ACTIVE LEARNING IN LARGE CLASSES

Inge Li Gørtz

DTU Informatics, Technical University of Denmark

ABSTRACT

In this paper we present our ideas of how to use active learning in the lectures when teaching large classes (more than 50 students), and describe how we successfully have in a second semester course in the Bachelor of Engineering (BEng) and Bachelor of Science Engineering (BSc Eng) program at the Technical University of Denmark (DTU). Approximately 200 students is attending the lectures in the course.

The main idea is to use inductive, case-based learning, with many small exercises/ discussions during the lectures. We describe a framework for the lectures, that most lectures in the class were based on. The framework contains the conceive, design, and implement stage from the CDIO principle.

KEYWORDS

large classes, active learning, lectures, inductive teaching.

INTRODUCTION

When giving lectures it is necessary to keep the students active. This improves the students learning outcomes. As it says in Standard 8 of the CDIO standards, active learning in lecture-based courses engage the students and can include methods such as small group discussions. It is our experience, that many lecturers find it difficult to implement active learning in large classes (classes containing more than 50 students). We give a description of how this can be implemented, and a way to structure the lectures using active learning and inductive teaching. The use of inductive teaching and active learning is of course not new. What we try to convey here is how to use it in lectures for large classes. We also give a concrete way to structure the lectures, that incorporate both inductive teaching and active learning.

CONCEPTS

In this section we describe our ideas. In the first subsection we describe how we get the students to actively take part in the lectures. The second subsection contains our framework for the lectures.

Active Students

To keep the students active during lectures, they are given small exercises that they can solve either by themselves or together with the students sitting next to them. They are given 5-10 minutes to solve the exercises, and then their solutions and ideas are discussed in plenum. The motivation for this is

- It forces the students to think and get in touch with the material during the lectures. For example, the students appreciate a solution more if they have tried to solve the problem by themselves first.
- It gives the students time to get some understanding of the material before we proceed. E.g, if two new concepts/definitions are going to be compared in the lecture, the lecturer can ensure that the students really understand both definitions before they are compared. Another example is to let the students try to run an algorithm on an example before analysing it.
- It gives the lecturer a chance to see what the students find easy/difficult during the lecture, and thereby an opportunity to adjust during the lecture.
- When the students are allowed to talk to each other and solve the exercises together, they are more inclined to answer/participate in the discussion afterwards (this can otherwise be a big difficulty in large classes).

Structure of Lectures

We have developed the following framework that the lectures are build around:

1. **Lecturer:** Introduction to the problem
2. **Exercise:** Try to solve an example instance of the problem (and try to come up with a general method/algorithm to solve the problem that works on all possible instances).
3. **Discussion:** Discussion of the students solutions and ideas.
4. **Lecturer:** Explain solution/algorithm + give an example.
5. **Exercise:** Run the algorithm/use the method on a new example.
6. **Lecturer:** Guide to analysis of the method/algorithm (leading questions).
7. **Exercise:** Show/find properties of the method/algorithm.
8. **Lecturer:** Put together the properties of a proof.

It is a mix of standard lectures and exercises. The exercises/small group discussions are progressing during the lecture. Starting with small examples and ending with questions that lead to a mathematical proof. In the last exercise (7.) the students are given questions in an order that also show how to build up a mathematical proof.

PROOF OF CONCEPT

In this section we describe how we have used this concept in a large class at DTU for the last 3 years.

Background

The class of study is a 2nd semester introductory algorithms course at DTU. Students from many different study lines participate in the course, but it is mandatory for BEng students in ”IT” and BSc Eng students in “IT and Communication Technology” and “Software Technology”. The teaching consisted of 1 1/2 hours of lectures followed by 2 hours of exercise classes in smaller groups/classes (around 30 students in each exercise class). Approximately 200 students are attending the lectures. Since the students come from many different study lines, they have very different prerequisites.

Evaluation

The described lecture form has been used for 3 years now at the course, and the students are very positive. In the midway evaluation the students were asked whether they thought that the small exercises were helping them to understand the material covered at the lecture. This year 76% answered to a high or very high degree. If we look at the results for all 3 years between 57-76% answered to a high or very high degree, and only around 1-4% didn’t think they helped at all.

SESSION AT THE CONFERENCE

In the freestyle session at the conference, we will give a concrete example of a lecture from the course. The example will be from the first lecture in the course, and does not require any previous knowledge about algorithms.

Biographical Information

Inge Li Gørtz is Associate Professor at DTU Informatics.

Corresponding author

Inge Li Gørtz
Technical University of Denmark
DTU Informatics
Building 322, Office 128
DK-2800 Kgs. Lyngby
Denmark
Phone: +45 45 25 36 73
Email: ilg@imm.dtu.dk