.

Una vez hecho esto solo mande a llamar al diccionario pasándole la variable que contenía el índice del for esto con el método get() que devuelve el valor del diccionario de acuerdo al índice.

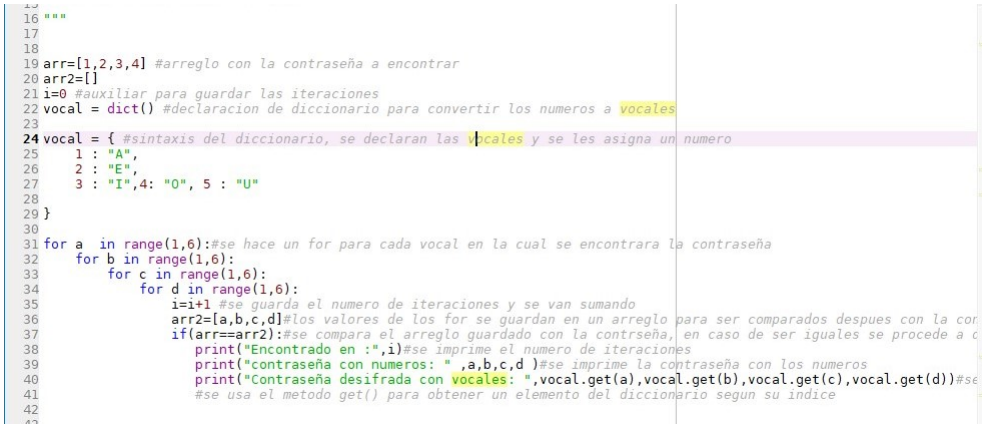

Figura 4: Codigo Completo.

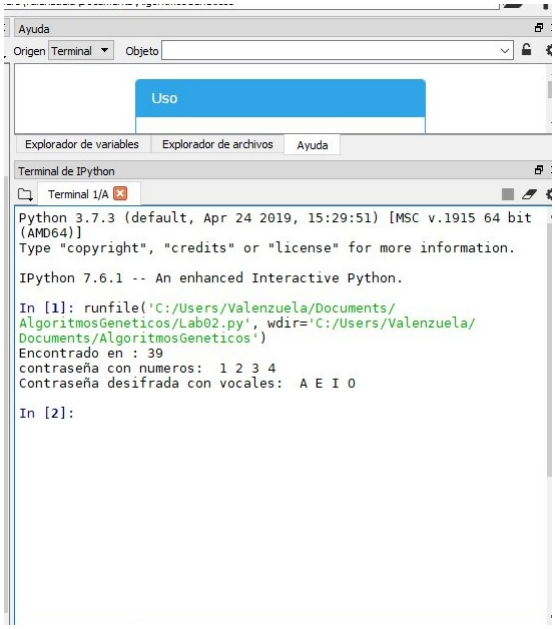

Figura 5: Resultado.

Esto permitió resolver de buena manera convertir los números a sus respectivas vocales.

2. Conclusión

En conclusión, el uso de fuerza bruta para encontrar una contrasenia no es la mejor opción ya que en este caso la longitud y los valores posibles en la contrasenia lo hicieron fácil y rápido de hacer, pero para contresenias mucho más elaboradas y con mucho mas posibles combinaciones sería muy tardado. en este caso es de gran ayuda el uso de los algoritmos genéticos para acelerar en gran medida el proceso para encontrar la contrasenia.

3. Referencias

Montoro, A. F. (2012). Python 3 al Descubierto. RC Libros.