

ALGORITMOS Y ESTRUCTURAS DE DATOS AVANZADAS

Take Home / 4-OCT-2022

Fecha de entrega: 18-OCT-2022.

1. Dados n números enteros donde se sabe que hay a lo sumo $\frac{n}{\log n}$ de ellos caen fuera del rango $[0, 2n]$.
 - (a) Dar un algoritmo eficiente para ordenarlos de menor a mayor.
 - (b) Mostrar la correctitud y determinar la complejidad del algoritmo propuesto.
2. Dados n números, se quieren determinar los 3 cuartiles de ellos. El primer cuartil es la mediana de la mitad de los números más chicos, el segundo es la mediana de todos los números y el tercer cuartil es la mediana de la mitad de los números mayores. Dar algoritmos eficientes (randomizado y no randomizado) para resolver este problema.
3.
 - (a) Buscar al menos tres problemas conocidos donde el uso de Fibonacci Heap permite reducir las complejidades de los algoritmos que los resuelven (en comparación con la implementación natural del algoritmo y la implementación usando binary heap). Para cada uno, explicar cómo sería el algoritmo final que implementa Fibonacci Heap y justificar la mejora de complejidad.
 - (b) Indicar cuáles de las operaciones de Fibonacci Heap donde su costo amortizado es aceptable pero puede no serlo si consideramos únicamente la operación en sí (el peor caso). Buscar estructura alternativa que pueda solventar este inconveniente.
4.
 - (a) Mostrar y explicar la complejidad espacial de los árboles de Emde Boas.
 - (b) Dar las modificaciones necesarias para reducir dicha complejidad y explicar cómo impactarían en las operaciones.
5. Definimos la siguiente variante del problema de superstring de talle mínimo: dado un conjunto de strings $S = \{s_1, s_2, \dots, s_n\}$, un string t es superstring de S si $\forall 1 \leq i \leq n, s_i$ o s_i^R (la reversa de s_i) es substring de t .
 - (a) Escribir un algoritmo aproximado de factor $2H_n$. (Es posible conseguir uno de factor constante, por si quieren intentarlo)
 - (b) Mostrar la correctitud y determinar la complejidad del algoritmo propuesto.