



LINEAR DATA STRUCTURES AND ALGORITHMS.

ASSIGNMENT 2: ALGORITHMS

BACKGROUND.

In this assignment we are going to implement **divide&conquer** and **greedy**-based algorithms for solving different problems.

Note: The exercises proposed in this assignment are related to the exercises seen in the lectures. Thus, I strongly recommend to download, get to understand, run and debug the code examples of the lectures before start attempting the exercises of the assignment.

ASSIGNMENT 2 – HINT 1

(Week 9)

Divide and Conquer: First set of exercises.

BACKGROUND.

The folder `/src` contains the following files:

- **(MyStaticList.java, MyNode.java, MyDynamicList.java):** These classes we have seen previously in the lectures of the Block II: Data Structures.
- **DivideAndConquerAlgorithms.java:** This class contains the proposed divide&conquer functions you have to implement. *This is the only class you currently need to modify!*
- **MyMain.java:** This class tests the functionality of the divide&conquer functions.

EXERCISE.

Implement the following functions of the class DivideAndConquerAlgorithms.java.

1. `public int maxInt(MyDynamicList m);`
The function returns the maximum item of m (-1 if m is empty).
2. `public boolean isReverse(MyDynamicList m);`
The function returns whether m is sorted in decreasing order or not.
3. `public int getNumAppearances(MyDynamicList m, int n);`
The function returns the amount of times that the integer n appears in m .
4. `public int power(int n, int m);`
The function returns n^m .
5. `public int lucas(int n);`

Mathematically, the Lucas series is defined as:

$$L_n := \begin{cases} 2 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ L_{n-1} + L_{n-2} & \text{if } n > 1. \end{cases}$$

Thus, the Lucas series is as follows:

2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123

The function returns the n -est item of the lucas series.

Examples: $lucas(0) \rightarrow 2$, $lucas(4) \rightarrow 7$

6. `public void drawImage(int n);`

The function prints a pattern of a given length.

```
*
**
***
...
```