

# Parcial-1-2022-ENUNCIADO + SOLUC...



EPS\_Apuntes



Algoritmia y Estructuras de Datos Avanzadas



2º Grado en Ingeniería Informática



Escuela Politécnica Superior  
Universidad Autónoma de Madrid

**vodafone** **yu**

**FANTASÍA  
DE FIBRA**

**yu Fibra**

 **300Mbps**

**20€**  
/mes

SOLO ONLINE



**SIN  
PERMANENCIA**



ENUNCIADO**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 mini 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.

SOLUCIONES





vodafone yu

# FANTASÍA DE FIBRA

SIN PERMANENCIA

yu Fibra



**300Mbps**

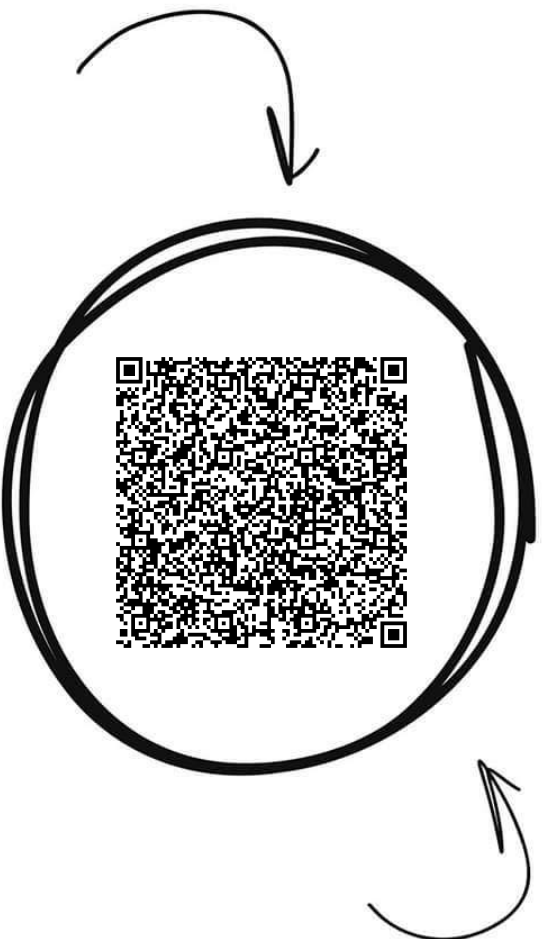
**20€**  
/mes



# Algoritmia y Estructuras de...



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**



**Banco de apuntes de la**

**WUOLAH**

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



a) En el find con compresión de caminos todos los nodos entre el nodo dado (real) y el representante pasan a ser hijos directos del representante

Ventajas: Hace árboles menos profundos

Inconvenientes: No se puede conocer la profundidad real del árbol

b)

a)

b)

c)

d)

5 6 5 6 4 5 5 8

c) Huffman

a	b	c	d	e	f
15	8	4	2	2	1

$D_f = 15 \cdot 1 + 8 \cdot 2 + 4 \cdot 3 + 2 \cdot 4 + 2 \cdot 5 + 1 \cdot 5 = 15 + 16 + 12 + 8 + 10 + 5 = 66 \text{ bits}$

vodafone yu

yu Fibra

300Mbps

20€/mes

SOLO ONLINE



SIN PERMANENCIA



Sh a n n n

a	15	0				
b	8	1	10			
c	4	1	11	110		
d	2	1	11	111	1110	
e	2	1	11	111	1111	1
f	1	1	11	111	1111	1

Misma codificación  
que Huffman

FANTASÍA DE FIBRA



WUOLAH