

Práctica 2: Análisis de matrices de datos adquiridas desde archivos de texto

Programación en Ingeniería

3 de mayo de 2020

Fecha de entrega: 18 de Mayo del 2020
Medio de entrega: Electrónico (dora.almanza@ugto.mx)
Profesor: Dra. Dora Luz ALMANZA-OJEDA

1. Objetivo

Utilizar punteros simples y dobles para almacenar datos generados aleatoriamente y leídos desde archivos de texto. Se ocuparán varias funciones que realizarán los cálculos estadísticos correspondientes y recibirán como parámetros de entrada la matriz o regresarán punteros a una matriz según convenga al programador.

2. Instrucciones

Esta práctica se realiza en dos partes:

2.1. PARTE 1: Matrices con datos aleatorios

1) Llenar con datos aleatorios 4 matrices de tamaño 32×32 , a cada matriz se calcularán datos estadísticos como promedio, dato mayor, y promedio de ambos, los resultados por renglón se anexarán en 3 columnas adicionales respectivamente. La figura 1 muestra un ejemplo numérico, considerando una matriz de tamaño 3×4 , los datos en rojo en las columnas 5, 6 y 7 son el promedio de datos por renglón, el número más grande y el promedio de estos dos últimos, respectivamente.

2) Una vez que se generen las 4 matrices con sus columnas de datos estadísticos adicionales, entonces se va a comparar cada matriz con las siguientes. Para ello, genere seis matrices adicionales del mismo tamaño que la matriz 1 y que guardarán el resultado de las comparativas. Realiza una comparativa renglón por renglón entre la matriz 1 y la matriz 2 y registra en una de las matrices resultantes un 1 si en la misma posición en la matriz 1 y 2 hay datos diferentes

MATRIZ 1

Prom Ma Prom

7	1	0	5	3.25	7	5.12
3	11	4	3	5.25	11	8.12
0	23	6	7	9	23	16

Figura 1: Ejemplo de cálculo de las estadísticas por Matriz.

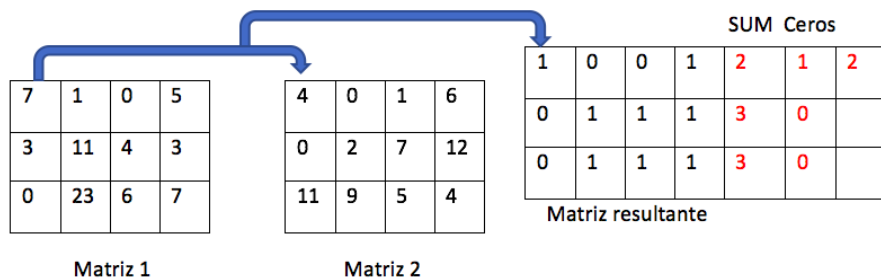


Figura 2: Ejemplo de cálculo de comparativa matriz por matriz, evaluación renglón por renglón.

de cero. Es decir, solo busque por renglón las posiciones en la matriz 1 que están ocupadas y registre en la matriz de salida solo si en la misma posición se encuentra un dato diferente de cero también en la matriz 2. Esta operación es como realizar la operación lógica AND pero con valores decimales. Si al menos una de las dos matrices tiene un cero en la misma posición, entonces en la matriz de salida será un cero. La figura 2 muestra un ejemplo de la comparativa entre dos matrices y como se obtendría la matriz final con ceros y unos.

3) Suma los renglones de la matriz resultante y guarda el resultado de la suma en una columna adicional de la matriz resultante, (primera columna marcada en rojo en la figura 2) y agregue el índice de la columna o columnas que tienen valor de cero, (siguientes columnas en rojo en la figura 2). Después, de la misma forma realice la comparativa entre la matriz 1 y la matriz 3, después, la 1 con la 4, la matriz 2 con la 3, la matriz 2 con la 4 y finalmente la matriz 3 con la 4.

4) Guarde las matrices resultantes de la comparativa en un solo archivo de

texto, de la siguiente manera: el primer renglón debe incluir los renglones y las columnas de la matriz resultante y en el siguiente renglón los datos de la matriz, después de un renglón de espacio vienen los datos del tamaño de la matriz resultante dos y enseguida sus datos, y así sucesivamente para la matriz resultante 3 hasta la 6. El orden de guardar las matrices resultantes es primero la resultante de la comparación matriz 1 con matriz 2, matriz 1 con matriz 3 y así sucesivamente en el orden como se mencionó en el punto 2 para calcularlas. El nombre del archivo con las matrices resultantes es: comparativa.txt.

2.2. PARTE 2: Llenar las matrices con datos leídos desde archivos

En la carpeta Dropbox, junto con esta práctica se anexan 4 archivos de texto, los cuales contienen los datos con los cuales se van a llenar las matrices. El primer dato de cada archivo indica el número de RENGLONES y el segundo renglón indica el número de COLUMNAS que deberá contener la matriz para poder guardar los datos contenidos en el archivo. No permita que el número de renglones y columnas de la matriz que contendrá los datos leídos sea fijo, solo porque parece que siempre son así, ya que para probar sus códigos yo tengo otras matrices adicionales, que no son del mismo tamaño y entonces su programa no va funcionar si estos datos los dejan fijos.

2.3. Funciones a programar:

Programa al menos las siguientes funciones, (si requiere mas funciones será mucho mejor para evitar tanto código en la función main):

- Función para generar espacio de la matriz

- Función que lea 1 archivo de texto de los 4 que se proporcionan y llene 1 matriz con la información proporcionada en el archivo. El tamaño de cada matriz estará indicada en los dos primeros renglones del archivo. Se sugiere que esta función regrese un puntero a la matriz con los datos. Puede programarse una sola función para leer los 4 archivos y generar las 4 matrices, pero ese diseño se debe plantear bien como se regresarían las matrices de datos, porque una función solo puede regresar una dirección y se tienen 4 matrices, es posible de hacer esto, pero deben pensar bien su diseño PERO SIN hacer matrices globales.

- Función para llenar la matriz de datos resultantes de la comparativa entre las matrices y los datos estadísticos.

- Función(es) para calcular los datos estadísticos.

- Función para guardar las matrices resultantes en archivos .txt

- Función para liberar memoria reservada.

Entregables:

1.- Únicamente uno o dos archivos .c o .cpp MAXIMO, con el main del programa y otro archivo con las funciones correspondientes, en caso de que no quieran escribirlas en main.