COMPUTER PROGRAMMING
Unit 1 teaching guide
RECURSION

Jon Ander Gómez Adrián
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
jon@dsic.upv.es

January 25, 2018

1 Contents

1. Introduction to the concept of recursion
2. Design of recursive methods
3. Kinds of recursion
4. Call stack
5. Recursion on arrays
    - Recursive algorithms for operating with arrays
    - Recursive algorithms for searching
    - Recursive version of the binary search algorithm
6. Recursion versus iteration

2 Specific objectives

Upon completion of this subject the student should be able to ...

Objective 1 define the concept of recursion and explain what recursion is.

Objective 2 identify the general rule(s) and the trivial case(s) given a recursive definition or a recurrence relation.

Objective 3 write, using a programming language, a recursive method given a recursive definition or a recurrence relation.

Objective 4 obtain the recurrence relation given a recursive method written in a programming language, identifying the trivial case(s) and the general rule(s).

Objective 5 obtain the equation with the trivial case(s) and the general rule(s) given a recursive method written in a programming language.

Objective 6 design linear recursive methods for solving problems that can be naturally formulated in a recursive way.

Objective 7 discuss about the correctness of a designed recursive method and its possible iterative version. This implies testing if the recursive version ends properly.

Objective 8 describe the execution model of a linear recursive method. Drawing the call stack while doing the trace of a recursive method, for instance.

Objective 9 use the debugger and the step-by-step execution in order to test how a recursive method performs.

Objective 10 express the differences between a recursive algorithm and an iterative algorithm for solving the same problem.

3 References

- `http://www.merriam-webster.com/dictionary/recursion`
- `http://en.wikipedia.org/wiki/Recursion`
- `http://en.wikipedia.org/wiki/Recursion_(computer_science)`
- `http://mathworld.wolfram.com/Recursion.html`
- `http://mathworld.wolfram.com/RecurrenceRelation.html`
- `http://mathworld.wolfram.com/RecurrenceEquation.html`
- `http://en.wikipedia.org/wiki/Call_Stack`
- "Empezar a programar usando Java"
  N. Prieto, A. Casanova, et al.
  Editorial UPV
  Chapter 11

4 Planning of each session

Duration of activities

	Inside classroom	Outside classroom
Before	–	2h
Session 1	1.5h	2h
Session 2	1.5h	3h
Session 3	1.5h	3h
Session 4	1.5h	–
	6h	10h

Before the first session

Preparation of the working groups in order to apply active learning methods that foster cooperative learning. And explanation of this plan to students, pointing out there will be some activities to develop in groups and others to develop individually.

Outside classroom activities (up to 2h)

- It is proposed to read the definitions of recursion and other related concepts that can be found in the URL given as references. It is important to have a preliminary idea of what recursion is in order to understand the explanations and examples exposed, and to be able to solve some exercises.
- It is also proposed to remember the concepts of: call stack and stack frame (or call frame or activation record or activation frame). These concepts were studied in IIP during the first semester, in the unit dedicated to methods or functions.

First session

Classroom activities (1h 30')

Minutes	Activity
20'	The teacher explains. Introduction to the concept of recursion, recurrence relation, recurrence equation, recursive method or recursive sequence. How important is that a function can call itself. A simple fact with a huge expressive power.
5'	Break.
20'	The teacher explains. Trivial or base cases of a recursive definition. General rule of a recursive definition. The importance of applying the general rule with a smaller instance of the problem in order to reach one of the trivial cases. Example to be used: $n!$ → the factorial of n
5'	Break.
30'	– First phase of an Aronson puzzle (puzzle 1) – Each member of a base group solves one of the following problems: (a) Fibonacci numbers. (b) Sum of the figures of an integer number. (c) Euclides algorithm for Greatest Common Divisor. Teacher will provide you the recursive equation of each problem and explain the Aronson puzzle as an active learning methodology. For each problem the student has to write the header of the method in Java. Then, inside the method, (s)he has to write properly the instructions for solving the trivial case(s) and the expression for applying the general rule. The final result is a complete Java method with comments.
10'	Resolution of doubts.

Outside classroom activities (up to 2h)

- Review of the following concepts: recursion, recurrence relation, recurrence equation, recursive method, recursive sequence, trivial case, and general rule.
- Deliverable #1.1 – Outcome of the first phase of the puzzle 1 –

  Individually, each student has to prepare the solution of the problem stated and worked in class. Remember, each member of a workgroup should work with a different problem and prepare two copies of the solution. One copy must be handed in at the beginning of the next session and the other one will be used in the expert meeting.

Second session

Classroom activities (1h 30')

Minutes	Activity
5'	Handing in of the deliverable #1.1 (Save a copy for the expert meeting)
25'	Expert meeting. – Second phase of the puzzle 1 – Those students who have solved the same problem meet in groups of four or five members for sharing their solutions, helping themselves answering questions raised individually, interchanging ideas, and resolving together the problem in order to obtain a better solution. Teacher solves doubts to expert groups.
30'	The base group meets. – Third phase of the puzzle 1 – During ten minutes each member explains to the other members of the workgroup the solution to the problem solved by himself. We recommend to use the improved solution generated in the expert meeting.
15'	Additional time for the third phase of the puzzle 1. Each group prepares a report with the solution to the three problems, i.e., the Java code of each recursive method with the appropriate comments. This report will be handed in to the teacher at the end of the class, as part of the deliverable #1.2.
15'	Each group evaluates the report of another group in accordance with the following evaluating signature. The solution to each problem is evaluated separately. Teacher will provide sufficient evaluating signatures. Handing in the deliverable #1.2: report evaluated with the evaluating signature.

Concept	Range	Punctuation
The code is well formatted and indented, and it is easy to understand	$0 \dots 10$	
The comments are well placed and explain properly the instructions related to them	$0 \dots 10$	
The solutions run correctly for the trivial cases of the problem	$0 \dots 10$	
The solutions run correctly for several general cases near the trivial ones	$0 \dots 10$	
Total	$0 \dots 40$	

Name of evaluated group:
Name of evaluator group:

Outside classroom activities (up to 3h)

- Solve problems 1, 2, 3, 4 and 5 of the chapter 11 of the book "Empezar a programar usando Java".
- Read the sections of the chapter 11 of the same book related to recursion on arrays. This topic is so easy to understand given the concepts studied in the sessions 1 and 2. Then, sessions 3 and 4 will be dedicated to solve recursive problems.

Third session

Classroom activities (1h 30')

Minutes	Activity
20'	The teacher solves problems 6 and 11 of chapter 11 of the book "Empezar a programar usando Java".
5'	Break.
30'	Students solve problems 10, 12, 13 and 14 of the same chapter of the same book. The teacher helps them.
10'	The teacher solves problem 15 of the same chapter of the same book.
30'	Students solve problems 16 and 17 of the same chapter of the same book. The teacher helps them.

Outside classroom activities (up to 3h)

- Individually or in group, students complete and improve the algorithms worked in class.
- All problems must run in the computer without errors.

Fourth session

Classroom activities (1h 30')

Minutes	Activity
10'	Teacher explains and writes the iterative version of the binary search algorithm.
20'	Students solve the problem 21 of chapter 11 of the book. This problem consists in designing and programming the recursive version of the binary search algorithm. The teacher helps students for solving this problem.
50'	Students solve all sections of problem 22 of chapter 11 of the book. The members of each workgroup have to solve different sections. Sections to be solved must be grouped as follows: $\{a, d, g, j\}$, $\{b, e, h, k\}$ and $\{c, f, i, l\}$.
10'	Resolution of doubts.

Outside classroom activities

- No activities are planned after last session of this topic, consult the teaching guide for the next topic, where you will find activities planned to be performed before the first session.

5 Issues and problems that, minimally, should be studied in this topic

- Design of recursive methods for solving problems. This implies the ability of expressing the recurrence relation and/or the recurrence equation.
- Express the recursive definition of a problem.
- Discover if a recursive method will reach the trivial case or not.
- Analyse the evolution of the call stack for small instances of a problem given the recursive method that solves it.
- The recursive version of the binary search algorithm.
- Searches on arrays using recursive algorithms.

6 Deliverables

Id	Description	Modality	Submission date	Points
#1.1	Solution to one of the three problems proposed: (a) Fibonacci sequence, (b) Sum of Figures of an integer number, and (c) Greatest Common Divisor.	Individually	At the beginning of the second session.	20
#1.2	Report with the three solutions and evaluated by other group.	In group	At the end of the second session.	40

7 Qualification

The grade for all activities performed in this topic will be computed as the sum of the points of each deliverable. The grade of activities performed in group will be the same for each member of the work group. Particularly, the mark obtained in the puzzle 1, $MP1$, is computed as follows:

$$GM = (MOI_1 + MOI_2 + MOI_3)/3$$

$$MP1_i = (GM + MOI_i)/2$$

$$MP1_i = \begin{cases} MP1_i += 0,1 \cdot TP & \text{if } GM \geq 0,6 \cdot TP \\ MP1_i & \text{if } GM < 0,6 \cdot TP \end{cases}$$

where MOI_i is the mark obtained individually in puzzle 1 by the i-th member of the group, calculated as the sum of the marks of deliverables #1.1 and #1.2, GM is the group average mark, $MP1_i$ is the mark assigned to i-th member, and TP is the total amount of points which can be obtained in puzzle 1. Notice how $MP1_i$ is increased if the average grade is good enough.

The knowledge of this subject will be assessed in the quizzes and in the lab practises, the grade obtained on the deliverables described herein is used to measure the work of formation, and is part of the NAS (Grade of the Follow-up Activities, the Spanish acronym is used), which, in turn, as said in the evaluation rules, contributes to the final score with a 20%.