

Algoritmos y Programación II – Cátedra Lic. Gustavo Carolo

Evaluación Final – 2009-02-23

--- Entregar teoría y práctica por separado --- Leer bien el enunciado ---

| | |
|-----------------------|--------------------|
| Nombre: | Padrón: |
| Mail: | |
| Cuatrimestre cursado: | T: P: F: |

Teoría Tema 1

a: b: c: d: e:

Dado el siguiente lote de datos:

108, 397, 327, 11, 333[†], 17, 375, 36[†], 25[†], 376, 354[†], 276, 229, 14, 258,
8[†], 93, 221[†], 373, 298, 106[†]

Se pide:

- a) Procesar el B-tree de $m=2$ resultante de ingresar los elementos del lote en el orden dado y luego eliminar los siguientes:

258, 229[†], 298[†]

Mostrar los resultados parciales tras insertar o eliminar un elemento marcado con [†].

- b) Desarrollar en orden ascendente los algoritmos de *Radix sort* (entero) y *Quick sort* (hasta haber recorrido tres particiones).
- c) Desarrollar el algoritmo de selección natural con un *buffer* de tamaño 5.
- d) Desarrollar todas las fases del algoritmo de *merge* polifásico para 3 vías de entrada con 21 particiones ordenadas tomadas del lote de datos.
- e) Como es sabido, los algoritmos de *Heap* y *Quick sort* son de orden $n \log_2 n$. Sin embargo, es posible utilizar un ABO para ordenar un conjunto de elementos y su orden también sería de $n \log_2 n$ ya que hay que hacer n inserciones ordenadas y cada inserción ordenada es de orden $\log_2 n$ debido a que para insertarlo se realiza una búsqueda binaria. Explique cual es la razón por la que este método no es utilizado frente al *Heap* o *Quick sort* cuya eficiencia es similar.