Ejercicio 1

Explique qué es el **Teorema Maestro**, cómo se aplica y cuáles son sus casos.

Escriba un algoritmo que mediante *división y conquista* encuentre en un vector desordenado de números el de menor valor. Justifique su complejidad.

```
int buscar_menor(int* vector, int tamaño);
```

Ejercicio 2

Explique qué es un **árbol B**, cómo se diferencia de árboles binarios de búsqueda y qué características tiene.

Inserte (en un arbol B de 3 claves por nodo), los elementos **J,L,A,B,T,M,P,R** y luego elimine los elementos **L** y **M**.

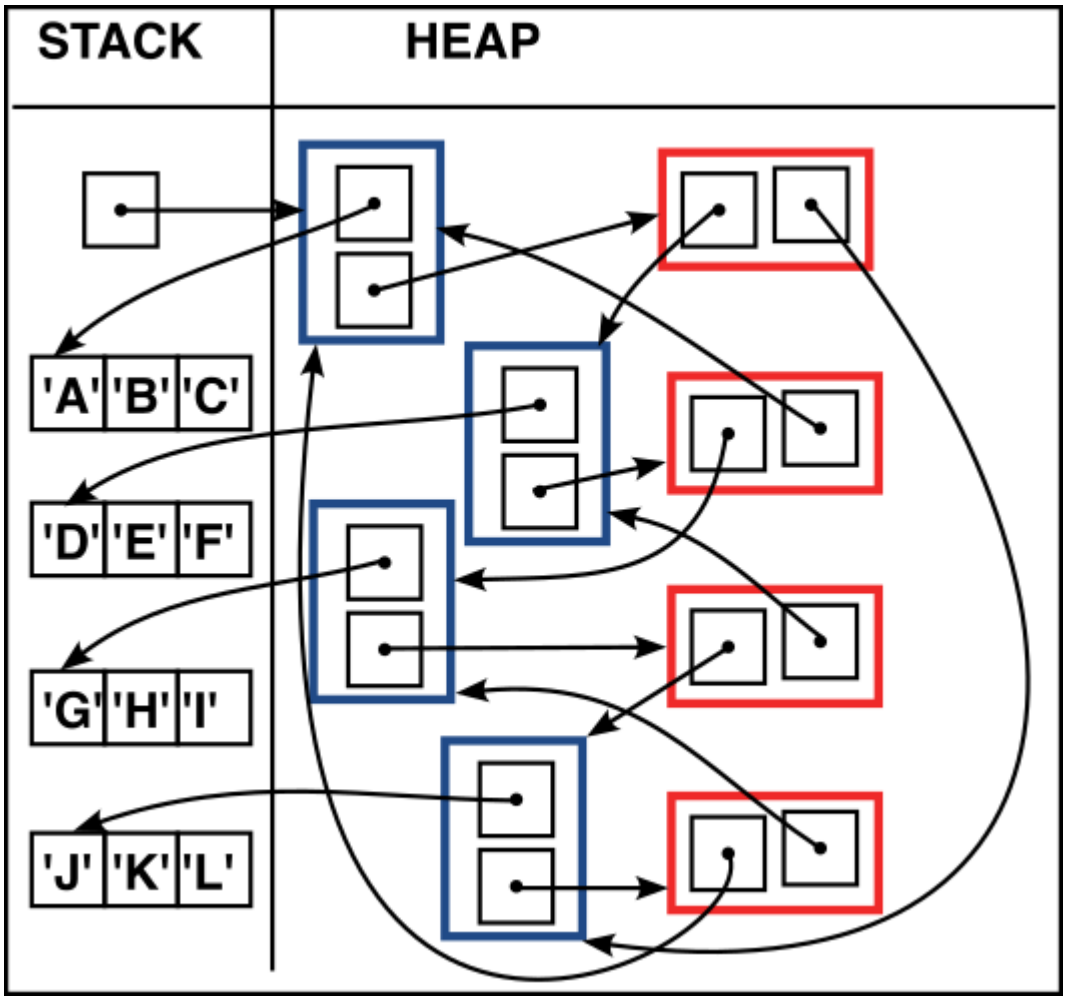
Dibuje el estado del árbol luego de cada operación y explique cómo se realiza.

Ejercicio 3

Escriba un programa en `C` que permita armar en memoria las estructuras presentadas en la imagen. Se pueden utilizar variables auxiliares extra en el stack si es necesario.

Agregue también el código para **liberar toda la memoria de forma correcta**. Notese que en el esquema las estructuras se representan con recuardos de colores y los punteros con cuadrados con un punto en el centro. Por otro lado recuerde que los elementos que se encuentran pegados físicamente (horizontal o vertical) representan datos contiguos en memoria, mientras que los elementos separados no.

ATENCION: Este ejercicio evalúa el uso de punteros y memoria dinámica. **Es fundamental que los accesos a memoria sean correctos para que el ejercicio esté aprobado**.



Ejercicio 4

Justifique si las siguientes afirmaciones son verdaderas o falsas

- Un **árbol Rojo/Negro** es mas eficiente que uno **AVL** para insertar, eliminar y buscar datos
- Un algoritmo de ordenamiento nunca puede ser mas eficiente en tiempo que **n*log(n)**
- Insertar al final en una lista enlazada (*sin referencia al último nodo*) y en un vector dinámico tienen la misma complejidad.
- La complejidad de un *TDA* depende de la implementación

Ejercicio 5

Escriba una función recursiva (no se permite utilizar **for**, **while**, **etc**) que dado un abb, devuelva el dato almacenado en la hoja mas cercana a la raíz (la primera que encuentre si existe mas de una).

Para este ejercicio puede suponer que las operaciones de memoria no fallan y si lo cree necesario puede utilizar todos los **TDAs** implementados en la materia.