



Convocatoria Extraordinaria. 21 Junio 2022.

Problema 1: Sea MyDList, la implementación de 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. Una vez encontrada esta combinación, el método no debe continuar la búsqueda.

Puedes encontrar algunos ejemplos a continuación:

Lista original	Llamada al método	Lista resultante
1<->2<->3<->4	<code>l.remove_section_by_sum (1)</code>	2<->3<->4
2<->1<->3<->4	<code>l.remove_section_by_sum (3)</code>	3<->4
1<->2<->3<->4	<code>l.remove_section_by_sum (2)</code>	1<->3<->4
3<->2<->1<->4	<code>l.remove_section_by_sum (4)</code>	3<->2<->1
1<->3<->2<->4	<code>l.remove_section_by_sum (5)</code>	1<->4
1<->2<->3<->4	<code>l.remove_section_by_sum (6)</code>	4
1<->4<->3<->2	<code>l.remove_section_by_sum (9)</code>	1
4<->2<->1<->3	<code>l.remove_section_by_sum (10)</code>	(empty)

1<->2<->5<->10	<i>l.remove_section_by_sum</i> (9) <i>l.remove_section_by_sum</i> (37)	1<->2<->5<->10
(empty)	<i>l.remove_section_by_sum</i> (x)	(empty)
(any list)	<i>l.remove_section_by_sum</i> (-x)	(any list) + error printed

Para conseguirlo, divide el ejercicio en dos pasos:

1. Implementa un método auxiliar, `remove_sublist(self, start, end, count)`, que elimine una sección de `MyDList` en un solo paso.
 - a. Los parámetros `start`, `end` son el **nodo inicial y final de la sublista** a eliminar. Ambos nodos también deben ser eliminado de la lista.
 - b. 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)
2. Implementar el método, `remove_section_by_sum(k)`, que se ocupa de localizar la primera sublista de la lista cuyos elementos suman `k`, para a continuación eliminarla, utilizando el método anterior.

No se permite el uso de estructuras de Python como listas o diccionarios de Python. No se pueden añadir nuevos atributos o funciones a las clases `DNode` y `MyDList`. El código propuesto debe resolver el problema, ser robusto y eficiente en términos de complejidad temporal y espacial. El código debe ser fácil de comprender y estar correctamente refactorizado (Zen de Python).