Algoritmos 2, Curso Mendez ~ 2do Recuperatorio, 2do Cuatrimestre 2023 ~ 2023-12-14

Apellido y nombre: _____

Padrón: _____

Nota final:				

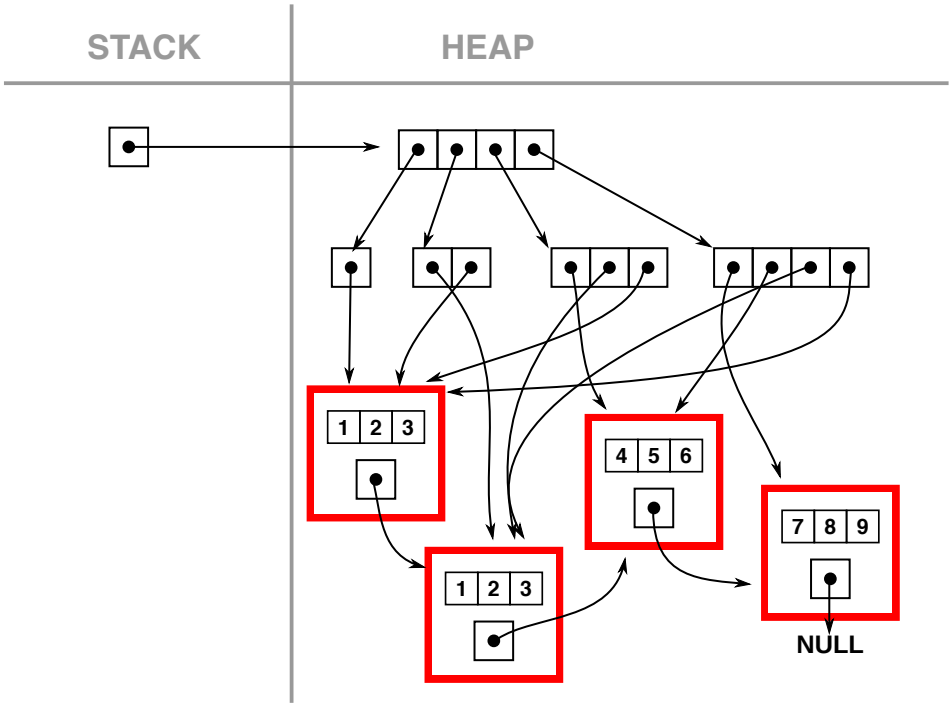
1) Explique qué es el **Teorema Maestro**, cómo se aplica y para qué lo utilizamos. Dada la ecuación de recurrencia **T(n) = 3 T(n-5) + O(n)**, calcule la complejidad de dicho algoritmo. Escriba en C99 una función que se corresponda con dicha ecuación. Justifique. Si falta información puede hacer suposiciones.

2) Dado el vector de valores **V=[5,8,1,2,5,3,9]**, muestre (paso a paso) cómo se ordena este vector de menor a mayor (in-place) aplicando heapsort.

3) Explique cómo funciona un árbol AVL. Inserte los elementos **'P', 'I', 'K', 'A', 'C', 'H', 'U'** (en ese orden). Muestre y justifique cada paso.

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, una función que prueba contraseñas numéricas de forma secuencial por fuerza bruta (empezando por **"0000"** y finalizando en **"9999"**) y devuelve la cantidad de intentos requeridos para adivinar la contraseña (o -1 si no se encuentra). La función **es_correcta** es provista por el usuario y devuelve **true** si la contraseña es correcta o **false** en caso contrario.

unsigned adivinar(bool (*es_correcta)(char*)) { ??? }