## UNIDAD TEMÁTICA 7 - CLASIFICACIÓN PARTE II

## PRACTICOS DOMICILIARIOS INDIVIDUALES - 2

## **EJERCICIO 1 - ARBOL PARCIALMENTE ORDENADO**

Analiza cuidadosamente las siguientes preguntas y selecciona la respuesta más adecuada en cada una fundamentando tu decisión. ¿Por qué las otras respuestas no te parecen correctas? ¿Agregarías alguna pregunta sobre algún aspecto de este tema que te resulte particularmente dificultoso de comprender?

- 1. Un Árbol Parcialmente Ordenado es
  - a. Un árbol binario que siempre está balanceado
  - b. Un árbol binario de búsqueda
  - c. Un árbol para el que las inserciones siempre toman entre 2<sup>h+1</sup> y 2<sup>h+1</sup>-1 comparaciones, siendo h la altura del árbol
  - d. Ninguna es correcta...
- 2. ¿qué operaciones primitivas del TDA ARBOL BINARIO (y el correspondiente Nodo) *no* precisaríamos para implementar las dos operaciones indicadas del TDA APO?
  - a. TNodoArbolBinario.hijoIzq()
  - b. TNodoArbolBinario.hijoDer ()
  - c. TNodoArbolBinario.Buscar(unaClave)
  - d. TNodoArbolBinario.Padre()
- 3. ¿cuál sería el orden del tiempo de ejecución de la operación TNodoArbolBinario.Padre() si el árbol está representado como los árboles comunes ya tratados?
  - a) O(n)
  - b) Constante
  - c) O(n2)
  - d) O(log (n))
- 4. Si el TDA Árbol Binario Parcialmente Ordenado TDA APO hereda del TDA Arbol Binario, para insertar un nuevo elemento deberíamos:
  - a) Usar la operación heredada del TDA AB "Insertar(unNodo)" directamente
  - b) Escribir una nueva operación de inserción, en la primer posición disponible del último nivel
  - c) Escribir una nueva operación que coloque el nuevo elemento en la posición más a la izquierda del último nivel y luego haga los intercambios necesarios para que finalmente quede en el lugar que le corresponde
  - d) Usar una operación externa que luego de insertar recomponga la condición necesaria para el TDA APO
- 5. ¿cuál sería el orden del tiempo de ejecución de la primitiva "Insertar" del TDA APO representado en la forma habitual de árboles?
  - a) O(n)
  - b) Constante
  - c) O(n2)
  - d) O(log (n))