Algoritmos 2, Curso Mendez ~ 2do Recuperatorio, 2do Cuatrimestre 2024 ~ 2024-12-12

Apellido y nombre: _____

Padrón: _____

Nota final:				

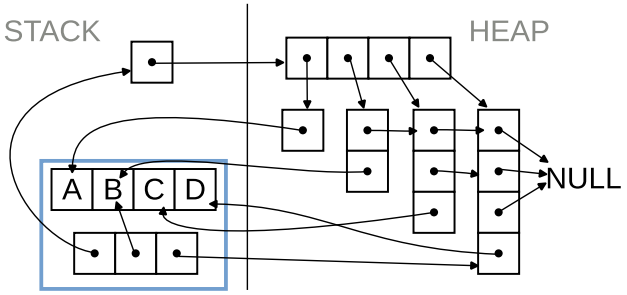
1) Dado un vector casi ordenado de menor a mayor (sólo 1 elemento fuera de orden), escriba en **C99** un algoritmo del tipo **divide y conquista** que encuentre el único elemento desordenado. Explique cómo funciona y cómo calcular su complejidad.

2) Dados los siguientes recorridos de un mismo **ABB**, defina un algoritmo que reconstruya el **ABB original**. Muestre el árbol final y explique el procedimiento justificando el resultado.
Preorden = [2,5,1,8,4,3,0,9] **Inorden** = [5,2,4,8,1,0,9,3]

3) Explique cómo funciona el algoritmos **Heapsort**. Dado el vector **V=[5,3,7,9,1,0,2]**, aplique el algoritmo para que quede ordenado de **mayor a menor**. El ordenamiento debe realizarse in-place (no se permite utilizar un vector auxiliar). Muestre cada paso del ordenamiento y explique su complejidad.

4) Explique cuál es la diferencia entre utilizar una matriz estática y una de memoria dinámica en **C99**. Muestre dos formas diferentes de armar una matriz con memoria dinámica. Compare las 3 formas (acceso a los elementos, almacenamiento en memoria, sintaxis de uso, etc).

5) Escriba un programa en **C99** que permita armar en memoria las estructuras presentadas en el diagrama. Se pueden utilizar variables auxiliares extra en el stack. Agregue también el código para liberar **toda la memoria de forma correcta**. 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**.



Algoritmos 2, Curso Mendez ~ 2do Recuperatorio, 2do Cuatrimestre 2024 ~ 2024-12-12

Apellido y nombre: _____

Padrón: _____

Nota final:				

1) Dado un vector casi ordenado de menor a mayor (sólo 1 elemento fuera de orden), escriba en **C99** un algoritmo del tipo **divide y conquista** que encuentre el único elemento desordenado. Explique cómo funciona y cómo calcular su complejidad.

2) Dados los siguientes recorridos de un mismo **ABB**, defina un algoritmo que reconstruya el **ABB original**. Muestre el árbol final y explique el procedimiento justificando el resultado.
Preorden = [2,5,1,8,4,3,0,9] **Inorden** = [5,2,4,8,1,0,9,3]

3) Explique cómo funciona el algoritmos **Heapsort**. Dado el vector **V=[5,3,7,9,1,0,2]**, aplique el algoritmo para que quede ordenado de **mayor a menor**. El ordenamiento debe realizarse in-place (no se permite utilizar un vector auxiliar). Muestre cada paso del ordenamiento y explique su complejidad.

4) Explique cuál es la diferencia entre utilizar una matriz estática y una de memoria dinámica en **C99**. Muestre dos formas diferentes de armar una matriz con memoria dinámica. Compare las 3 formas (acceso a los elementos, almacenamiento en memoria, sintaxis de uso, etc).

5) Escriba un programa en **C99** que permita armar en memoria las estructuras presentadas en el diagrama. Se pueden utilizar variables auxiliares extra en el stack. Agregue también el código para liberar **toda la memoria de forma correcta**. 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**.