

Laboratorio 3

Josué Herrera Martinez B63440

3 octubre del 2024

1. Preguntas

Ejercicio 1

Escriba un programa que calcule la suma de las diagonales de una matriz cuadrada. Puede partir de una matriz ya inicializada en el código. Sin embargo, su lógica deberá aplicar para matrices cualquier tamaño. Ejemplifique en su reporte al menos 3 casos diferentes.

Como parte de discusión en su reporte, agregue el pseudocódigo de su algoritmo y la explicación del mismo.

local matriz=("!1"): Esta línea crea un array local matriz desde los elementos del array pasado como primer argumento 1. local n=2: Esta línea establece la variable local n al valor del segundo argumento 2, que representa el tamaño de la matriz. local suma=0: Esta línea inicializa una variable local suma a 0, que se utilizará para almacenar la suma de las diagonales. El bucle for itera n veces, y para cada iteración: suma=((suma + matriz[i*n +i] + matriz[(n-i-1)*n +i])) calcula la suma de los elementos en la diagonal principal (de arriba izquierda a abajo derecha) y la diagonal secundaria (de arriba derecha a abajo izquierda). Los índices se calculan utilizando la fórmula $i*n + i$ para la diagonal principal y $(n-i-1)*n + i$ para la diagonal secundaria.

Ejercicio 2

Encuentre y arregle el problema con el siguiente código, cuyo objetivo es calcular el factorial de un número dado. Asegúrese de explicar el cambio realizado en el reporte.

En la función factorial, se devuelve siempre 0 en lugar del valor calculado de la factorial. Debe devolverse el valor de i. En la función factorial, se declara una nueva variable n dentro del bucle while, lo que hace que el bucle se ejecute indefinidamente porque la variable original n no se decrementa. En la función main, se intenta imprimir los valores de fact4 y fact5, pero se han declarado las variables como fac4 y fact. Debe utilizarse el nombre de variable correcto. En la función main, la firma de la función debe ser `int main(int argc, char *argv[])`, no `int main(int argc, char argv[])`.

Modifique el código para que el usuario pueda ingresar el número al que desea calcularle el factorial.

Ejercicio 3

1. Escriba un programa que recorra una matriz cuadrada binaria y determine la cantidad de 1s consecutivos más larga que encuentra, incluyendo los 1s que continúan en la siguiente fila. La matriz tiene un número fijo de filas y columnas. El programa debe devolver la longitud de la secuencia más larga de 1s encontrados.

Explicación del programa

El programa utiliza una función llamada `encontrarSecuenciaMasLarga` que recorre una matriz cuadrada binaria y devuelve la longitud de la secuencia más larga de 1s encontrada. La función utiliza dos variables, `secuenciaMasLarga` y `secuenciaActual`, para mantener la pista de la secuencia más larga encontrada hasta ahora y la secuencia actual que se está procesando.

La función recorre la matriz de dos maneras: primero, recorre la matriz fila por fila para encontrar secuencias de 1s en la misma fila. Luego, recorre la matriz de nuevo para encontrar secuencias que continúen en la siguiente fila.

Si se encuentra una secuencia más larga que la secuencia más larga encontrada hasta ahora, se actualiza la variable `secuenciaMasLarga`. Finalmente, la función devuelve la secuencia más larga encontrada.

En el programa principal, se define una matriz cuadrada binaria y se llama a la función `encontrarSecuenciaMasLarga` para encontrar la secuencia más larga de 1s en la matriz. El resultado se imprime en la consola.

2. Como parte de discusión en su reporte, agregue el pseudocódigo de su diseño y la explicación del mismo.

```

1  PROCEDURE encontrarSecuenciaMasLarga(matriz)
2      secuenciaMasLarga = 0
3      secuenciaActual = 0
4
5      FOR cada fila en matriz
6          FOR cada elemento en fila
7              IF elemento = 1 THEN
8                  secuenciaActual = secuenciaActual + 1
9                  IF secuenciaActual > secuenciaMasLarga THEN
10                     secuenciaMasLarga = secuenciaActual
11             ELSE
12                 secuenciaActual = 0
13
14     FOR cada fila en matriz, excepto la última
15         FOR cada elemento en fila
16             IF elemento = 1 AND elemento de la siguiente fila = 1 THEN
17                 secuenciaActual = secuenciaActual + 1
18                 IF secuenciaActual > secuenciaMasLarga THEN
19                     secuenciaMasLarga = secuenciaActual
20             ELSE
21                 secuenciaActual = 0
22
23     RETURN secuenciaMasLarga

```

Figura 1: Pseudocódigo

El programa utiliza un enfoque de recorrido de matriz para encontrar la secuencia más larga de 1s en una matriz cuadrada binaria. La función `encontrarSecuenciaMasLarga` recibe una matriz como parámetro y devuelve la longitud de la secuencia más larga de 1s encontrada.

La función utiliza dos variables, `secuenciaMasLarga` y `secuenciaActual`, para mantener la pista de la secuencia más larga encontrada hasta ahora y la secuencia actual que se está procesando.

El primer bucle FOR recorre la matriz fila por fila, y para cada elemento, verifica si es un 1. Si es así, incrementa la `secuenciaActual` y verifica si es más larga que la `secuenciaMasLarga` actual. Si lo es, actualiza la `secuenciaMasLarga`. El segundo bucle FOR recorre la matriz de nuevo, pero esta vez, verifica si el elemento actual es un 1 y si el elemento de la siguiente fila también es un 1. Si es así, incrementa la `secuenciaActual` y verifica si es más larga que la `secuenciaMasLarga` actual.

Si lo es, actualiza la `secuenciaMasLarga`.

Finalmente, la función devuelve la `secuenciaMasLarga` encontrada.

3. Extienda el programa para que llene la matriz de manera aleatoria, considerando siempre que sea binaria y cuadrada. Para ello, investigue y utilice la función rand. Asegúrese de ejemplificar esta funcionalidad en su reporte, mostrando diferentes matrices y el resultado de su código.

Link gitHub

<https://github.com/josueherreram/lab3completo.git>