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

**Imagen que contiene dibujo, señal, plato

Descripción generada automáticamente**

**Examen Final. 20 Mayo 2022.**

**Nombre:**

Instrucciones:

* Duración del examen: 60’.
* Descarga el zip de Aula Global y descomprimelo.
* En el fichero con el esqueleto (.py) deberás escribir tu solución. Además, añade un comentario python al principio del fichero con tu nombre y apellidos.
* NO MODIFIQUES el fichero con el unittest (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 con el código a 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.

**Enunciado:**

Te hemos proporcionado una implementación básica de la clase MyDList, una lista doblemente enlazada donde almacenaremos números enteros no negativos. Debes implementar un algoritmo, remove\_section\_by\_sum(k) que reciba un valor entero, k, busque la primera combinación disponible de números en la lista que sume k y los elimine de la secuencia.

Puedes encontrar algunos ejemplos a continuación:

|  |  |  |
| --- | --- | --- |
| Lista original | Llamada al método | Lista resultante |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (1) | 2<->3<->4 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (3) | 3<->4 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (2) | 1<->3<->4 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (4) | 1<->2<->3 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (5) | 1<->4 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (6) | 4 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (9) | 1 |
| 1<->2<->3<->4 | l.*remove\_section\_by\_sum* (10) | (empty) |
| 1<->2<->5<->10 | l.*remove\_section\_by\_sum* (9)  l.*remove\_section\_by\_sum* (37) | 1<->2<->5<->10 |
| (empty) | l.*remove\_section\_by\_sum* (x) | (empty) |
| (any list) | l.*remove\_section\_by\_sum* (-x) | (any list) + error printed |

Para conseguirlo, se divide el ejercicio en dos pasos:

1. Implementar una función auxiliar llamada remove\_sublist(self, start, end, count) que elimine una sección de MyDList en un solo paso.
2. Los parámetros start, end son el **nodo inicial y final de la sección** a eliminar de la lista.
3. El parámetro count es el número de nodos a eliminar (para actualizar el tamaño en consecuencia). Asumiremos que este recuento se proporciona correctamente (por lo tanto, no es necesario volver a comprobar que es correcto dentro de este método)
4. Implementar el método remove\_section\_by\_sum(k), aprovechando el anterior siempre que sea necesario.

No se pueden añadir nuevos atributos o funciones a las clases DNode y MyDList. El código propuesto debe resolver el problema, ser capaz de manejar entradas inesperadas y ser eficiente en términos de complejidad de tiempo y espacio. El código debe ser legible y fácil de seguir.

No se permiten estructuras de Python como listas o diccionarios de Python.