Work in Progress: Space Complexity in CS1: Difficulties in Perceiving the Concept

Tamar Vilner & Ela Zur
The Open University of Israel,
Raanana, ISRAEL 43107 tami@cs.openu.ac.il, ela@cs.openu.ac.il

Abstract - We all agree about the importance of teaching the efficiency of algorithms in the early stages of the study program in Computer Science. Nonetheless we recognize the difficulties encountered when introducing this concept. Usually emphasis is on time complexity, while space complexity is only mentioned briefly. In this paper we describe a misunderstanding that we encountered when teaching space complexity in the "Introduction to Computer Science" (CS1) course at the Open University of Israel.

Index Terms - CS1, Misconception, Space Complexity.

INTRODUCTION

The design of efficient algorithms to solve algorithmic problems is one of the important research fields within computer science [3, 4, 5, 6].

From Computing Curricula 2001, we learn that a large part of the core and elective course material is devoted to algorithms. Algorithm complexity is measured in terms of space and time. Space complexity is measured by elements such as the number and size of the data structures used; while time complexity is measured by the number of elementary actions carried out during the execution of the algorithm.

Though time and space complexity are fundamentals concepts, they are difficult to perceive. Students first become familiar with these concepts in the introductory course. When teaching CS1 at the Open University of Israel (OUI), we found that students had difficulty understanding the notion of algorithm efficiency and its implementation [1].

The OUI is an institution of higher learning with an open admission policy which is based solely on distance education. Because of its open admissions policy, the first courses serve as "the proof of the pudding". Many students, unfortunately, fail.

INTRODUCTORY COURSE IN CS: A CASE STUDY

At OUI, CS1 is similar to introductory courses given at other universities, and includes the topics recommended in Computing Curricula 2001.

According to Computing Curricula 2001, it is essential to teach the concept of complexity in CS1, therefore we introduce it as early as possible. The relatively early introduction of the concept encourages students to consider alternative designs of algorithms, to analyze various algorithms, and to formulate them correctly [4, 5].

However, such early introduction may lead to difficulties: the problems discussed at early stages of the introductory course are almost always toy problems, making it difficult to convince students that a more efficient algorithm is indeed needed. Also, the analysis of algorithm efficiency requires mathematics that students are not always familiar with when they take the introductory course. Misconceptions are encountered [2, 3]; for instance, students often bring up the myth of the speed and the growing capacity of the computer, saying that computers are so incredibly fast and small that there is no real problem.

When teaching the CS1 course, we found that many students have difficulty measuring space complexity. First, we looked for published studies on teaching complexity in general, and on space complexity in particular. We found papers that deal with misconceptions of fundamental concepts in Computer Science, but we found nothing regarding space complexity. We decided to conduct a study which investigates how CS1 students internalize space complexity. In the next section we will describe our study.

THE STUDY

We posed two main research questions:

• To what extent are students successful in analyzing the space complexity of a given program?
• What are the difficulties in perceiving the space complexity concept?

Our study was carried out on 247 CS1 students during the fall semester of 2004. Only 116 students (47%) passed the final examination, probably due to the university’s open admission policy.

The students could take one of two final exam sittings, each of which included a question relating to space complexity. We present below only one question.

The Question

Suppose mat is a given square matrix (2 dimensional array) of length n × n (n is a constant). Each element of mat is an integer.

• Write an efficient function which gets a square matrix mat filled with integers, as a parameter, and returns the Big-of-Small value in the matrix mat.

• What is the time complexity and the space complexity of the function you wrote?
For part a, students gave two solutions:

119 students took the exam which includes the question. O

The 57 students (98%) answered correctly on time complexity - and space complexity of their solutions. We found that 56 of the 44 students out of 57) gave this solution.

As for section b, we asked the students to write the time complexity both in the lectures and in the exercises. We intend to interview students, in order to learn why they distinguished between measuring time and space complexity.

Program - Code for the best solution to the question.

```c
int bigOfSmall (int mat[n][n]) {
    int max = mat[0][0];
    for (int i=0; i<n; i++) {
        int min = mat[i][0];
        for (int j=1; j<n; j++)
            if (mat[i][j] < min)      
                min = mat[i][j];
        if (min > max)            
            max = min;
    }
    return max;
}
```

We noticed this incorrect way of thinking when we taught the issue of binary search in a sorted array. The students had no difficulty understanding that the time complexity of the binary search algorithm is logarithmic (O(log₂ n)). When they measured the time, they didn't add the time for inserting the data to the array or the sorting of the array. They understood perfectly that they have to compute only the number of actions that the binary search algorithm performs. All the other actions (like inserting the data and sorting the array) are already given and that's why they are inherent to the problem. But when they had to calculate the space complexity, although the algorithm itself uses only few variables and its space complexity is fixed, the students thought that they have to add to this calculation the array itself (which is linear).

On their exam papers, we found the same misconception. On the first exam, 28% of the students who passed the exam said that the space complexity was O(n²), since they computed the matrix size as well.

Teaching efficiency in CS1 is only an introduction to a very important topic in CS studies. We believe that time should be dedicated to analyzing space complexity and not only to time complexity.

As a result of this study, we changed the way we teach efficiency in our CS1 course. We put more emphasis on space complexity both in the lectures and in the exercises. We intend to continue examining students’ perceptions of this concept, and to see if the change helped them to understand. Another option is to interview students, in order to learn why they distinguish between measuring time and space complexity.

REFERENCES