

Tarea 5 Programación y Algoritmos

Carlos Giovanni Encinia González

17 de septiembre 2021

Problema 1

Escribir un programa realice las siguientes operaciones con strings:

Cada uno de los algoritmos implementados se ejecuta de manera de manera estándar en codeblocks. Los datos se le piden al usuario desde consola. Una vez ingresados los datos se da la respuesta que se busca para cada uno de los incisos.

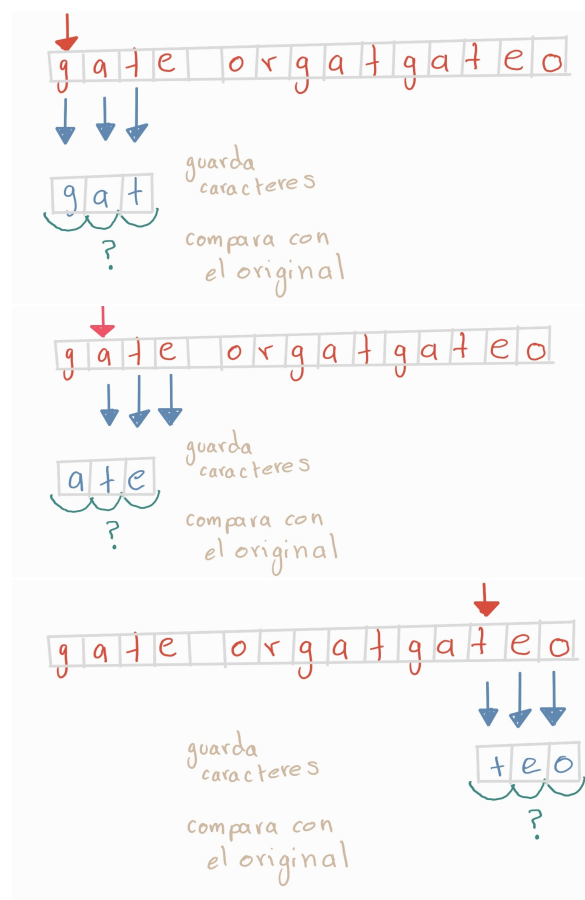
a)

Encuentre si la palabra 'p' está presente en un string 'str', y cuantas veces se repite: `int encuentraPal(char *p, char *str);` regresa: 0 si no se encontró la palabra en 'str'; de otra manera regresa el número de veces que se encontró (1, 2, 3, ...).

Respuesta

El algoritmo "núcleo" en este caso se llama "count_word.c" y se encuentra en el archivo "funciones_a_b.c". Lo que hace el algoritmo es leer el string caracter por caracter, cuando cae en un caracter este se expande hasta la cantidad de caracteres de la palabra que se busca (es necesario calcular el tamaño de lo que se buscará). Por ejemplo en la figura 1 vemos la frase "gate orgatgateo" y lo que queremos buscar es la palabra "gat", entonces iniciamos en el primer caracter y como la palabra a buscar tiene 3 caracteres se comparan los siguientes caracteres hasta contar tres con la palabra a buscar para determinar si se encuentra la palabra, si esto es así, entonces mediante un contador se lleva dicha cuenta. Este mismo procedimiento se repite hasta terminar el arreglo y como resultado da el número total de coincidencias. Es importante señalar que como tal no se guardan los caracteres en un nuevo sub-arreglo, si no que se emula esto, pero solamente con un proceso meramente iterativo. En la figura 1 a la 3 se puede observar como funciona el algoritmo en grandes rasgos.

Figure 1: Movimiento



Se creó una carpeta con un archivo .c donde se encuentran funciones para trabajar con strings, tales como calcular el tamaño o limpiar el string si tiene salto de línea, entre otras cosas. Estas funciones se usaran también para los otros incisos.

En seguida se muestra el resultado del algoritmo cuando se ejecuta y se alimenta con ciertos datos.

El algoritmo le pide al usuario que agregue la palabra/frase en donde se hará la búsqueda y la palabra que se buscará.

Figure 2: Resultado

```

Busqueda de palabras en una frase
=====
Ingresa la frase donde se buscara
Mi gato es Ingles y se llama Indio i

Ingresa la palabra o caracter que se buscara en tu frase
i

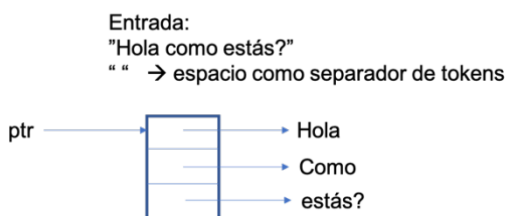
El numero de apariciones es 5
=====

```

b)

Separe un string en tokens de cuerdo a un caracter especial dado como entrada (puede ser espacio, /, %, etc.) y que regrese un arreglo que apunte a cada uno de los tokens encontrados: char ** tokens(char *str, char ch); regresa: NULL → no se encontraron token de otra forma: PTR → ver imagen

Figure 3: Ejemplo



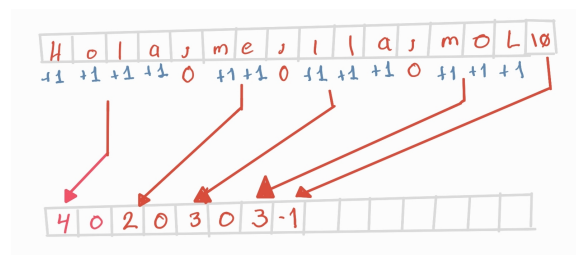
Respuesta

El algoritmo que se implementó lo que hace primero es hacer una pequeña limpieza en los datos de entrada, con el propósito de eliminar los posibles saltos de línea, después se calcula el tamaño del arreglo.

Lo que sigue es calcular el tamaño de cada uno de los token, esto se hace recorriendo el arreglo y guardando el resultados del tamaño sumando 1 por cada caracter antes del caracter que funciona como separador, si se encuentra con el separador se agrega un 0, al finalizar el arreglo a la nueva estructura se le agrega un -1 al final. Figura (4).

Una vez que se tiene este arreglo se procede a crear la memoria dinámica donde se almacenaran los nuevos strings, se comienzan a guardar, mediante el recorrido del string original, muy parecido a lo hecho al inicio, al final de arreglo de apunadores se agrega un apuntador NULL para saber donde termina el arreglo dinámico.

Figure 4: Búsqueda de tamaño



En seguida se muestran los resultados obtenidos al correr el programa.

Figure 5: Resultado

```

Separador de palabras
=====
Ingresa la frase
MianombreaesFlor

Ingresa el separador que se utilizara
a
Los token separados por 'a' son
Mi
nombre
esFlor
=====

```

c)

Concatene dos strings str1 y str2 (creados con memoria dinámica), y regrese el resultado en str1 (como str1 no tiene suficiente memoria para almacenar ambos strings, sera necesario crear nueva memoria y eliminar la anterior):

char *concatena(char *str1, char *str2) regresa: Apuntador hacia str1;

Respuesta Para este problema simplemente se pide al usuario que ingrese dos palabras y estas se concatenan, mediante memoria dinámica, se utiliza la función realloc para almacenar el nuevo string en la memoria del primer string, es importante mencionar que para que esto sea posible, primero se calcula el tamaño de los strings con las funciones que se crearon y se encuentran en el archivo llamado "strings_.h"

En seguida se muestra el resultado al ejecutar el programa.

Figure 6: Resultado

```
Concatena palabras
=====
Ingresa la primera palabra
Carlos

Ingresa la segunda palabra
Giovanny

La palabra concatenada es
CarlosGiovanny
=====
```

que se van haciendo estos movimientos, el cuadro rojo muestra la posición en la que va el recorrido y los cuadros amarillos muestran la matriz 3 x 3 que se genera, esto es emulado de manera iterativa, una vez creada la submatriz de 3x3 se pasan los elementos a un arreglo unidimensional, esto con la intención de poder calcular, el promedio o la mediana. Una vez calculado algún parámetro, métrica o estadística esta se guarda en una nueva matriz la cual contiene en su periferia elementos de la matriz original pero en su centro contiene los valores encontrados con el recorrido de la matriz figura (10). Si el paso de la matriz esta en las coordenadas $i=2, j=3$ entonces el valor que se calcula se almacenara en la nueva matriz en la posición $i = 3, j = 4$, es decir existe un desfase de una unidad. Todo esto se evalúa hasta completar el recorrido total de la submatriz.

Figure 7: Resultado

Problema 2

Programa que calcule el " $W_{3 \times 3}$: moving average and median" de una matriz de tamaño $L_1 \times L_2$ (el programa debe pedir las dimensiones).

$$A(n) = Average(W_{3 \times 3})$$

$$M(n) = Median(W_{3 \times 3})$$

Reemplazar los valores de la matriz por $A(n)$ y $M(n)$ en su valor central.

Respuesta

Lo que hace el algoritmo es primero leer una matriz desde un archivo de texto y transformada en un arreglo dinámico.

Una vez leída y creada la matriz esta es mandada a una función llamada moving_3_3() la cual recorre la matriz mediante la emulación de una matriz de 3 x 3, el recorrido total de la matriz es hasta el elemento $n-3$ de la matriz original figura (9). Podemos observar en la figura (7) y en la figura (8) como es

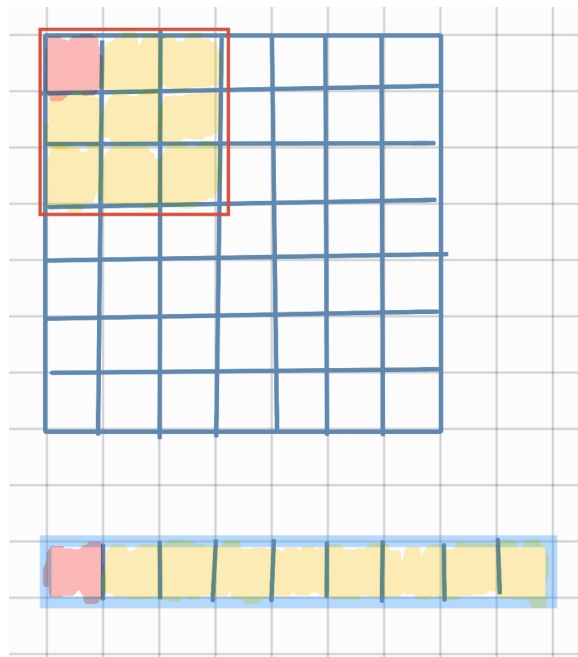


Figure 8: Resultado

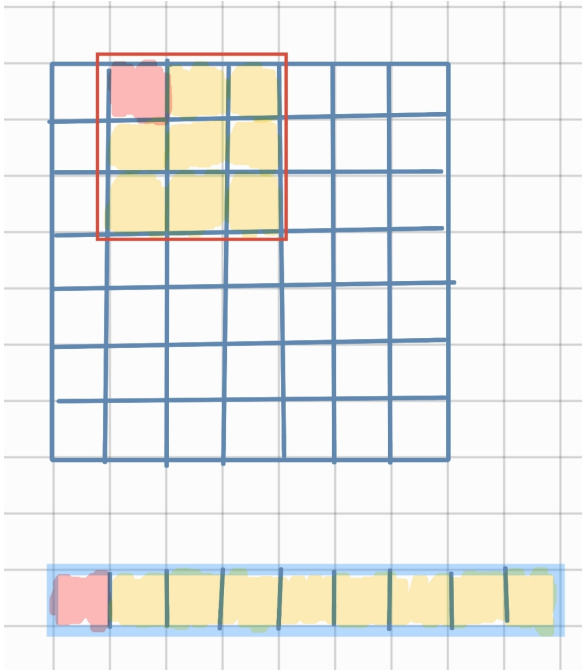


Figure 9: Recorrido total matriz 7x7

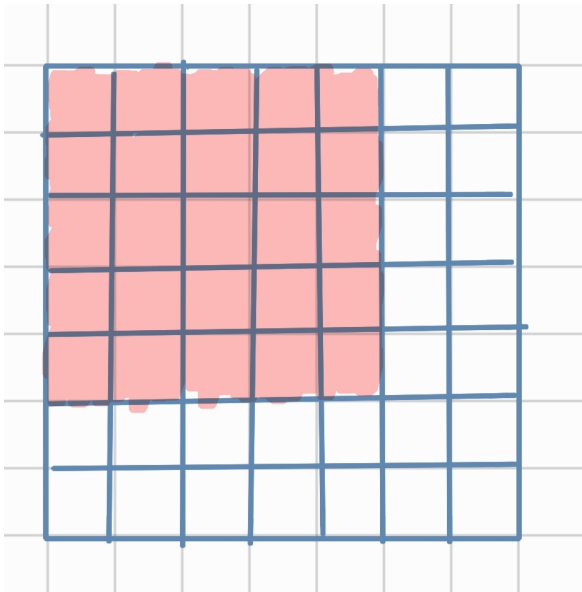
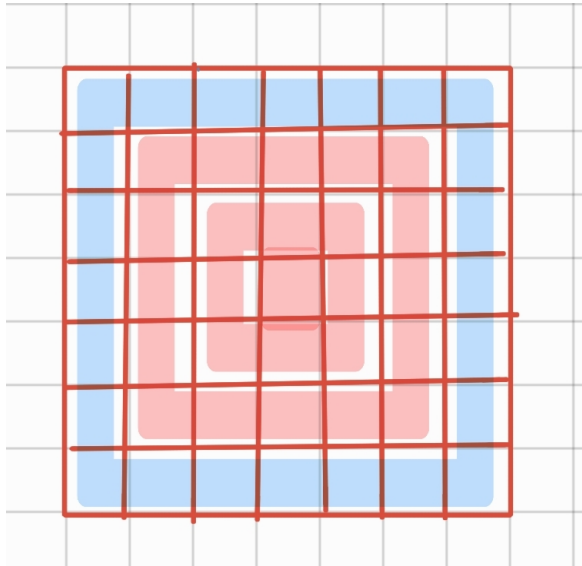
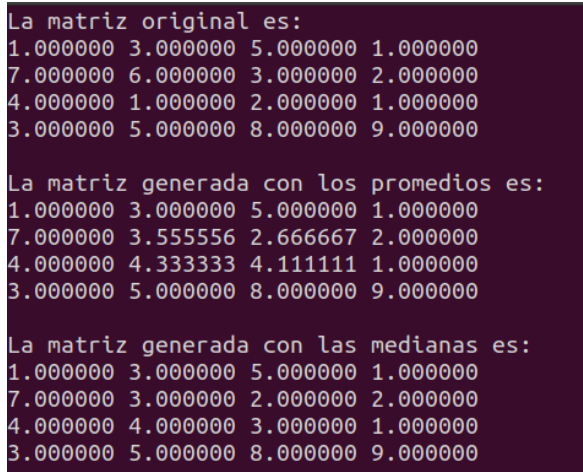


Figure 10: Nueva matriz, nuevos elementos recuadros rojos



En seguida se muestran los resultados obtenidos.

Figure 11: Resultado



Los resultados se compararon con los valores de la matriz mostrada de una manera manual, y corresponden los resultados manuales como los hechos con el programa.