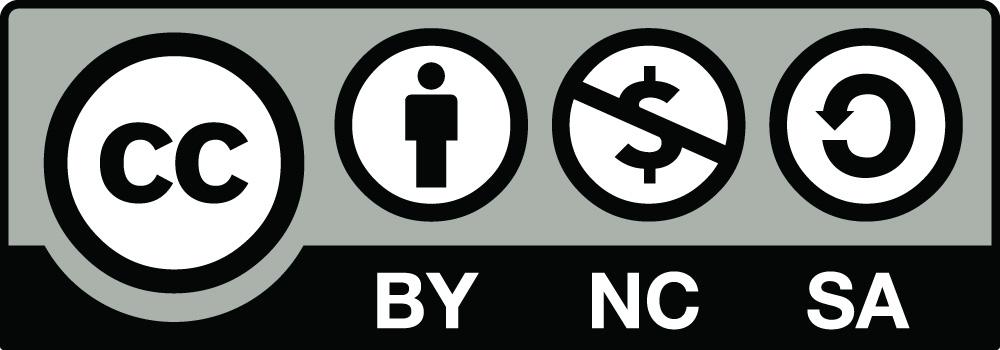
|  |  |
| --- | --- |
| **Estructura de Datos y Algoritmos**  **Grado Ingeniería Informática**  **Universidad Carlos III de Madrid**  **CURSO 2021-2022** | **Logotipo de la Universidad Carlos III de Madrid** |

****

**Grupo 801. Segundo Examen Parcial. 25 Abril 2022.**

**Nombre:**

Instrucciones:

* Duración del examen: 60’.
* Descarga el zip parcial2.zip y descomprimelo.
* En el fichero second801.py deberás escribir tu solución. Además, añade un comentario python al principio del fichero con tu nombre y apellidos.
* NO MODIFIQUES el fichero test801.py (te ayuda a testear tu solución).
* Durante el examen, en tu equipo únicamente podrás tener abierto el entorno de desarrollo de Python (por ejemplo, spyder o pycharm). No puedes consultar ningún material (código, apuntes, libros, etc).
* Cuando queden 5 minutos el profesor te avisará para que subas tu solución. Podrás abrir el navegador y entrar en tu grupo reducido de aula global. Sube únicamente el fichero second801.py a la actividad “Segundo Parcial” en Aula Global.
* Es tu responsabilidad comprobar que has subido la versión correcta del fichero.
* Los móviles deben estar apagados y nunca pueden estar encima de la mesa.
* No puedes salir del aula antes de que pasen 30 minutos.
* No está permitido ir al baño.

**Problema:**

La clase MyBinarySearchTree permite representar árboles binarios de búsqueda para almacenar números enteros.

En la clase MyBinarySearchTree, implementa una nueva función *removeInsideRange,* que recibe dos enteros (max y min) y **elimina todos los nodos hoja** que tienen un valor dentro del rango que hay en el intervalo [min, max].

Además, la función devuelve una lista de python con los nodos hoja eliminados en orden descendente. Si el árbol no tiene ningún elemento que borrar en el intervalo [min, max]., la función deberá devolver una lista vacía.

**Nota: Solo se borran aquellos nodos que son hoja al comienzo de la ejecución del método. Si un nodo se convierte en hoja al borrar a su hijo/s no se considera hoja al inicio y por tanto no se eliminaría.**

A continuación, tienes algunos ejemplos:

Gráfico

Descripción generada automáticamente con confianza baja

|  |  |
| --- | --- |
| **Operación** | **Salida de la función (nodos hoja eliminados)** |
| tree.removeInsideRange(1,120) | [80, 54, 24, 18, 5] |
| tree.removeInsideRange(15,20) | [18] |
| tree.removeInsideRange(0,0) | [] |
| tree.removeInsideRange(54,80) | [80,54] |
| tree.removeInsideRange(-10,0) | [] |

Está permitido crear funciones auxiliares. Está permitido utilizar listas de Python. No está permitido utilizar diccionarios. Para que la solución se considere correcta, la función propuesta debe ser **correcta** (resuelva el problema), **robusta** (no tenga errores y funcione para cualquier entrada) y **eficiente** en términos de complejidad temporal y espacial (evitar el uso de estructuras auxiliares). Además, el código debe ser fácil de entender y mantener.