

LAB GUIDE. SESSION 5

GOALS:

a) Dynamic programming

1. The money exchange problem

Given a monetary system made up of coins of different values, the problem of the exchange consists of decomposing any given quantity into coins, using the smallest possible number of coins of that monetary system.

We know that i is the number of different types of coins, j is the amount that we want to decompose and $T(1..i)$ is a vector with the value of each type of coins in the system.

You should calculate the minimum amount of coins to return the amount j and how many coins of each type would be returned.

We have the function $C(i, j)$ that provides us with the solution to the problem to obtain the quantity j restricting itself to the types of coins T_1, T_2, \dots, T_i

$$C(i, j) = \begin{cases} \infty & \text{if } (i = 0) \text{ or } (j < 0) \\ 0 & \text{if } (j = 0) \text{ and } (i > 0) \\ \text{Min}(C(i-1, j), C(i, j - T_i) + 1) & \text{otherwise} \end{cases}$$

TO DO:

A. Work to be done

- A package `session5.exchange` in your course project, with all the files that were given with the instructions for this session together with new classes/code:
 - `Exchange.java`. You are asked to write a program to solve this problem using dynamic programming. The application must include a public method to return the number of coins (`getNumCoins(AMOUNT_TO_RETURN)`) and another public method to return an array with the list of coins used (`getBestCoins()`).

The constructor will receive the name of a text file `FILE_INPUT` as input parameter with the following format:

- The first line contains the number of different types of coins.
- The second line are integers separated by tabs where the values of each type of coins are indicated.

For example:

1 4 5 12 20 50 100 200

The output of the program should be the minimum number of coins that must be used to return the corresponding amount and how many coins of each type must be returned.

For example, for the previous input, the output of the two requested methods should be:

- `getNumCoins(28)` would return: 3
- `getBestChoice()` would return an array with the following values: [0, 1, 0, 2, 0, 0, 0, 0]
- `ExchangeTests.java`. You should modify this file. In order to check if everything is OK, you should, at least, pass all the test cases of this file.
- `ExchangeTimes.java`. Class to complete a table with time measurements. You must choose a monetary system with 10 different coins (you can decide the values of each type of coin), being n the value of the amount of money to be returned. You just need to calculate the result but not to return it.
- A **PDF document** using the course template. The activities of the document should be the following:
 - **Activity 1. Table made “manually”**
 - Create in a document the table to solve a problem with the following input with dynamic programming (returning the change for an amount of 14):

3

1 3 6
 - **Activity 3. ExchangeTimes**
 - Fill in the table with the values obtained from `ExchangeTimes.java`. What is the complexity of the method `getNumCoins()`?

n	<i>time</i>
10000
20000
40000
80000
160000
320000
640000
1280000

.....
Until you get at least 10 representative values	

B. Delivery method

You should **commit and push** your project with your new `session5` package in your Github repository with the following content inside it:

- All the requested source files.
- The requested PDF document called `session5.pdf` with the corresponding activities.

Important:

- Make sure your Github course project is up to date after the delivery and that the last commit is strictly before the deadline. If not, the mark for this session will be 0 without any exception.
- Your mark will be eventually included in the `Marks.xlsx` file in your repository. Please, do not modify anything in that file to avoid conflicts between versions.

Deadlines:

- Group L.I-01 (Thursday): March 25, 2020 at 11:55pm.
- Group L.I-02 (Wednesday): March 24, 2020 at 11:55pm.
- Group L.I-03 (Tuesday): March 23, 2020 at 11:55pm.
- Group L.I-04 (Monday): March 29, 2020 at 11:55pm.