

Algoritmos 2, Curso Mendez ~ 1er Final, 2do Cuatrimestre 2024 ~ 2024-12-12

Apellido y nombre: _____

Padrón: _____ Modalidad: Completo / Reducido

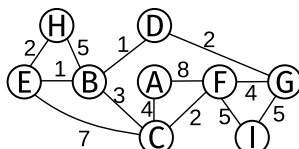
Nota final:				

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) 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 para qué sirve y cómo funcionan el algoritmo de **Dijkstra**. Muestre cómo se aplica paso a paso al siguiente grafo desde **A**.



4) Escriba una función (en **C99** o **Python**) que dado un grafo devuelva un posible orden topológico del mismo. Explique cómo funciona y dé un ejemplo.

5) a) Explique qué es un diccionario, qué es una colisión y de qué maneras se pueden resolver.

b) Explique por qué en algunos diccionarios es necesario utilizar métodos de resolución de colisiones y en otros no. Dar ejemplos.

c) Muestre cómo quedaría almacenado en memoria el grafo del punto 3 en un diccionario (el tipo de diccionario que le parezca mas fácil de dibujar). Para el tipo de diccionario utilizado, explique la complejidad de acceder a los vecinos de un vértice del grafo.

Algoritmos 2, Curso Mendez ~ 1er Final, 2do Cuatrimestre 2024 ~ 2024-12-12

Apellido y nombre: _____

Padrón: _____ Modalidad: Completo / Reducido

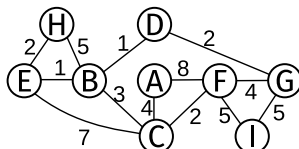
Nota final:				

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) 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 para qué sirve y cómo funcionan el algoritmo de **Dijkstra**. Muestre cómo se aplica paso a paso al siguiente grafo desde **A**.



4) Escriba una función (en **C99** o **Python**) que dado un grafo devuelva un posible orden topológico del mismo. Explique cómo funciona y dé un ejemplo.

5) a) Explique qué es un diccionario, qué es una colisión y de qué maneras se pueden resolver.

b) Explique por qué en algunos diccionarios es necesario utilizar métodos de resolución de colisiones y en otros no. Dar ejemplos.

c) Muestre cómo quedaría almacenado en memoria el grafo del punto 3 en un diccionario (el tipo de diccionario que le parezca mas fácil de dibujar). Para el tipo de diccionario utilizado, explique la complejidad de acceder a los vecinos de un vértice del grafo.