

# Examen-practicas-Extraordinaria-2...



osnofla



Algoritmia y Estructuras de Datos Avanzadas



2º Grado en Ingeniería Informática



Escuela Politécnica Superior  
Universidad Autónoma de Madrid

**ESTUDIOSA SIEMPRE PA' SER  
DOCTORA, INGENIERA,  
PERIODISTA...**

**COLACAO BATIDOS  
MOTIVA TU FUTURO YO.**





ColaCao

UNA PERSONA SENTADA EN SU HABITACIÓN  
PORQUE TIENE QUE ESTUDIAR. ¿CÓMO SE  
LLAMA LA PELÍCULA? TU VIDA AHORA MISMO.  
**COLACAO BATIDOS TE ACOMPAÑA EN ESTO.**

## Ordenando Max Heaps

Algoritmos y Estructuras de Datos Avanzadas, junio 2023

### Examen de prácticas

**IMPORTANTE:** se pueden utilizar las notas del curso y se debe usar el código de la práctica entregado en su día.  
**NO SE PUEDE UTILIZAR NINGÚN OTRO CÓDIGO**

#### INSTRUCCIONES

Los pasos a dar son:

1. Arranca un ordenador del laboratorio y abre una terminal Windows o Linux y, si vas a usar un notebook, ábrelo.
2. Accede a Moodle y descarga el archivo `exam_pract_junio_2023.py` a tu directorio local.
3. Trabaja sobre dicho archivo de acuerdo a las instrucciones de la sección siguiente.
4. Puedes hacerlo directamente en el archivo mediante un editor o bien copiando y pegando su contenido a un notebook.
5. Durante tu trabajo puedes comprobar tu código ejecutando  

```
python exam_pract_junio_2023.py
```

  
si trabajas con Windows o, si trabajas en Linux,  

```
./exam_pract_junio_2023.py
```
6. Cuando hayas completado tu trabajo y comprobado que tu código ejecuta correctamente, guarda tu versión del archivo `exam_pract_junio_2023.py` en otro de nombre `apellido1_apellido2.py` donde `apellido1`, `apellido2` son tus apellidos y súbelo al buzón Moodle activado al efecto.

#### ORDENANDO ARRAYS MEDIANTE MAX HEAPS

En la práctica 1 hemos creado min heaps e implementado diversas funciones para su gestión. En este examen vamos a desarrollar una variante de lo anterior para ordenar arrays pero trabajando sobre max heaps. Para ello

- En primer lugar adaptaremos algunas de nuestras funciones para min heaps de manera que podamos usarlas para crear max heaps.
- A continuación desarrollaremos una función `extract_max_heap(t: np.ndarray) -> Tuple[int, np.ndarray]` que extrae el elemento en la raíz de un max heap y lo devuelve junto con el max heap restante tras la extracción. Para esto podemos adaptar la función `pq_remove(t: np.ndarray) -> Tuple[int, np.ndarray]` desarrollada en la primera práctica.
- A continuación desarrollaremos una función `sort_max_heap` que use la función `extract_max_heap` anterior para que vaya extrayendo repetidamente la clave en la raíz del max heap (esto es, el mayor elemento en cada momento), la vaya añadiendo a una lista y continúe procesando de manera similar los max heaps que devuelve `extract_max_heap`. De esta manera dicha lista tendrá una ordenación invertida del array inicial, ordenación que corregiremos antes de devolverla como un array.

Para todo lo anterior, y sobre el archivo `exam_pract_junio_2023.py` accesible en Moodle, dar los siguientes pasos

1. Completar el código de la función  

```
max_heapify(t: np.ndarray, i: int)
```

  
adaptando a max heaps de manera adecuada el código de la función `min_heapify` de tu práctica, trabajando siempre de manera in place.
2. Completar el código de la función  

```
create_max_heap(t: np.ndarray)
```

  
adaptando de manera adecuada el código de la función `create_min_heap` de tu práctica y trabajando también de manera in place.
3. Completar el código de la función  

```
extract_max_heap(t: np.ndarray) -> Tuple[int, np.ndarray]
```

  
que extraiga la raíz de un max heap y devuelva dicha raíz así como el nuevo max heap tras la extracción.
4. Completar el código de la función  

```
sort_max_heap(t: np.ndarray) -> np.ndarray
```

  
que devuelva una ordenación del max heap t aplicando de manera adecuada la idea expuesta más arriba.

Tras cada uno de los pasos anteriores, puedes ejecutar el script del archivo `exam_pract_junio_2023.py` comprobando que tus funciones actúan correctamente.

WUOLAH