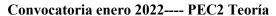


Universidad de Alcalá Departamento de Ciencias de la Computación

Estructuras de Datos





Nombre y apellidos:

DNI:

- Duración del examen: 2 horas.
- Todas las hojas entregadas deben tener nombre y DNI del alumno y № de página/Total de páginas.
- Las respuestas a los ejercicios deberán escribirse en <u>pseudocódigo y estar</u> <u>claramente justificadas</u>, es decir, acompañadas de una breve y clara explicación.
- En todas las preguntas pueden incluirse <u>operaciones auxiliares</u>, debidamente justificadas, si se considera necesario.
- NO SE CORREGIRÁ NINGUN EJERCICIO SIN INCLUIR LA DESCRIPCIÓN DE LAS OPERACIONES BÁSICAS DEL TAD QUE SE UTILICEN (indicando los tipos de entrada y salida de las mismas, las ecuaciones de definitud y explicando su funcionamiento).

Ejercicio 1.- (2,5 puntos)

a) Define árbol binario de búsqueda, AVL y montículo de máximos.

(0,5 puntos)

- b) Insertar los siguientes nodos (15,41,31,20,64,29,88,21,22) en un árbol AVL vacío, indicando los pasos y rotaciones necesarias. (0,5 puntos)
- c) Borrar el 64 y 41 en el AVL resultante del aparado b). Indicar los pasos y las rotaciones realizadas, en caso de que sean necesarias. (0,5 puntos)
- d) Indicar el contenido de un montículo de máximos, representado por un vector, después de insertar en un montículo vacío los siguientes enteros: 13, 0, 6, 1, 10, 9, 2, 8, 3, 4, 7.

 (0,5 puntos)
- e) Indicar su contenido después de eliminar el máximo. (0,5 puntos)

Ejercicio 2.- (2 ptos)

Partiendo de las operaciones básicas del tipo abstracto de datos ARBOL_BINARIO[ELEMENTO] y suponiendo conocida la operación *iguales*: elemento elemento boolean (que comprueba si dos elementos son iguales) escribe la siguiente operación:

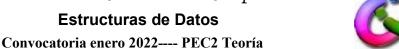
 hermanos? elemento elemento abin → bool, comprueba si los elementos dados son hermanos en el árbol binario.

> Grado en Ingeniería Informática Grado en Ingeniería de Computadores Grado en Sistemas de Información



Universidad de Alcalá Departamento de Ciencias de la Computación

Estructuras de Datos





Ejercicio 3.- (3 ptos)

Partiendo de las operaciones básicas del tipo abstracto de datos ARBOL_GENERAL[ELEMENTO] y suponiendo conocida la operación *iguales*: elemento elemento → boolean (que comprueba si dos elementos son iguales) escribe las siguientes operaciones:

- a) es_padre? elemento bosque -> bool, comprueba si el elemento es la raíz de uno de los árboles del bosque. (1 punto)
- b) hermanos? elemento elemento agen > bool, comprueba si los elementos dados son hermanos en el árbol general. (2 puntos)

Ejercicio 4.- (2,5 ptos)

Suponiendo conocidas las operaciones <=: natural natural → bool, que comprueba si un número natural es menor o igual que otro:

- 4.1 Escribir en pseudocódigo la siguiente operación (puede ser parcial), partiendo únicamente de las operaciones básicas del tipo abstracto de datos PILA[NATURAL]:
- parte_de?: pila pila → bool, comprueba si todos los elementos de la primera pila están en la segunda, en el mismo orden y consecutivos. (1,25 puntos)

Ejemplo s:	Pila1:	Pila2	Pila3:
			1
			8
	5		2
	4	8	3
	3	2	5
	2	3	6
	1	5	4

 $Parte_de?(pila1, pila3) \rightarrow false \quad Parte_de(pila2, pila3) \rightarrow true$

- 4.2 Escribir en pseudocódigo la siguiente operación (puede ser parcial), partiendo únicamente de las operaciones básicas de los tipos abstractos de datos COLA[NATURAL] y LISTA2[NATURAL]:
 - Ordenados: cola \rightarrow lista crea una lista con los elementos de la cola en orden de mayor a menor. (1,25 puntos)