**Universidad de Costa Rica**
**CI-0116: Análisis de Algoritmos y Estructuras de Datos**
**Laboratorio #1**

**Objetivo:** Familiarizar al estudiante en la implementación de los métodos estudiados y los análisis de tiempos de duración de algoritmos

**Enunciado**

Resuelva los siguientes problemas. Escriba un código en C++ que resuelva los siguientes problemas. Además, previo a cada función indique cuál considera que sería la cota O de dicho algoritmo como comentario y de considerarlo necesario una explicación de su razonamiento.

1. El algoritmo de ordenamiento por mezcla (merge sort) es uno de los algoritmos de ordenamiento más rápidos de hoy en día. Implemente dicho algoritmo para que reciba un arreglo de elementos (de cualquier tipo) y los ordene de menor a mayor.

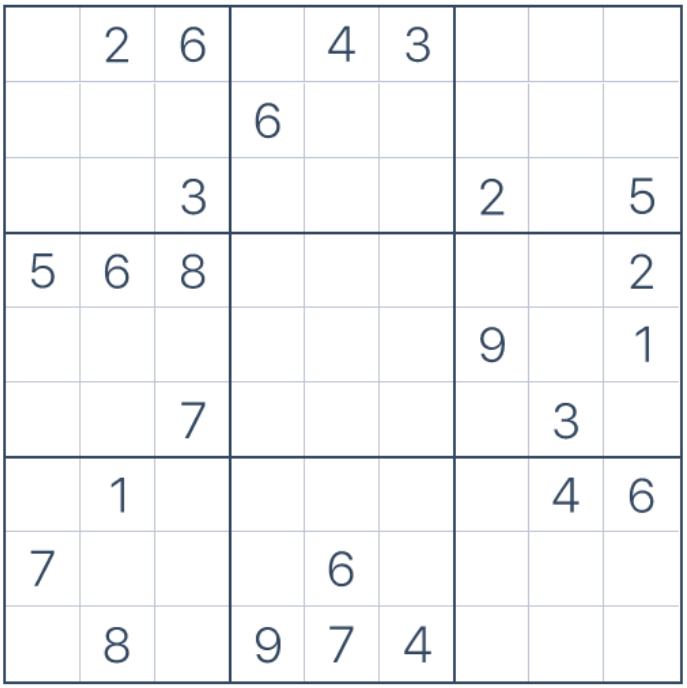
   Templates

2. El algoritmo de multiplicación de matrices especifica que para poder multiplicar dos matrices ellas deben ser de tamaño "n x m" y "m x p" respectivamente, coincidiendo así en el valor 'm'. Para multiplicar matrices deben tomarse cada una de las filas de la primera matriz y multiplicarlas uno a uno con cada una de las columnas de la segunda matriz. Cada una de esas multiplicaciones es en realidad un producto punto: los valores se multiplican y el resultado de dichas multiplicaciones se suma. Por cada multiplicación de una fila por una columna se obtiene un nuevo valor correspondiente al valor de la nueva matriz en el la posición (fila,columna). Implemente una función de C++ que reciba tres matrices de números reales (float** input1, input2, output) y sus respectivos valores n, m y p y realice la multiplicación de matrices.
   a. Ejemplo de multiplicación de matrices:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | X | 1 | 0 | 1 | = | 1*1 + 2*2 | 1*0 + 2*1 | 1*1 + 2*0 | = | 5 | 2 | 1 |
| 3 | 1 | | 2 | 1 | 0 | | 3*1 + 1*2 | 3*0 + 1*1 | 3*1 + 1*0 | | 5 | 1 | 3 |

3. Escriba una función recursiva que permita solucionar el juego sudoku.
   a. Los sudokus se recibirán como una matriz de enteros de 9x9 dónde los espacios en blanco se representarán con 0. Las reglas normales de Sudoku aplican: en una misma fila, columna o sub-cuadrante (de 3x3) no se vale repetir ningún número. Solo se vale utilizar números del 1 a 9.

b. Ejemplo:



Se podría declarar en C++ como: int sudoku[9][9] = {{0,2,6,0,4,3,0,0,0},{0,0,0,6,0,0,0,0,0},{0,0,3,0,0,0,2,0,5},{5,6,8,0,0,0,0,0,2},{0,0,0,0,0,0,9,0,1},{0,0,7,0,0,0,0,3,0},{0,1,0,0,0,0,0,4,6},{7,0,0,0,6,0,0,0,0},{0,8,0,9,7,4,0,0,0}}

Y tiene solución:

9 2 6 5 4 3 8 1 7
8 5 1 6 2 7 3 9 4
4 7 3 1 9 8 2 6 5
5 6 8 3 1 9 4 7 2
3 4 2 7 8 6 9 5 1
1 9 7 4 5 2 6 3 8
2 1 9 8 3 5 7 4 6
7 3 4 2 6 1 5 8 9
6 8 5 9 7 4 1 2 3