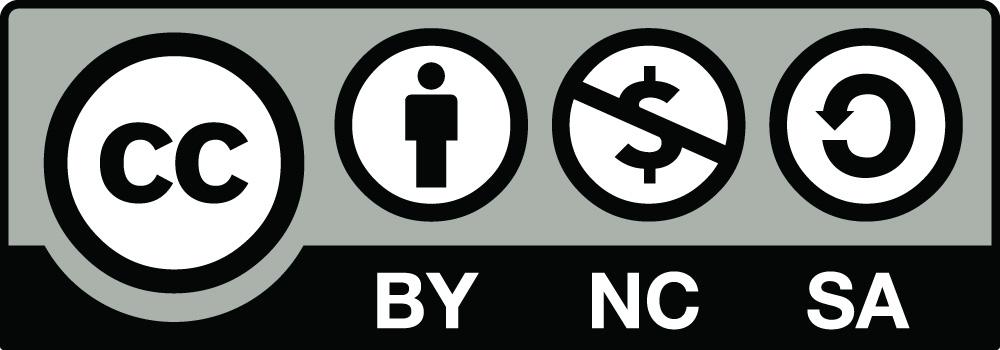
|  |  |
| --- | --- |
| **Estructura de Datos y Algoritmos**  **Grado Ingeniería Informática**  **Universidad Carlos III de Madrid**  **CURSO 2021-2022** | **Logotipo de la Universidad Carlos III de Madrid** |

****

**Examen Final. 20 Mayo 2022.**

**Nombre:**

Instrucciones:

* Duración del examen: 60’.
* Descarga el zip parcial1.zip y descomprimelo.
* En el fichero parcial84.py deberás escribir tu solución. Además, añade un comentario python al principio del fichero con tu nombre y apellidos.
* NO MODIFIQUES el fichero test84.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 parcial84.py a la actividad “Primer 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.

**Problem 1**:

En la clase **MyDList**, complete la función **removeDividers**(), que toma como parámetros una lista ordenada, una instancia de la clase **MyDList**. La función elimina todos los nodos cuyo elemento es un divisor de otro elemento de la lista. La función no devuelve nada. La función modifica la lista.

Por ejemplo, si la lista está formada por: **2<->5<->6** entonces la lista resultante será **5<->6** ya que 2 es un divisor del elemento 6.

**More examples:**

|  |  |
| --- | --- |
| **La lista antes de la operación** | **La lista después de la operación** |
| **7** | **7** |
| **0<->0<->0** | **0<->0<->0** |
| **1<->1** | **1** |
| **2<->2<->2** | **2** |
| **0<->1<->1<->3<->3<->3<->5<->6<->7<->8<->11<->12** | **0<->5<->7<->8<->11<->12** |
| **1<->1<->2<->2<->4<->5<->7<->7<->8<->9<->9<->13<->14** | **5<->8<->9<->13<->14** |

No puedes añadir nuevos atributos o funciones a las clases **DNode** y **MyDList**.

Para considerar que su solución es correcta, la función propuesta debe ser correcta (resolver el problema), robusta (no tener errores y funcionar 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.

No se permite el uso de estructuras de Python como diccionarios o listas.

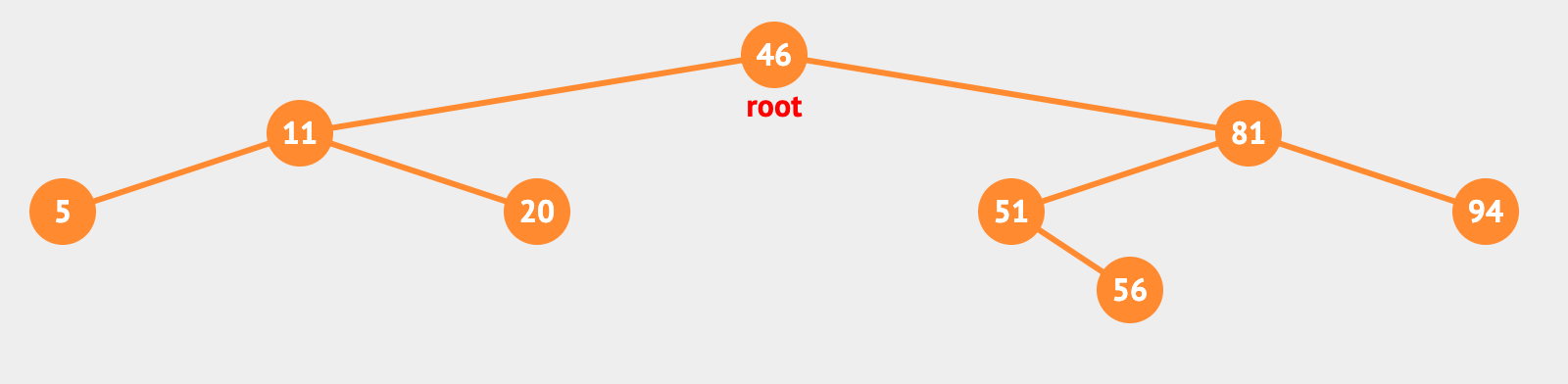
**Problema 2:**

La clase **MyBST** permite representar árboles de búsqueda binarios para almacenar enteros. En la clase **MyBST**, implementa una nueva función recursiva, **getCount**, que recibe dos enteros, low y high como rango, y cuenta el número de nodos que se encuentran en el rango dado. No es necesario que low y high sean elementos del árbol.

Para que la solución se considere correcta, la función propuesta debe ser **correcta** (resolver el problema), **robusta** (no tener errores y funcionar 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.

Está permitido crear funciones auxiliares. No se permite el uso de estructuras de Python como diccionarios o listas.

**Ejemplos:**



|  |  |  |
| --- | --- | --- |
|  | Salida | Explicación |
| tree.getCount (0, 5) | 0 | Si el árbol está vacío |
| tree.getCount(5,0) | 0 | start > end [start = 5, end = 0] |
| tree.getCount(1,4) | 0 | No hay ningún elemento en el rango [1,4], |
| tree.getCount(95,100) | 0 | No hay ningún elemento en el rango [95,100] |
| tree.getCount(21,45) | 0 | No hay ningún elemento en el rango [21, 45] |
| tree.getCount(57,80) | 0 | No hay ningún elemento en el rango [57, 80] |
| tree.getCount(4,100) | 8 | Número de elementos en el rango [4,100] |
| tree.getCount(10,81) | 6 | Número de elementos en el rango [10, 81] |
| tree.getCount(12,100) | 6 | Número de elementos en el rango [12, 100] |
| tree.getCount(4,50) | 4 | Número de elementos en el rango [4, 50] |
| tree.getCount(50,90) | 3 | Number of elements in the range [50, 90] |
| tree.getCount(5,10) | 1 | Number of elements in the range [5,10] |