

Apellido y nombre: _____

Padrón: _____

Nota final:				

1) Explique qué es el **Teorema Maestro** y cómo se utiliza. Escriba las ecuaciones de recurrencia que representan a los algoritmos de **Mergesort** y **Quicksort**. Explique si estas ecuaciones pueden o no ser resueltas mediante el uso del **Teorema Maestro**. En caso negativo, explique de qué otra forma se puede llegar a un resultado.

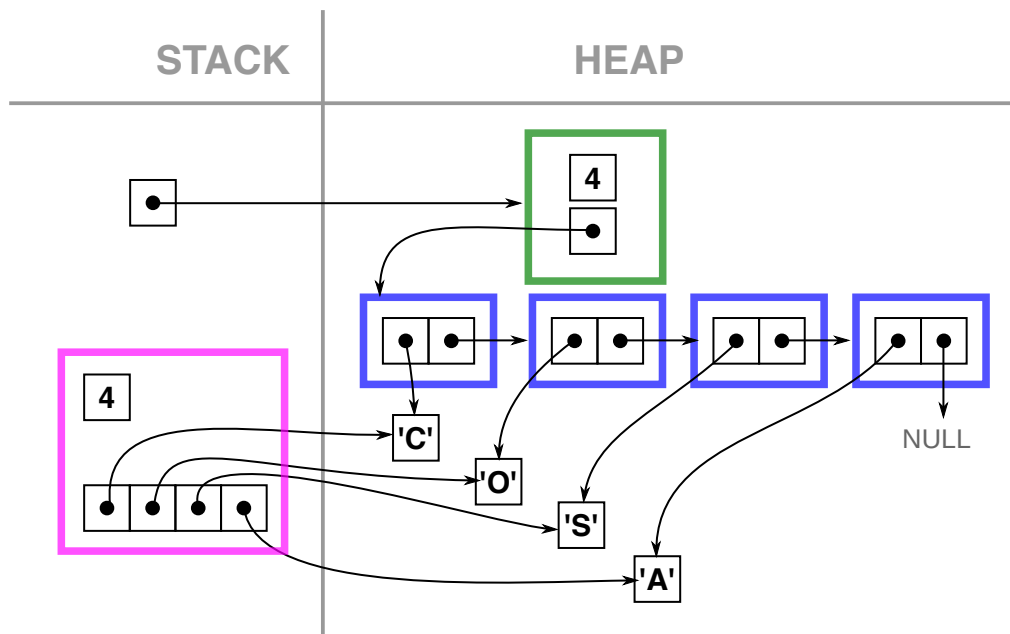
2) Inserte los siguientes elementos en orden de aparición en un árbol **AVL**. Muestre el paso a paso. ¿Cómo cambia el resultado final con un **ABB** y por qué es importante la distinción?

Elementos=[A,B,C,G,D,E,F]

3) Dados el vector [73, 61, 88, 15, 54, 43, 31, 22, 10], ordenarlo de mayor a menor utilizando **mergesort** de 3 vías. Muestre cada paso del algoritmo y calcule su complejidad.

4) Escriba un programa en C (definiendo las variables, estructuras y tipos que crea conveniente) de forma tal que el uso de memoria del mismo sea como el que se muestra a continuación (puede agregar variables o punteros auxiliares si es necesario). Muestre el código que hace que dicho programa libere correctamente toda la memoria reservada.

Recuerde que para este ejercicio es fundamental que se haga la correcta verificación de las operaciones de memoria dinámica y de acceso a los punteros. Si el uso de memoria es incorrecto, el ejercicio queda invalidado. En la imagen los elementos que se muestran pegados son elementos contiguos en memoria (vectores) y los recuadros de colores estructuras de datos.



5) Escriba (sin utilizar **for**, **while**, **do**) de manera recursiva, un algoritmo que dado un string, devuelva la suma de los caracteres en posiciones pares (0 cuenta como posición par en este caso). No se permite el uso de **strlen** o similares. No se permite el uso de funciones auxiliares.