Algoritmos 2, Curso Mendez ~ 2do recuperatorio, 1er Cuatrimestre 2025 ~ 2025-07-03

Apellido y nombre: _____

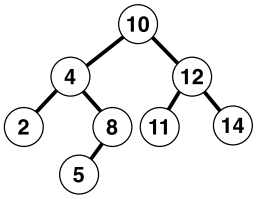
Padrón: _____ Modalidad: Completo / Reducido

Nota final:				

1) Ordene el siguiente vector de menor a mayor utilizando **QuickSort**. Muestre cada paso del algoritmo. Justifique la complejidad. Explique qué cuidado hay que tener para al aplicar el **Teorema Maestro** y por qué estas consideraciones no son necesarias en el caso de **Mergesort**. **V = [6,4,2,9,8,1,7,3]** .

2) Explique cómo funciona un árbol B y qué características lo definen. En un árbol **B** con 3 claves por nodo, inserve los siguientes elementos en el orden dado: '**M**', '**A**', '**L**', '**T**', '**G**', '**R**', '**I**', '**O**', '**S**'. Luego elimine '**M**' y '**R**'. Muestre el estado del árbol en cada paso.

3) Explique qué es un árbol AVL. Justifique en qué orden debieron ser insertados los elementos para lograr el AVL presentado si se sabe que se tuvieron que realizar dos rotaciones dobles opuestas.



4) Explique qué es un error en tiempo de ejecución y cómo se diferencian de los errores en tiempo de compilación. Para los siguientes errores, explique qué significan, sus consecuencias y muestre un ejemplo: **a)** Lectura inválida, **b**) Escritura inválida, **c)** Double free, **d)** Salto condicional no inicializado

5) El algoritmo **Babilónico** para calcular la raíz cuadrada de **n** consiste en calcular una estimación $\mathbf{n_0 = n/2}$ y luego refinarla iterativamente calculando $\mathbf{n_i = (n_{i-1} + n/n_{i-1})/2}$. Esto se repite hasta que ni se acerque a la respuesta correcta lo suficiente. Implementar el algoritmo de manera recursiva en **C99** o **Python** y justificar su complejidad.

Algoritmos 2, Curso Mendez ~ 2do recuperatorio, 1er Cuatrimestre 2025 ~ 2025-07-03

Apellido y nombre: _____

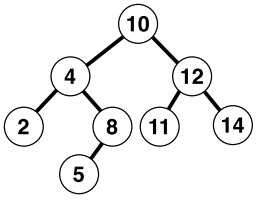
Padrón: _____ Modalidad: Completo / Reducido

Nota final:				

1) Ordene el siguiente vector de menor a mayor utilizando **QuickSort**. Muestre cada paso del algoritmo. Justifique la complejidad. Explique qué cuidado hay que tener para al aplicar el **Teorema Maestro** y por qué estas consideraciones no son necesarias en el caso de **Mergesort**. **V = [6,4,2,9,8,1,7,3]** .

2) Explique cómo funciona un árbol B y qué características lo definen. En un árbol **B** con 3 claves por nodo, inserve los siguientes elementos en el orden dado: '**M**', '**A**', '**L**', '**T**', '**G**', '**R**', '**I**', '**O**', '**S**'. Luego elimine '**M**' y '**R**'. Muestre el estado del árbol en cada paso.

3) Explique qué es un árbol AVL. Justifique en qué orden debieron ser insertados los elementos para lograr el AVL presentado si se sabe que se tuvieron que realizar dos rotaciones dobles opuestas.



4) Explique qué es un error en tiempo de ejecución y cómo se diferencian de los errores en tiempo de compilación. Para los siguientes errores, explique qué significan, sus consecuencias y muestre un ejemplo: **a)** Lectura inválida, **b**) Escritura inválida, **c)** Double free, **d)** Salto condicional no inicializado

5) El algoritmo **Babilónico** para calcular la raíz cuadrada de **n** consiste en calcular una estimación $\mathbf{n_0 = n/2}$ y luego refinarla iterativamente calculando $\mathbf{n_i = (n_{i-1} + n/n_{i-1})/2}$. Esto se repite hasta que ni se acerque a la respuesta correcta lo suficiente. Implementar el algoritmo de manera recursiva en **C99** o **Python** y justificar su complejidad.