



UNIVERSIDAD SIMÓN BOLÍVAR

Dept. Computación y Tecnología de la Información

Laboratorio de Algoritmos y Estructuras II

Abr-Jul 2016

Laboratorio 3

1. En el archivo `heapsort.py`, implemente el algoritmo de `heapsort` según el libro *Algorithms in a Nutshell* de George T. Heineman, Gary Pollice, and Stanley Selkow que se encuentra en la página http://chimo.ldc.usb.ve/Algorithms_in_a_Nutshell_Second_Edition.pdf
2. Implemente una cola de prioridad que se encontrará en el archivo `colaprioridad.py`. Para ello deberá hacer uso de clases, creando los siguientes métodos:
 - `crearCola()`: crea una cola vacía
 - `eliminar()`: Destruye una cola de prioridad
 - `agregarElem(p,k)`: Inserción de un elemento `p` en la cola con prioridad `k`.
 - `consultarMax()`: Consulta del elemento con menor prioridad
 - `extraerMax()`: Extracción del elemento con menor prioridad
 - `actualizarElem(p,k)`: Actualización de la prioridad `k` de un elemento `p` que ya esta en la cola
 - `elimElem(p)`: Eliminación de un elemento que esta en la cola

Para que su cola de prioridad sea más eficiente deberá hacer uso de `heaps` binarios.

3. Pruebe su cola de prioridad en un archivo cliente llamado `cadena.py`. Suponga la secuencia:

`C O * * * L * A * D * E * P R I O R * * * I * D * A * * *`

donde una letra significa que insertó la misma en la cola de prioridad y un asterisco significa eliminar el elemento con el máximo valor (El orden está definido según el orden alfabético donde `a > b...` y `> z`). Esto se aplica a una cola de prioridad inicialmente vacía. Debe de dar la secuencia de valores devueltos según la instrucciones de la cadena de caracteres antes descrita.

Condiciones de la entrega

Cree un archivo comprimido del tipo “tgz” llamado `Lab3-X-Y.tgz`, donde `X` es su número de carné y `Y` el de su compañero. Su contenido serán los archivos `heapsort.py`, `coladeprioridad.py` y `cadena.py`. Debe subir el archivo en el aula virtual, en la sección de Semana 4, el día 5 de mayo de 2016 antes de las 4:30 p.m.