

# Estructuras de datos

Área personal / Mis cursos / (RG-999-1656-AP)-Estructuras de datos / Secciones / Exámenes Parciales / PARCIAL 1: AA-ORD-BUSQ-AVL

## PARCIAL 1: AA-ORD-BUSQ-AVL

1) **Análisis de algoritmos.** Dado el siguiente programa con 3 métodos y un main, se pide obtener mediante las notaciones asintóticas, el T(n) e indicar su tipo de complejidad. La solución debe estar acompañado de su desarrollo matemático.

```
void inicializarMatriz (int[][] matriz, int N) {
    int i, j;
    for (i=0; i < N; i++)
        for (j=0; j < N; j++)
            if( (i+j) % 2 == 0)
                matriz[ i ][ j ] = i;
            else
                matriz[ i ][ j ] = i + j;
}

void mostrarMatriz (int[][] matriz, int N) {
    int i, j;
    System.out.println("Matriz: ");
    for (i=0; i < N; i++) {
        for (j = 0; j < N; j++)
            System.out.print(matriz[ i ][ j ]);
        System.out.println(" ");
    }
}

void funcionMagica (int matriz [][][10], int N) {
    int i, j;
    for (i = 1; i < N - 1; i++)
        for (j = i; j < N - 1; j++)
            matriz[ i ][ j ] = 0;
}

// programa principal
public static void main(String[] args){
    int[][] matriz=new int[10][10];
    inicializarMatriz(matriz, 10);
    mostrarMatriz(matriz, 10);
    funcionMagica(matriz, 10);
    mostrarMatriz(matriz, 10);
}
```

2) **Ordenamiento.** Un array contiene los siguientes elementos: 40, 5, 20, 45, 30, 15, 25, 10, 35, 0. Utilizando el algoritmo de ordenación Quicksort, encuentre las pasadas y los intercambios que se realizan para su ordenación. Como pivote se elige el elemento central (opcional) de la lista. Implementarlo en código Java, y de manera manual (opcional).

3) **Búsqueda.** Aplicar **Hashing** para guardar y buscar Automóviles (cuya clave principal es la patente). Utilizar una función hash que se adecue al tipo de problema. El método de resolución de colisiones a utilizar será algunos de los siguientes: Prueba Cuadrática - Rehashing - Doble Direccionamiento Hash. Indicar que método Hash, y de resolución de colisiones fueron aplicados. Realizar la implementación correspondiente en código Java.

4) **AVL.** Dibujar el árbol AVL que resulta de insertar las siguientes claves 18, 23, 11, 10, 20, 26, 5, 12, 15, 9, 7, 16, y 4. Posteriormente, realizar las siguientes eliminaciones: 18, 23, 26, 15, y 16. Graficar paso a paso las inserciones y/o eliminaciones, indicando que tipo de rotación se utiliza para cada caso.

### Estado de la entrega

Estado de la entrega	Enviado para calificar
Estado de la calificación	Sin calificar
Fecha de entrega	viernes, 18 de octubre de 2024, 20:25
Tiempo restante	La tarea fue enviada 7 minutos 12 segundos antes de la fecha límite
Última modificación	viernes, 18 de octubre de 2024, 20:17
Archivos enviados	<div><div></div><div>Parcial 1 2024 Gonzalo Ulloa.rar</div><div>+</div></div> <div>Exportar al portafolios</div>
Comentarios de la entrega	<div><div></div><div>Comentarios (0)</div></div>

Editar entrega

Realizar cambios en la entrega

### Comentario

Comentarios de retroalimentación			
	TEMAS	RESULTADO	COMENTARIOS
	AA	B	Muy bien los planteos de cada AA de los metodos, tuvo pequeño error de constantes en primer metodo, correcto e mal planteada la funcion; muy bien la la aplicacion de las propiedades de la sumatoria. De todos modos, muy bien
	ORDENAMIENTO	MB	Muy bien el desarrollo del ejercicio, y completo.
	BUSQUEDA	B	Bien desarrollo del ejercicio, separa las clases de manera correcta. Falto implementar un metodo para buscar un e
	AVL	B	Bien descripto el paso a paso, solo falto colocar los FE para ver donde no se cumple la condicion. Correcta la oper todos modos, ok.
	Corregido por Mg. Héctor REINAGA		

BIBLIOTECAS DIGITALES



OTROS RECURSOS

