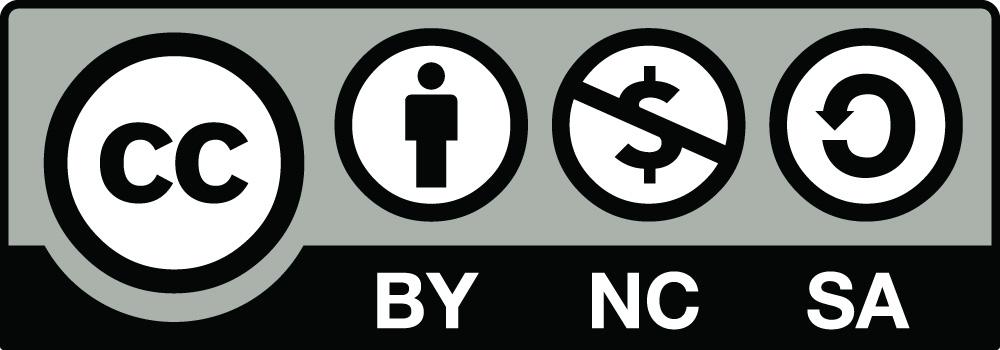
|  |  |
| --- | --- |
| **Data Structures anb Algorithms**  **Science and Data Engineering**  **Universidad Carlos III de Madrid**  **CURSO 2021-2022** | **Logotipo de la Universidad Carlos III de Madrid** |

****

**Group 96. March 18th, 2022.**

**Name:**

Instructions:

* Exam duration: 75’.
* Download and uncompress the partial1.zip file.
* Solution must be provided within partial96.py file. Add a comment at the beginning of the file with your name, surname and NIA.
* DO NOT MODIFY testpartial96.py file. This is a file to help you check whether your implementation works as intended.
* During the exam you will only be allowed to work on your chosen Python IDE (i.e. Spyder, Pycharm, VSCode…). You are not allowed to check any other resources such as slides, code samples from internet, etc.
* The teacher will do some time checks during the session. He will send a reminder when the exam is about to finish (5 mins left) to upload your solution (partial96.py file only) to Aula Global (reduced group).
* Make sure that you upload the correct version before leaving, It’s your responsibility.
* Mobile phones and any other device must be off.
* You cannot leave the classroom before 30 mins have passed.
* You cannot go to the toilet during the exam.

**Problem statement:** We provide a basic implementation of MyDList class, a doubly linked list where we will store non-negative integer numbers. You must implement an algorithm, *remove\_section\_by\_sum(k)* that receives an integer value, k, looks for the first available combination of numbers in the list summing k and removes them from the sequence.

Find some samples below:

|  |  |  |
| --- | --- | --- |
| Calling list inner values | Method call | Calling list after calling the method |
| 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 |

To achieve it, we will split the exercise in two steps:

1. Implement an auxiliar function named *remove\_sublist(self, start, end, count)* which removes a section of MyDList in a single step.
   1. Parameters *start, end* are the **initial and ending node of the section** to remove from the list.
   2. Parameter *count* is the number of nodes to remove (to update the size accordingly). We will assume that this count is provided correctly (thus, no need to double check that it is correct within this method)
2. Implement *remove\_section\_by\_sum(k)* method, leveraging the previous one whenever needed.

You cannot add new attributes or functions to clases DNode and MyDList. Proposed code must solve the problem, be able to handle unexpected inputs and be efficient in terms of time and space complexity. Code must be readable and easy to follow.

Python structures such as Python lists or dictionaries are not allowed.