Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Computación
Estructuras de Datos y Algoritmos I

Práctica 3

1. Quicksort

Basado en lo visto en clase y en los archivos provistos, realice las siguientes tareas:

- Modificar *my_qsort* para que ordene los elementos en orden descendente. Probar con distintas permutaciones de entrada (ordenadas, desordenadas, con elementos repetidos, etc.)
- Implementar *my_qsort_random*, un quicksort con elección de pivote aleatorizada, que use una función de partición *randomized_partition* similar a la mostrada en sección 8.3 del libro de Cormen.

2. Árboles AVL

Basado en lo visto en clase y en los archivos provistos, realice las siguientes tareas:

- Implementar los casos right-right y right-left en la inserción.
- Escribir casos de testing unitario para left-right, right-right y right-left.
- Implementar las operaciones remove y destroy.
- Implementar un caso de testing donde se inserten 1 millón de strings de entre 2 y 30 caracteres alfanuméricos cada una, generadas aleatoriamente y alocadas con *mem_new*. Luego realizar 300000 consultas con strings del mismo tipo, generadas aleatoriamente y NO alocadas en el heap, sino generadas en el momento. Medir el tiempo de las 300000 consultas. ¿Cuánto tardó en promedio cada consulta? ¿Cuánta memoria pico llegó a ocupar el proceso que ejecutó el programa, medido por el sistema operativo?
- ¿Cuál fue el pico de memoria alocada por el programa con *mem_new* (modificar *mem_new* y mem_free para que lleven la cuenta de los bytes que alocan en el heap)?

3. Binary Heap

- Implementar las operaciones *heap_new*, *heap_destroy*, *heap_insert* y *heap_delete* con el código base enviado por mail a la lista de la materia.
- Escribir tests de unidad para testear la implementación del heap binario.