

# Examen de Estructura de Datos y de la Información

## Parte de Martín

- 1) (2pto) Construcción de una cola enlazada de claves numéricas sobre un vector de 10 posiciones (Usando la simulación de memoria sobre vectores). Dibujar el estado del vector al término de cada bloque de operaciones propuestos:
  - a) Constructor de la Cola.
  - b) Insertar 5, 10, 15.
  - c) Sacar, Sacar (o sea sacar dos veces) e insertar 18.
- 2) (1pto) Construir un árbol B2 ( $n=2$ ) y dibujar como queda el árbol al final de los siguientes bloques de acciones.
  - a) Insertar: 15, 24, 32, 5, 9, 36, 14, 13, 2
  - b) Insertamos el 40, el 48 y borramos el 15.
- 3) (2pto) Tenemos un árbol binario de búsqueda al que se le aplica un algoritmo hipotético de inserción que va rellenando el árbol de izquierda a derecha completando los niveles para pasar al siguiente. Hacer una función que se le pase el número de nodos del árbol y un nodo, a partir de la determinación de su altura saber si es un árbol completo o no, implementarlo con una complejidad logarítmica y demostrar que en efecto se obtiene dicha complejidad.

## Parte de Puerto

- 1) (0,4pto) En el algoritmo de ordenación quicksort, ¿por qué  $i, j$  no pueden parar aunque encuentren elementos iguales al pivote?
- 2) (0,3pto) Dados {9,"Hola", "a", 11.3, "Adios"}, ¿Es aceptable hacer un conjunto?
- 3) (0,4pto) En una tabla hash abierta de  $FC=1$  una búsqueda exitosa=1,5 y una infructuosa=1, ¿Por qué cuesta más una búsqueda exitosa?
- 4) (0,4pto) En una tabla hash abierta un  $FC=1$  es lo ideal, ¿Por qué?
- 5) (0,3pto) Relación de los diferentes tipos de ficheros y las estructuras de datos.
- 6) (0,4pto) Sea un grafo con  $n$  nodos. ¿Cuántas aristas forman el árbol de costo mínimo calculado por Prim?
- 7) (0,3pto) Al insertar un nodo en un montículo como vector es necesario situar el elemento en la última posición al final del vector, ¿Por qué?
- 8) (0,4pto) En el siguiente montículo insertar el elemento 25. ¿Complejidad de dicha inserción? ¿Es el caso mejor o peor? ¿Por qué?

[21|15|18|12|14]

- 9) (1pto) Sabiendo que la bondad de un nodo 'i' de un grafo es la suma del mínimo de las longitudes de 'i' a cualquier nodo 'k' más la suma del mínimo de las longitudes de cualquier nodo 'k' a 'i'. Realizar una función que calcule el nodo más bondadoso de un grafo (aquel que tiene una bondad mayor)
- 10) (0,4pto) En la inserción directa, si tenemos el pivote en la posición 'i', ¿Cuál es el número máximo y mínimo de comparaciones y movimientos a realizar para colocarlo en la posición adecuada?
- 11) (0,3pto) Complejidades del caso mejor y del caso peor de una búsqueda en anchura para un nodo con lista de adyacencias.