

Evaluación Heap

Leer atentamente a la siguiente información que especifica cómo se evaluará el tema Heap.

La evaluación de Heap constará de tres partes:

- 1) Programación de la clase `Heap` (entrega hasta el martes 8 sep) (**4 puntos**)
- 2) Ejercicios de programación (martes 8 sep) (**3 puntos**)
- 3) Preguntas y ejercicios teóricos (jueves 3 sep) (**3 puntos**)

Además, para aprobar la evaluación hay que alcanzar al menos el 50% del puntaje de cada ítem. A continuación se detalla en qué consiste cada ítem:

Programación de la clase `Heap`

Se les compartió un [archivo .zip](#) con código incompleto:

practico_heap 3 elementos

 `__init__.py`

 `interactive_heap.py`

 `test_heap.py`

- Lo primero que deben hacer es agregar un archivo con el nombre `heap.py` donde implementen la clase `Heap` tal como se especificó en las filminas correspondientes a dicho tema.
- El módulo `test_heap` importa la clase `Heap` y corre algunos tests de [caja negra](#) para chequear que funcione como se espera. Se ejecuta con el comando `python -m unittest` en una consola, situados en la misma carpeta donde están los archivos.
- El módulo `interactive_heap` es una aplicación de línea de comandos para utilizar el heap programado por ustedes. Se ejecuta con el comando `python3 interactive_heap.py`.

Se aceptan los trabajos por mail a mebordone@gmail.com hasta el martes 8 de septiembre a las 23:59:59. Mandar un archivo comprimido .zip con los archivos originales más el archivo heap.py agregado por ustedes.

Notar que se espera que corra bajo python3.

El criterio de evaluación es el siguiente (de mayor a menor importancia):

- ¿el código implementa el heap mediante un arreglo?
- ¿pasan todos los tests?
- ¿la aplicación `interactive_heap` funciona correctamente?

- ¿el código fue entregado a tiempo?
- ¿el código contiene errores que los tests no detectan?
- ¿el estilo de código está cuidado? Por ahora, sólo voy a corregir:
 - nombres de variables
 - presencia de comentarios

Ejercicios de programación

Se les pedirá que resuelvan problemas simples (funciones, corregir un código que está mal, escribir algunos tests para probar que una función se comporta como esperamos) en base a los últimos temas trabajados: cola de prioridades, Heap, implementación mediante un arreglo(ver [aquí](#) y [aquí](#)).

Preguntas y ejercicios teóricos

Se presentarán situaciones problemáticas similares a las trabajadas en los últimos prácticos (ver [aquí](#) y [aquí](#)) para resolver utilizando lápiz y papel. Se harán preguntas teóricas relacionadas a la terminología aprendida y a la complejidad de las operaciones asociadas a esta estructura de datos.