Algoritmos 2, Curso Mendez ~ 5to Final, 2do Cuatrimestre 2024 ~ 2025-02-20

Apellido y nombre: _____

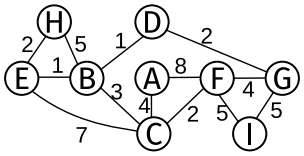
Padrón: _____ Modalidad: Completo / Reducido

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) 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 = [1,7,4,3,9,5,2,8]**

3) Explique qué es el algoritmo de **Dijsktra** y para qué sirve. Aplíquelo al siguiente grafo mostrando el resultado de cada paso y el resultado final comenzando desde **H**.



4) Explique qué es un **árbol**. De un ejemplo. Escriba (en **C99** o **Python**) un algoritmo que permita detectar si un grafo es un árbol o no. Muestre de qué forma debe estar representado el grafo para su algoritmo. Explique y muestre cómo funciona.

5) Explique qué es un **diccionario** y cómo difiere de una **tabla de hash**. Escriba en (**C99** o **Python**) el código para insertar un elemento en una tabla de hash con **direccionamiento abierto**. Explique los puntos importantes de esta función y cómo funciona.

Algoritmos 2, Curso Mendez ~ 5to Final, 2do Cuatrimestre 2024 ~ 2025-02-20

Apellido y nombre: _____

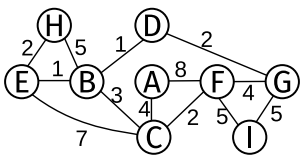
Padrón: _____ Modalidad: Completo / Reducido

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) 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 = [1,7,4,3,9,5,2,8]**

3) Explique qué es el algoritmo de **Dijsktra** y para qué sirve. Aplíquelo al siguiente grafo mostrando el resultado de cada paso y el resultado final comenzando desde **H**.



4) Explique qué es un **árbol**. De un ejemplo. Escriba (en **C99** o **Python**) un algoritmo que permita detectar si un grafo es un árbol o no. Muestre de qué forma debe estar representado el grafo para su algoritmo. Explique y muestre cómo funciona.

5) Explique qué es un **diccionario** y cómo difiere de una **tabla de hash**. Escriba en (**C99** o **Python**) el código para insertar un elemento en una tabla de hash con **direccionamiento abierto**. Explique los puntos importantes de esta función y cómo funciona.