

Algoritmos 2, Curso Mendez ~ 3er Final, 1er Cuatrimestre 2024 ~ 2024-07-18

Apellido y nombre: _____

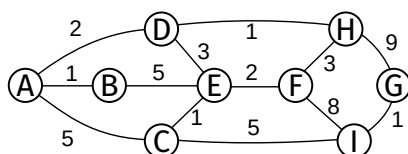
Padrón: _____ Modalidad: Completo / Reducido

Nota final:				

1) El algoritmo **Babilónico** para el cálculo de la raíz cuadrada de x consiste en seleccionar una estimación inicial de la raíz a calcular (ej: $e=x/2$) y luego iterativamente verificar si la diferencia entre $e \cdot e$ y el numero original difieren a lo sumo un valor máximo **PRECISION**. En cada iteración se vuelve a calcular una nueva estimación de la forma $nueva_e=(e+x/e)/2$. Implemente este algoritmo en **C99** sin utilizar **do**, **while**, **for**, etc.

2) Explique (**con diagramas**) cómo funciona **heapsort**. Muestre paso a paso cómo aplicar el algoritmo de **heapsort** al siguiente vector de forma in-place $V=[6,8,3,1,0,9,2,5,4]$. El vector debe quedar ordenado de mayor a menor.

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



4) Explique qué es un recorrido **BFS**. Suponga que tiene a disposición un **TDA Grafo**. Explique cómo está implementado este grafo (explique la estructura) e indique qué operaciones necesitaría tener implementadas para poder implementar fácilmente un recorrido **BFS**. Implemente el algoritmo **BFS** (suponga que las operaciones de manejo de memoria no fallan nunca). Puede utilizar los **TDAS** implementados en la materia si los necesita.

5) Explique qué es una **tabla de hash cerrada**. Explique por qué y en qué casos en este tipo de tablas la eliminación de elementos puede llegar a afectar al resto de las operaciones. Ejemplifique con diagramas y explique alguna posible solución al problema.

Algoritmos 2, Curso Mendez ~ 3er Final, 1er Cuatrimestre 2024 ~ 2024-07-18

Apellido y nombre: _____

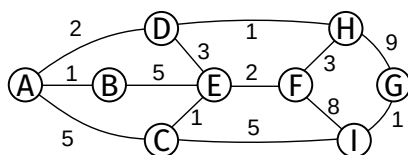
Padrón: _____ Modalidad: Completo / Reducido

Nota final:				

1) El algoritmo **Babilónico** para el cálculo de la raíz cuadrada de x consiste en seleccionar una estimación inicial de la raíz a calcular (ej: $e=x/2$) y luego iterativamente verificar si la diferencia entre $e \cdot e$ y el numero original difieren a lo sumo un valor máximo **PRECISION**. En cada iteración se vuelve a calcular una nueva estimación de la forma $nueva_e=(e+x/e)/2$. Implemente este algoritmo en **C99** sin utilizar **do**, **while**, **for**, etc.

2) Explique (**con diagramas**) cómo funciona **heapsort**. Muestre paso a paso cómo aplicar el algoritmo de **heapsort** al siguiente vector de forma in-place $V=[6,8,3,1,0,9,2,5,4]$. El vector debe quedar ordenado de mayor a menor.

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



4) Explique qué es un recorrido **BFS**. Suponga que tiene a disposición un **TDA Grafo**. Explique cómo está implementado este grafo (explique la estructura) e indique qué operaciones necesitaría tener implementadas para poder implementar fácilmente un recorrido **BFS**. Implemente el algoritmo **BFS** (suponga que las operaciones de manejo de memoria no fallan nunca). Puede utilizar los **TDAS** implementados en la materia si los necesita.

5) Explique qué es una **tabla de hash cerrada**. Explique por qué y en qué casos en este tipo de tablas la eliminación de elementos puede llegar a afectar al resto de las operaciones. Ejemplifique con diagramas y explique alguna posible solución al problema.