

DOCUMENTO TÉCNICO

Requisitos

Especificación

#RESUMEN DEL PROBLEMA (opcional)

Buscar el modo de contar cual es el carácter con mayor número de repeticiones en una cadena de texto. Si hay 2 caracteres que tienen el mismo número de repeticiones, se buscara mostrar el menor valor en términos alfabéticos.

Entrada:

Se toma una cadena de texto con n número de caracteres en la cual se va a contar el carácter con mayor aparición y que está presente en el alfabeto ASCII.

Salida:

Una respuesta la cual enuncia el carácter con mayor repetición en la cadena de texto acompañado con su número de apariciones en dicha cadena.

Diseño

Estrategia

Se crea una cadena de texto de manera aleatoria del alfabeto ASCII por medio de la librería random. Ya teniendo la cadena de texto, se procede a analizar carácter por carácter para ver cuál es el que tiene mayor aparición dentro de la cadena de texto. Para esto, se va almacenando cada carácter dentro de un diccionario, donde la clave va a ser el carácter y el dato almacenado en esta va a ser el número de repeticiones de dicho carácter, es decir que cuando encuentre el carácter, le va a sumar 1 al número de repeticiones en su respectiva posición del diccionario.

A medida que se van comparando los caracteres de la cadena de texto, se va almacenando la información de cuál es el carácter con mayores repeticiones y su número de apariciones en dos variables distintas. Se impone una condición de que, en caso de encontrar 2 caracteres con el mismo número de repeticiones, se deja en la variable el carácter menormente hablando en el alfabeto ACSII.

Finalmente se muestra el carácter con mayor número de apariciones, así dando por finalizado el programa

#ESTRUCTURA DE DATOS

Se usaron los diccionarios en el momento en el que se guardaban los caracteres de la cadena de texto (claves del diccionario) con su respectivo número de apariciones en esta.

#ALGORITMO

Programa adjunto

Casos de prueba

Entrada	Justificación	Salida
Cadena de texto con 10000 caracteres	Cadena de texto extensa	El carácter con mayor repetición sin demora alguna
	Cadena vacía	(-inf, none)
+	Cadena con 1 solo dato	(1, '+')

Análisis

```

1 import math
2 from time import time
3 from random import randint
4 SIZE = 1000
5 TESTS = 2
1 usage
6 def maxCharOccurrence(S):
7     max_ocurr, max_char = -math.inf, None
8     ascii_tb = {}
9     for c in S:
10         if c in ascii_tb.keys():
11             ascii_tb[c] = ascii_tb[c]+1
12         else:
13             ascii_tb[c]=1
14         if max_ocurr <= ascii_tb[c]:
15             if max_ocurr == ascii_tb[c]:
16                 max_ocurr, max_char = ascii_tb[c], min(c, max_char)
17             else:
18                 max_ocurr, max_char = ascii_tb[c], c
19
20     return (max_ocurr, max_char)
1 usage
21 def stringBuilder(size):
22     s = ""
23     for i in range(size):
24         s+=chr(randint(a=33, b=165))
25     return s
1 usage

```

```

1 usage
26 def main():
27     for test in range(TESTS):
28         s = stringBuilder(int(SIZE))
29         t0 = time()
30         out = maxCharOccurrence(s)
31         tf = time()
32         print(s)
33         print("Time optimum", out, tf-t0)
34 main()

```

Temporal

Esta escrito como comentarios en el código.

Código

Al igual que el punto anterior pueden guiarse con este ejemplo

Documentación

Dentro del código.

Fuentes

Las fuentes son el código sobre el cual escribieron el documento

/arena_1_a