

Parcial-1-2022-AlgoryED.pdf



Anónimo



Algoritmia y Estructuras de Datos Avanzadas



2º Grado en Ingeniería Informática



Escuela Politécnica Superior
Universidad Autónoma de Madrid

**ESTUDIOSA SIEMPRE PA' SER
DOCTORA, INGENIERA,
PERIODISTA...**

**COLACAO BATIDOS
MOTIVA TU FUTURO YO.**



ColaCao

APROBASTE LA COURSE NAVETTE,
SUPERASTE A TU EX E HICISTE
NUEVOS AMIGOS. ESTE EXAMEN NO
ES NADA PARA TI. **TÚ PUEDES.**

Preguntas

1. a. (2 puntos) ii. Nos han dado una tabla de 17 elementos que tenemos que convertir en un min heap de manera "in place". ¿Cuántas veces habría que ejecutar `heapify` ?
- ii. Calcular $\lg^* 179$ calculando los logaritmos de forma aproximada si fuera preciso (por ejemplo, bastaría usar aproximaciones como que $\lg 42 = 5.xx > 5$).
- b. (3 puntos) Queremos crear un min heap "in place" sobre la lista $[4, 5, 6, 7, 1, 2, 3]$, para lo que vamos a usar la representación en heap de la misma. Indicar cómo debemos abordar dicha creación y representar el estado del heap tras cada uno de sus pasos. Dar también la lista del min heap final.
- c. (5 puntos) Vamos a crear una cola de prioridad (PQ) sobre la lista $1 = [10, 11, 12, 7, 6, 5, 4]$ usando un min heap como estructura de datos (ED). Para ello:
- Crear primero la PQ a partir de un min heap vacío insertando sucesivamente los elementos de 1 indicando la evolución del heap tras cada inserción.
 - A continuación extraer el primer elemento de la PQ y a continuación insertar 9 , indicando el estado del heap tras la extracción e inserción.
 - A continuación volver a extraer el primer elemento de la PQ y a continuación insertar 2 , indicando el estado del heap tras la extracción e inserción.

2. a. (2 puntos) Explica brevemente en qué se diferencia el `find` con compresión de caminos respecto al `find` normal. ¿Qué ventajas tiene el `find` con compresión de caminos frente al otro? ¿Y qué inconvenientes?
- b. (4 puntos) La lista $[5, 6, 5, 8, -3, -2, 3, -2]$ representa un conjunto disjunto (CD) con índices entre 1 y 8 donde vamos a ejecutar `finds` con compresión de caminos.
- Representar dicho CD como una familia de árboles.
 - Sobre dicha familia, aplicar `find(2)` y `find(8)` y a continuación `union` sobre los resultados de los `find`.
 - Sobre el resultado, aplicar a continuación `find(7)` y `find(4)` y a continuación `union` sobre los resultados de los `find`.
 - Representar el CD resultante sobre una lista.

Indicar los pasos dados en suficiente detalle.

- c. (4 puntos) Dar las codificaciones Huffman y Shannon de un archivo con caracteres a, b, c, d, e, f y frecuencias respectivas $15, 8, 4, 2, 2, 1$. Dar también el tamaño en bits de las respectivas compresiones del archivo. Indicar para cada algoritmo los pasos dados en suficiente detalle.

WUOLAH