Proceedings of the 10th International Conference on Applied Informatics
Eger, Hungary, January 30–February 1, 2017. pp. 279–284
doi: 10.14794/ICAI.10.2017.279

Guess the code of conditional summation

Zsuzsanna Szalayné Tahy

Eötvös Loránd University, Faculty of Informatics
sztzs@caesar.elte.hu

Abstract

There is a question in many countries whether ICT and application usage should be taught. Many suggested to learn programming for some practice in computational thinking. But learning programming is very difficult for many people. In this publication a method is suggested how to teach computational thinking and how to prepare students for programming. The focus is set on a typical problem, conditional summation, but the method is more general: the teaching of programming is embedded into the teaching of application usage [4, 5, 6]. While analysing the possible solutions of conditional summation, computational thinking and programming skills are developed. Almost a dozen methods are known to solve conditional summation. One of them is to choose an imperative, procedural programming language and write the code of structured programming algorithms. Other tool could be a database management system where users write declarative SQL query or visualize the query on a QBE grid, both are based on mathematical abstraction of databases[1, 2]. The solution methods mentioned are used at a higher level of informatics and it is closer to the hardware abstraction. In everyday practice and at earlier stage of informatics study a spreadsheet application gives several tools for solution. Guess the code to understand why there are so many tools to solve the same problem! This is a way of improving computational thinking and understanding the concepts of procedural and functional programming.

Keywords: teaching methods, spreadsheet's tools, database, algorithms, computational thinking

MSC: 97Q30, 97Q60, 97R99

1. Tools to solve a conditional summation problem

In public education, database management tasks are solved mostly in spreadsheets, as it is a well-known software and satisfies personal needs. We have at least five tools in the most popular spreadsheets to solve a simple conditional summation problem. In the public education summation means five counting methods: sum,

counting, average, maximum and minimum. In this article, all of these are represent by SUM.

The most simple way to solve a conditional summation problem when the problem is divided into two parts: conditional function (typically $IF()$) shows the proper value and leave empty cells for the remaining in a set of auxiliary cells that are summed up in the next step. Shortly:

$$\textbf{if-sum}: \text{ auxiliary cells}: IF(condition, value, "") \Rightarrow SUM(auxiliary\_cells) \tag{1.1}$$

To eliminate auxiliary cells, array formulas can be used.

$$\textbf{array formula}: \{SUM(IF(condition, value, ""))\} \tag{1.2}$$

To solve the most frequent problems there are some combined functions in spreadsheets:

$$\textbf{sumif}: SUMIF(range, criteria, sum\_range) \tag{1.3}$$

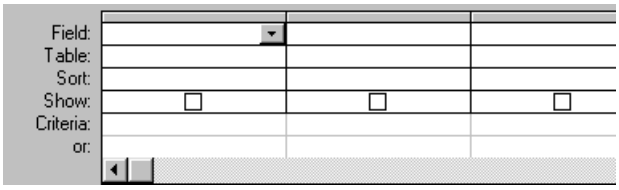
Nowadays more and more "IFS" functions become available:

$$\textbf{sumifs}: SUMIFS(sum\_range, range1, criteria1, range2, criteria2, \dots) \tag{1.4}$$

Following the concept of databases, conditional summation can be solved using database function:

$$\textbf{dbsum}: DBSUM(database, sum\_field, criteria\_range) \tag{1.5}$$

But big databases, mostly hosted on the internet, raise the need of knowledge of database managers what gives two more types of solution. Query by Example (QBE) grid gives a visual solution:

qbe: 

(1.6)

The using of SQL means programming in a declarative programming language:

$$\textbf{sql}: Select\ SUM(field)\ From\ table\ Where\ condition \tag{1.7}$$

According to the National Curriculum and the challenges of XXI century, the well-trained matriculated students are expected to write a program what solves a conditional summation problem.

$$\textbf{code}: SUM = 0;\ foreach\ (a\ in\ array)\ if\ (condition)\ SUM += f(a); \tag{1.8}$$

2. Teaching practise of conditional summation

Though, conditional summation arises frequently in everyday practice, the core curriculum does not grant time for practice. Only the (1.1) **if-sum** method is taught within the general framework of public education.

The ECDL and the final exam of public education on normal level require the knowledge and use of basics. The curriculum includes the use of (1.1) **if-sum**, (1.3) **sumif** and (1.6) **qbe** tools. The computer driving license doesn't mean to understand what that computer does it focuses on the case of use only.

Matriculated students, who has passed final exam in Informatics on higher level are expected to know many solutions of conditional summation. They are expected to know (1.7) **sql** and (1.8) **code** above the basic level. Implicit expectations are the usage of (1.4) **sumifs** and (1.5) **dbsum** but complex tasks show the difficulties in problem-solving. The huge difference between core curriculum and final exam implies the low success in innovative usage. Students learn rules of solution without understanding them.

The use of (1.2) **array formula** is not expected in public education but it is very useful on competitions.

The teaching sequence of different solutions depends on priority of knowledge. A common problem in teaching is that the last topics are not taught due lack of time. In the table 1 the sign "?" shows that the tool is mentioned but not taught, "???" signs the topics planned but not taught.

	Theme-oriented	ECDL	Exam on higher level	Prof. training	*Tested*
if-sum	1	1	1	1	*1*
array formula	-	-	-	-	*2*
sumif	2	2	2	2	*2*
sumifs	2?	2?	2?	2?	*2*
dbsum	3	4???	5???	5?	*3-5?*
qbe	4??	3	3	4?	*3-4*
sql	5???	5???	4??	3	*4-3*
code	independent or missing				*included*

Table 1: Curriculum Structure

Two observations can be made:

1. Programming is totally independent from using spreadsheets and database manager.
2. Array formula is neither taught, nor mentioned.

I have taught array formulas in my classes for 10 years at least. I do not know, how talented my students are but they perform significantly better in informatics exams and competitions than others. The main differences are:

1. I teach in parallel the solutions of conditional summation in spreadsheets. By showing 3-4 methods to solve a problem I let students choose their most preferred method. Later they have to try two or three methods to solve the same problem...

2. I teach programming in parallel with repeating the problem-solving in spreadsheets and database manager. Student have to solve the same problem using spreadsheets, database manager and they have to write a solution in a learned programming language. There are some printed tasks that are very usable for parallel teaching, for example "Europe" in [3], but conditional summation is the most typical task for exploring the solution in different environments.

Solving one problem on different ways gives more experience in each method, improve thinking skills. To guess the code means to understand the similarities and the reasons of differences.

In many cases, parallel teaching of programming and spreadsheets are not recognised by students. For example: while teaching $IF()$ function students explore the result in cases when the condition is a value, like 0, 3, -2, 2010.06.15, "*hello*". They learn the differences between the simple and the composed value; the logical interpretation of 0 and not 0 values. "Guess the Code" in teaching practice means that students explore the tool they use. They learn programming concepts and principles while they solve a problem – in this case – while using a spreadsheet function.

3. Programming aspects of conditional summation

For many students, an **array formula** is the magic use of functions what could solve all problems. It is a compact version of elementary solution and it helps to understand the usage of **sumif** Moreover one has to guess the rules of the evaluation of expressions.

1. How the program understands a written logical expression?

2. Why is it not allowed to write logical expression in **sumif**?

3. What is the differences between the expressions *condition* and *criteria*

4. Why are the range separated from the criteria by programmers in **sumif** and in **sumifs**?

5. Which tools are able to evaluate a relation between two cells or two fields? (For example, which tools are capable to count dancing pairs where boys are taller than their partners?)

6. Seeing that the order of parameters in **sumif** follows the rule of conditional expression "condition precedes the consequences" arises the question "why is the *sum_range* the first in **sumifs**"?

7. ... and why is *database* the first in **dbsum**? Is *database* really the first?

8. Which tool is universal? What are the specialities of the learned tools?

Teachers have to know that answers derived from different models of the conditional summation problems. While students try to guess the answer, they explore possible models of solving the problem. This cognitive activity results progress in programming skills. On the other hand it is very difficult to remember rules and syntax without concept. Therefore, students who explore the code become better in the use of spreadsheets and database application.

To prepare for higher level of computational thinking and abstraction skills, it is suggested to ask: Why should one press $Ctrl+Shift+Enter$ at the use of **array formula**. Depending on the interest of students the static and dynamic memory allocation can be mentioned as well as the use of pointers or references.

While solving conditional summation problems users have to tell the software what the condition is. Different solving methods expect different codes. Users write a string or strings that turns into a logical expression by the parsing and results a value by the evaluation. Students could gain experience in the use of prefix and infix formulas, in the use of logical values and Boolean algebra, classification of logical expressions by complexity.

Students don't have to learn what conjunctive query means but they have to know what kind of problems are solvable using **sumifs**. They have to understand without knowing the term, that **dbsum** functions and **qbe** prefer disjunctive normal form while **sql** and **array formula** let users write any kind of logical expression. ... with the all risk of highly complex expressions.

4. The profit

Solving conditional summation problems on different ways is one of the best practice to prepare learning programming. Though most of students do not understand each solution but by working with different solutions let them explore models to understand some concepts, accept (or reject) paradigms, imagine ideas. In short: students get practice in thinking. This teaching method could motivate students to see what is under the tip of an application's iceberg. Last but not least: students learn programming without noticing it.

References

[1] Abiteboul, S., Hull, R., Vianu, V. Foundations of Databases: The Logical Level, *Addison-Wesley Longman Publishing Co., Inc.* (1995).

[2] Gajdos. S Adatbázisok, *BME*, (2015).

[3] Reményi, Z., Siegler, G. Szalayné Tahy, Zs. Érettségire felkészítő feladatgyűjtemény, *NTK*, ISBN:978-963-19-6783-8 (2011).

[4] Szalayné Tahy, Zs. Teaching Programming Indirectly with "Paint", *ISSEP 2015 proceedings*, University of Ljubljana Faculty of Computer and Information Science, (2015), 54–55.

[5] Szalayné Tahy, Zs. How to Teach Programming Indirectly – Using Spreadsheet Application, *Acta Didacta Napocensia* Vol. 9 (2016), 54–55.

[6] Szalayné Tahy, Zs., Czirkos, Z. Linear Search – the Breaks in Teaching-Learning Practice *Proceedings of XXIX. DidMatTech 2016, "New methods and technologies in education and practice" Conference* (2016), 63–71.