INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Flipped classes for algorithm learning

Luka Gorgadze

Work developed under the supervision of

Rui Pedro Lopes

Koba Gelashvili

Master in Information Systems

2016

Flipped classes for algorithm learning

Dissertation of the Master in Information Systems

Escola Superior de Tecnologia e de Gestão

Luka Gorgadze

2016

A Escola Superior de Tecnologia e Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Dedication

To my father Vasili.

Resumo

Nos últimos anos, tem-se vindo a assistir ao aparecimento de várias técnicas e abordagens pedagógicas que procuram incrementar o papel ativo dos alunos no próprio processo de aprendizagem. Uma das abordagens, designada por aula invertida (do inglês flipped classes), incentiva o aluno a preparar a aula antecipadamente, por intermédio de vídeos, conteúdo teórico e problemas para resolver. Há alguma investigação em torno desta abordagem, com resultados positivos.

O objetivo deste trabalho é contribuir para a investigação deste tipo de abordagem, investigando trabalho relacionado e comparando a eficiência das aulas invertidas com as aulas tradicionais. A hipótese colocada é que as aulas invertidas constituem um método eficiente para a aprendizagem de algoritmia.

Os resultados obtidos foram positivos, o que confirma a recomendação feita em certos trabalhos relacionados no sentido de adotar aulas invertidas em algumas áreas. No entanto, os resultados não assinalam uma diferença considerável com as técnicas tradicionais, provavelmente devido ao facto de a experiência decorrer durante duas semanas apenas, não dando tempo suficiente para os alunos se ambientarem e incorporarem a filosofia desta abordagem.

Palavras-chave: Aulas invertidas, uso da tecnologia no ensino, aprendizagem de algoritmia

რეზიუმე

ბოლო წლებში განსაკუთრებით გამოიკვეთა ისეთი პედაგოგიური მოდელების არსებობის და გამოყენების საჭიროება, რომლებიც განაპირობებენ სტუდენტის აქტიურ ჩართულობას სასწავლო პროცესში. ერთ-ერთი ასეთი მოდელია „შებრუნებული საკლასო ოთახი“ (The Flipped Classroom), რომელიც საშუალებას აძლევს სტუდენტებს მოემზადონ ყოველი შემდეგი ლექციისთვის წინასწარ, ვიდეო ლექციის და სავარჯიშოების საშუალებით. ამ მოდელის გარშემო არაერთი კვლევა ჩატარდა ბოლო ხანებში, რომლებიც ძირითადად დადებითად აფასებენ მას.

ამ ნაშრომის მიზანია გარკვეული წვლილის შეტანა შებრუნებული საკლასო ოთახის გარშემო მიმდინარე კვლევაში ექსპერიმენტის შექმნით და ჩატარებით, რომელიც შეადარებს ერთმანეთს შებრუნებული და ტრადიციული სწავლების მოდელების ეფექტურობას. კვლევის ჰიპოთეზა მდგომარეობს შემდეგში, შებრუნებული საკლასო ოთახი ეფექტური მოდელია ალგორითმების სწავლებისათვის.

კვლევამ პოზიტიური შედეგები აჩვენა, თუმცა არცისე მკვეთრი. ამის მიზეზი ისაა, რომ ექსპერიმენტის ხანგრძლივობა მხოლოდ ორი კვირა იყო. რაც არ აღმოჩნდა საკმარისი იმისათვის რომ სტუდენტებს აეთვისებინად ის თუ როგორ უნდა ისწავლონ შებრუნებული საკლასო ოთახის მეშვეობით. აქედან გამომდინარე ამ სწავლების მეთოდის სრული პოტენციალის დასადგენად რეკომენდირებულია უფრო ხანგრძლივი ექსპერიმენტების ჩატარება.

Keywords: შებრუნებული საკლასო ოთახი, ტექნოლოგიის გამოყენება სწავლებაში, ალგორითმების სწავლება

Abstract

Recent years have brought the need for new pedagogical approaches, that appeal to the involvement and participation of students in the learning process. One of this approaches is the flipped classroom, which gives to students the possibility to prepare for the next class, through pre-recorded video lectures and close-ended problems. There is some research going on around this model of teaching-learning, showing some promising results.

The purpose of this work is to contribute to this line of research by designing and applying an experiment to compare the efficiency of the flipped classes methodology to the traditional classes. The research hypothesis is that the flipped classes methodology is an efficient method for teaching algorithms.

The results were positive, although mildly. The reason for this was the fact that the experiment lasted only two weeks, not giving enough time for students to fully understand how to learn by the flipped classes and to get used to it. Thus, longer experiments are recommended in order to check full capacity of this method.

Keywords: Flipped classes, technology in learning, algorithm learning

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

Emerging technologies brought educators many new tools that facilitate teaching. This tools and other associated benefits can be used to make the teaching-learning process more efficient. Nowadays, using these possibilities in order to improve learning process is important, because alongside with the development of technology and science, in many areas of education content to be learned in order to become a professional has grown significantly.

Educational researchers work hard to create teaching models that use the possibilities created by technological advances. In recent years one of such models, the flipped classroom, became quite popular. According to this model, students try to learn new material beforehand an home, through pre-recorded video lectures and close-ended problems, while in the traditional classroom they do problem-based learning activities, ask questions about new material and participate in discussions. Research around this methodology of teaching generally has shown positive results and many of them recommend its implementation in educational institutions. However, there is still not enough evidence to establish this opinion, therefore, further research is required.

1.1 Objectives

The purpose of this thesis is to contribute to this line of research around the flipped classroom by exploring different research works on this topic and by designing and implementing an experiment to check the efficiency of the flipped classroom when compared to the traditional classroom.

Besides, introduction to the algorithms and data structures seems to be quite problematic course for students to master and sometimes even just to pass. This is definitely one of such courses that need help from technology to make comprehension of it easier. In addition, this course of algorithms and data structures has several characteristics that make the flipped classes especially suitable for it. Therefore, we decided to design our experiment so that it evaluates how efficient are flipped classes compared to traditional classes specifically for teaching algorithms.

1.2 Structure of the document

After this introductory chapter, the second chapter discusses the importance of learning algorithms and the problems that students face while learning them. Further, it describes the flipped classes approach, reasons that lead to its development and how it works. This is followed by the description of its benefits specifically for algorithm learning and the description of several implementations of the flipped classroom by different teachers and institutions. The last section of the chapter describes different supplementary resources that can be used while implementing flipped classes.

The third chapter is about experiment that was designed for this thesis. It describes its objectives, possible subjects, elements and how it should be implemented.

The fourth chapter discusses the application of this experiment in the Polytechnic Institute of Bragança and its results.

The last chapter presents conclusions made from exploring different research works around the flipped classes and from conducting the experiment.

Chapter 2

2 Pedagogical Strategies

This section presents the flipped classes pedagogical approach and the importance of learning algorithms and data structures for computer science students. Some of the problems they face while learning this subject are also described.

2.1 Algorithm learning

Learning algorithms and algorithmic thinking is an essential part of becoming a professional computer scientist. There are many reasons why one should study algorithms. Firstly, their impact is very broad and far reaching. Algorithms are used in biology (human genome project, protein folding, …), physics (N-body simulation, particle collision simulation, …), multimedia (MP3, JPG, face recognition, …), security (e-commerce, voting machines, …), computer graphics and in many other fields (Figure 1).

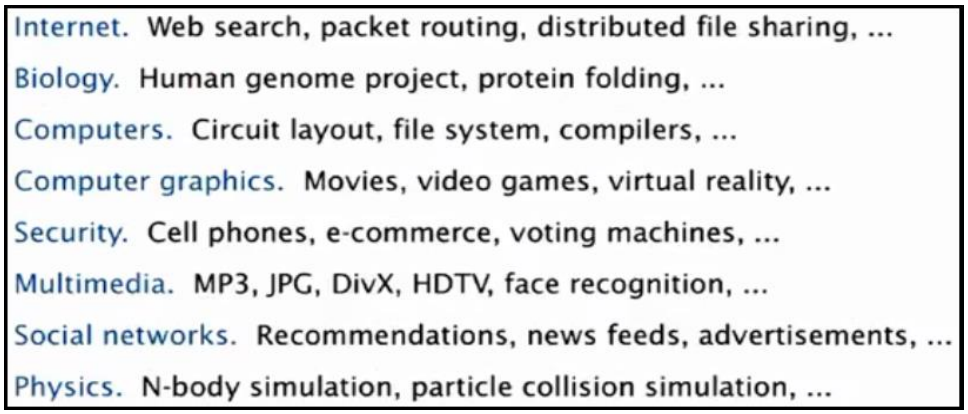

Figure 1 - Examples of applications of algorithms.

They give us possibility to solve problems that would otherwise be difficult to address. Secondly, it is necessary to understand algorithms and data structures in order to become a proficient programmer. Creator of Linux, Linus Torvalds says “difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships”. Thirdly, algorithms have become a common language for understanding natural phenomena. Algorithmic, computational models are replacing mathematical models for studying nature because it turns out its difficult to find solutions for this type of mathematical models in order to test different hypotheses. By computational models it is possible to simulate what might be happening in nature and as result understand it better. In addition, algorithms are good for intellectual stimulation and they are very interesting subjects to study. Donald Knuth says “Algorithm must be seen to be believed”, while Francis Sullivan says “The great algorithms are the poetry of computation. Just like verse they can be terse, allusive, dense and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing” [1, 2].

Because of their importance and because algorithms are prerequisite for many courses, in most of the schools they are taught alongside with programming at the beginning of computer science course. This courses, for introduction to programming, algorithms and data structures are characterized universally with high drop out and failure rates, it seems they are very hard for most students to comprehend. There are various reasons for it. Firstly, it seems the biggest problem lies in the lack of general problem solving and logical reasoning abilities in first year students, accompanied with the lack of mathematical knowledge. This are skills that student entering high education course is expected to have, but studies show that this often is not the case [3, 4]. Another reason for failure in these courses is use of incorrect study methodologies by students. Many of them are used to solve problems simply by memorization of formulas and procedures, sometimes without understanding concepts behind them. This method does not work for algorithms and programming learning, subjects that require abstract thinking and problem solving abilities. Main activities for learning this subjects should be practical ones. For example, solving as many programming problems as possible and tracing different algorithms for different inputs in order to understand how they really work. One more reason that makes this courses so hard for the freshmen is that teaching is not personalized. It would have been a great help for many students to have a teacher always available to give immediate feedback during problem solving and to explain aspects that are

not understood by that student, as detailed as needed for him and with speed that is suitable for him. But this type of support is impossible in traditional classroom setting because of time constraints and amount of students in the classroom. In addition, different people learn in a different manners and have different approaches concerning how they study new materials. For example, some prefer to study alone, while others may prefer to study in more dynamic environment such as learning through discussions with peers. In traditional classroom all students learn in the same way and in the same rhythm, needs of different types of learners are not met [3]. One teaching methodology that addresses these problems and tries to improve situation is the flipped classroom approach. How it works and what improvements it offers will be discussed in the next section.

2.2 Flipped Classroom

Before defining what a flipped classroom is, we will discuss some reasons that led educational researchers towards its development. Generally, there are two major areas of development that have big impact on education: technological and ideological development. Technological development gives humans the ability to replicate and spread information in more and more efficient ways. The first breakthrough in this area was the printing press, that was followed by radio, television and lately by the World-Wide Web (WWW). Development of this technologies made fast and free flow of information physically possible, but while physical barriers were removed by technology, still there were man made barriers that kept information from becoming available to everyone. Thus, ideological development was equally important part of this process. For example, even though lots of institutions had ability to share publicly information that they had, they were only making it available to limited amount of subscribers who paid money to them [5].

However, in recent years this man made barriers began to fall, significant steps towards it were advent of Wikipedia in 2001 and in high education area MIT's OpenCourseWare which made widely accessible information that was before only available to students who paid 40 000$ annual tuition. Nowadays, we have several websites that offer high quality courses for free. Courses are from best universities of the world – MIT, Harvard, Stanford, Princeton, etc. [5]. This courses seem to be highly effective. Most of them use video lectures, weekly quizzes and assignments as the means of teaching and evaluating. Recent studies support this means of teaching. Research has shown that video lectures are not just effective, but they are even slightly more effective than in-person lectures and online quizzes and assignments are as

effective as traditional ones [9, 10]. Thus, students nowadays have an alternative way to learn subjects that before were only possible to study through enrolling in a higher education institution. Furthermore, tuition fees for Universities are constantly rising and students protest about what they get from University for the money that they pay, demanding more and better outcomes from schools. Thus, higher education institutions need to change something in order to respond to challenges and competitions of the present day.

Besides, students are not the only ones who demand changes and improvement from Universities. Accreditation institutions also increase their pressure. For example, the Accreditation Board for Engineering and Technology requires 3a-k outcomes to be meet by University programs in order to accredit them. These outcomes include, "an ability to function on multidisciplinary teams", "an ability to communicate effectively" and "an ability to identify, formulate, and solve engineering problems" [5]. Traditional study methods, such as informative lectures and closed form questions are not effective in reaching this goals. Studies show that problem-based learning methods are much more effective in reaching 3a-k outcomes [11]. But the problem is that curriculum is already too demanding to add more working hours to address this needs by specific learning methods. This demands lead to development of the flipped classes that are utilizing technology to make learning more efficient and to automate teaching elements that can be automated in order to free teachers time for student-centered, problem-based learning activities. These activities can be a great help in achieving learning objectives that are crucial for graduate students and are something that they cannot get from online courses [6].

There are a lot of people working on and writing about flipped classroom, but still there is no consensus about its exact definition. Most basic definition is given by Lage et al., who define it by "Inverting the classroom means that events that have traditionally taken place inside the classroom now take place outside the classroom and vice versa" [12]. That is, students learn about new material through the video lectures at home and they do exercises in classroom. This definition reveals what we mean under the word "flipped", but it does not show full capacity of the flipped classroom. Specifically, how it addresses problems mentioned in previous paragraph. In practice there is much more to the flipped classroom then a mere change of places for activities (of lecture and of homework) [5].

As mentioned above, there is no consensus about definition of the flipped classroom, so we will try to describe different teaching elements of the various flavors of it. Commonly, in the flipped classroom model video lectures about new content are assigned as a homework, in

some implementations students also have assigned readings. In addition, they have to do close-ended problems or quizzes in order to check how they understood the new material. In some implementations this quizzes and problems are moved into classroom where teacher can observe students in the process and give them immediate feedback. If there are questions or difficulties, he can assist in class. In many traditional courses informative lectures, close-ended problems and quizzes are all the instruction students get. However, by moving lectures into homework, in the flipped classroom model teachers get more free time with students that they can use to expand curriculum, to employ student-centered, problem-based and active learning methods in the classroom. So for example in classroom students may get open questions followed by feedback from a teacher. Students may also ask some questions about content that they tried to learn from a video lecture. Another activity can be discussions where students can discuss open-ended problems with their peers and share ideas. Also they may get group quiz where students will be encouraged to collaborate and to do quiz together in groups. If significant amount of groups will not get to the correct answer, teacher may ask them to give arguments to each other and to explain why they think their answer is correct. As a result, through discussion with each other and some guidance from the teacher students find out what is correct answer. Besides, it is also common for students to do exercises in the classroom, that are followed by peer grading, critiquing and instructing [5,6,8].

Research around the flipped classroom shows generally positive results and most of the students tend to like this model of teaching, even if there are also few of them who strongly dislike the change. Studies also support individually those means of teaching that are an elements of the flipped classes. For example, they indicate that students like interactive classes more than in-person lectures and that they tend to come in class much more prepared when assigned video lectures, as opposed to textbook readings [13, 14]. Besides, one of the significant trend is that students who learned course through flipped classes tend to score higher on examinations than their peers who were taught through traditional methods [5,6,7,8]. However, still there is not enough evidence to assert this opinion, because there are few studies that having experimental and control sections of the course examined students' performance during the whole semester. Thus, even though results look promising, further research is needed for more evidence.

2.3 Flipped Classroom for algorithm learning

Our Hypothesis is that the flipped classroom is an efficient way for teaching algorithms, when compared to the traditional model. General benefits of the flipped classroom were described in previous sections, here we will discuss its benefits specifically for teaching algorithms.

Firstly, materials that are covered in the lectures of algorithms are dense and for many students hard to understand at the speed that teachers present them. Therefore, from time to time student needs to stop and think on his own to catch up with the lecture, but on traditional lecture such pause is impossible because need for such pause varies from individual to individual. Further, content is sequential and tightly connected to each other, thus, if for a moment student's attention will drift away, he may not understand most of the material from rest of the lecture and just waste time by sitting on the lecture. Both of this problems are solved by the video lectures that are assigned to students as homework in the flipped classroom model. Student can watch video lecture at the speed that is suitable for him, pausing when he needs to think about concept or rewinding if he missed anything.

Another reason why teaching algorithms is efficient through the flipped classes is that many exercises and problems in this field are suitable for automated grading. Thus, homework part of the flipped classroom format, where student should learn new material by watching video lectures and then do exercises and problems that are graded automatically can be utilized successfully. Many exercises can be graded based just on comparison of two strings, that is, students answer and correct answer. This is the method that is used for example in most of the algorithms courses on Coursera website. Besides, programming assignments can be evaluated using auto grader that can assess code thoroughly and at the same time give detailed feedback.

In addition, classroom time of the flipped classes, that is usually used for group-based activities, can also be utilized efficiently for algorithm teaching. There are many advanced problems, for example ones from programming competitions, that are good topic for discussions and for sharing ideas between students in order to go to the best solutions. Students can share ideas how to solve problem in most time or memory efficient way or how to write most compact or easy to understand solutions. Peer instruction can also be effective, because often students skills and understanding speed varies, thus often is the case that some student understands algorithm well, while other gets stuck at some point and needs individual instruction. In such cases peer instruction can be quite productive. In conclusion, the flipped classes are well suited for algorithm teaching, because as examples show materials, exercises

and programming problems of course of algorithms are efficiently taught through elements of the flipped classes like video lectures, auto graded homework and group-bases activities.

2.4 Related Work

In this section will be described three cases of implementing the flipped classroom by different educational institutions. Namely, what elements of teaching they used in classroom and as homework and what were their outcomes.

Basic Pharmaceutics II in the UNC Eshelman School of Pharmacy

The University of North Carolina Eshelman School of Pharmacy implemented flipped classes for the Basic Pharmaceutics II course in the following way. Students had to watch video lectures and read assigned textbook and background readings before the class, while in-class time was devoted to student-centered learning exercises that were created to promote critical thinking, to rise discussions and to assess student's knowledge of the content. Specifically, first in-class activity was "audience response and open questions", which included clicker questions and open questions to students to assess how they understood the assigned content. This was followed by feedback from a teacher and later questions from students. Second type of in-class activities were "pair and share activities", where for example teacher would give discussion question to students and give them time to pair and share ideas. After, some students would share their ideas to the class, this would be followed by teacher giving feedback, perspective and expanded discussion. Third in-class activity was "student presentations and discussion", where group of four or five students had to present a summary and their interpretation of the assigned content and to answer other students questions related to that topic. For each class, up to three groups had to prepare presentation and during the class, presenter group would be chosen by a dice roll. However, other groups were graded also, based on their presentation materials. Forth in-class activity was "individual or paired quiz", which was held at the end of each class, it included 10 multiple-choice questions quiz on paper and it covered only that class's material. Sometimes students were permitted to do it in pairs and openly discuss questions. Last but not least activity was "the microlectures", it was usually one to three minutes in length, incorporated when needed during class to reinforce and in some cases to redirect students learning.

Researchers at the UNC Eshelman School of Pharmacy did several surveys in order to evaluate efficiency of the flipped classrooms and to understand what are students perceptions

about it. They did two surveys before and after the course to see students opinions about active learning activities, about delivery format and about typical engagement behavior. Also, they tracked down standard end-of-semester course evaluation scores and they compared final exam results of flipped course to previous year's traditional one.

From course evaluation scores especially interesting was that 91.2% of students agreed or strongly agreed that "learning materials and resources were helpful" and 93.1% of students agreed or strongly agreed that "teaching and learning methods in the flipped classroom promoted understanding and application of key concepts". Besides, attendance was higher in the flipped classroom. Also, there was dramatic change in preferences of students in favor of the flipped classroom before and after taking the course. Prior to learning through the flipped version of the course 72.7% of students preferred traditional format, with only 27.3% of students preferring flipped format. Postcourse survey, however, showed the opposite results, where 84.6% of students were in favor of the flipped classes and only 15.4% preferred traditional classes. There was statistically significant difference in final exam results too, students from the flipped version of the course scored 165.48 from 200 on average, while students from previous year's traditional course scored 160.06. Additional information about students' perceptions before and after taking the flipped course can be seen in Figure 2.

Survey question	Precourse mean ± SD	Postcourse mean ± SD	*P* value
Lectures greatly enhance my learning/Prerecorded lectures greatly enhanced my learning.	3.29±0.62	3.67±0.53	<.001
Learning key foundational content prior to coming to class greatly enhances(d) my learning of course material in class.	2.85±0.68	3.57±0.63	<.001
Interactive, applied in-class activities greatly enhance(d) my learning.	2.80±1.08	3.39±0.72	<.001
I participate(d) and engage(d) in discussions in class.	2.66±0.71	2.97±0.63	<.001
In-class discussions of course concepts with my peers greatly enhance(d) my learning.	2.53±0.93	3.05±0.76	<.001
I read assigned readings prior to coming to class.†	2.29±0.77	1.67±0.89	<.001
Assigned readings from textbooks/articles enhance(d) my learning.	2.53±0.71	2.17±0.79	<.001

*Data based on 150 pre- and postcourse survey responses. SD indicates standard deviation. Likert scale items measured on a four-point scale ranging from (1) strongly disagree to (4) strongly agree, unless otherwise noted.
†Likert scale items measured on a five-point scale: 1 = never; 2 = rarely; 3 = some of the time; 4 = most of the time; 5 = all of the time.

Figure 2 – Students perceptions before and after flipped format course.

Based on their experience and the feedback of students researchers at UNC Eshelman School of Pharmacy think that flipping the traditional classroom is a feasible and even necessary in order to educate a large group of students. Thus, they will continue conducting Basic Pharmaceutics II course in the flipped format. However, they made some changes, for example, they made textbook no longer a required reading because many students thought it

was redundant and also they moved graded quizzes outside classroom time, to be done online through the course web site. Besides, they developed online 411 Pharmacopedia as an information portal for expanding knowledge. They hope instructors in the other schools will also use UNC Eshelman School of Pharmacy's model of the flipped classroom to reinvent their classrooms and as a result help their students to gain higher-order cognitive skills and engage them in a meaningful learning [6].

Math courses in Minnesota's Byron High School

Another example of implementing the flipped classroom is Minnesota's Byron High School's math courses. Math teachers from this school developed their own digital curriculum with video lectures, tests and worksheets. In addition, they have created library of all BHS math teachers course videos so that students could choose from which teacher they would learn that days content. Most students were choosing their own teacher, but some also watched videos from other teachers to see same concept from a different angle.

During the course, students have to watch video lecture about new material before each lesson. In the classroom they have free environment to study, some review video, some work on exercises, some go through exercises fast and move to night's assigned video. Some students work in groups, while others prefer to sit alone, few students even sit on the floor or in the hall with earbuds to block out everything and everyone around them. The teacher is moving from student to student, watching, listening and helping to those who need help. If several students get stuck on the same problem he will go through exercises related to it on the board to make problem clear to everyone. Later students have quiz, often using clickers. After getting immediate results teacher gives students feedback that is followed by group discussions and peer instruction on the problems that many students are finding difficult.

Concerning results, it is hard to separate effects of digital curriculum from the effect of the flipped classroom model. However, combined effect looks promising. Calculus and pre-calculus proficiencies (students with 80% or more score on unit's assessments) increased on average by 9.8% and 6.1%, respectively, compared to previous years traditional format course. Detailed comparison of traditional teaching model versus flipped model for teaching pre-calculus course can be seen in Figure 3.

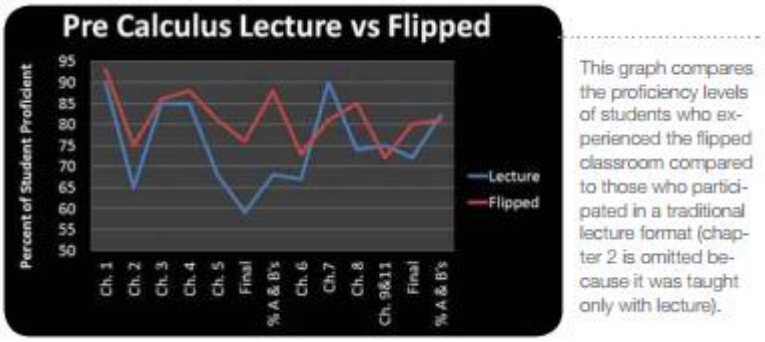

Figure 3 – Pre-calculus traditional vs flipped format.

Besides, there was a significant increase in student's scores on the external exams. BHS math mastery level rouse from 65.6% to 73.8%. Student's and Parent's feedback was generally positive, but as with any new practice there were some that didn't liked the change. Based on positive results in the flipped classes of math courses, other departments and grade levels at BHS are also creating their versions of the flipped classes. Further, since teachers have class time freed from lectures, they are developing open-ended, cross-curricular projects to engage students and to teach math through real-life problems [8].

Introduction to Media Computation course from The University of North Carolina

The last example of the flipped class implementation that will be considered is ITIS 1212: Introduction to Media Computation course from The University of North Carolina. In this course students are required to read the textbook, enter textbook programs, watch video lectures and to do short online quiz before they will come to the classroom. They have one class on Wednesday where students do programming exercises in same-gender peers (reason for same gender peers is that sometimes guys take over when paired with a female and are doing most of the work by themselves). Lab partners are changed in every two weeks. Further, they have workshop using lightweight teams. Lightweight teams are teams of students working together, but where their work as team has minimal impact on the final grade of each student. Thus, while it gives students opportunity to know each other, become comfortable and learn from one another, it doesn't have drawbacks that normal teams have. Namely, in light weighted teams students don't have stress associated with high-stakes group work, reason of which usually is a fact that some members of the team don't do their fair-share of work. On Friday workshop class starts with student's questions about a course content. After which the clicker quizzes are conducted, it consists of multiple choice questions, students are encouraged to talk with their team mates before answering. Responses are evaluated after each question, if instructor feels that not enough students are answering correctly, he asks students to find a person with other answer and convince them that they are

right, this leads to teams discussing with other teams. This quizzes worth only 10% of student's final grade but the feedback from students shows that team based quizzes are one of the most useful part of testing and developing their understanding of the core programming principles. On the second half of the class students work on hands-on activities with their team. Usually, this are paper problem solving activities, where students should write pseudocode or Parson's Problems, where given lines of pseudocode they should put them in correct order [18]. Further, sometimes teams are asked to trade solutions and do pair grading or pair critiquing. Other times this second half of the class is devoted to one of the five tests which students take individually. For increasing motivation and engagement light weighted teams are complemented by gamification elements. Namely, by stamps, leaderboard and question tokens. Students get stamps for non-mandatory, bonus activities, for 10 stamps they get 1% grade increase. In order students to encourage each other teacher is using a leaderboard where teams are ranked by amount of stamps collected by all of the team members. Further, each week student is given 1 question token. Question tokens are items that can be used by students to ask question during the Friday lab or they can exchange them into a stamp at the end of the lesson. But if student will ask question and teacher will determine that he or she should have known an answer from his weekly preparation, then question token will be taken from the student and as a result he will not be able to exchange it into a stamp.

In order to understand how this model of teaching fostered learning, researchers compared scores of students who took ITIS 1212 class with students who took ITCS 1212 class. Difference between ITIS 1212 and ITCS1212 was that the latter was not flipped, had no teams or gamification elements and programming language it taught was procedural C++ as opposed to java in ITIS 1212. Scores that researchers used for comparison was from follow-on object oriented Java course ITCS 1213 that students from both pre-courses took. It is hard to tell which elements of class made biggest difference, but generally students who studied through flipped class received better grades (Figure 4) [7].

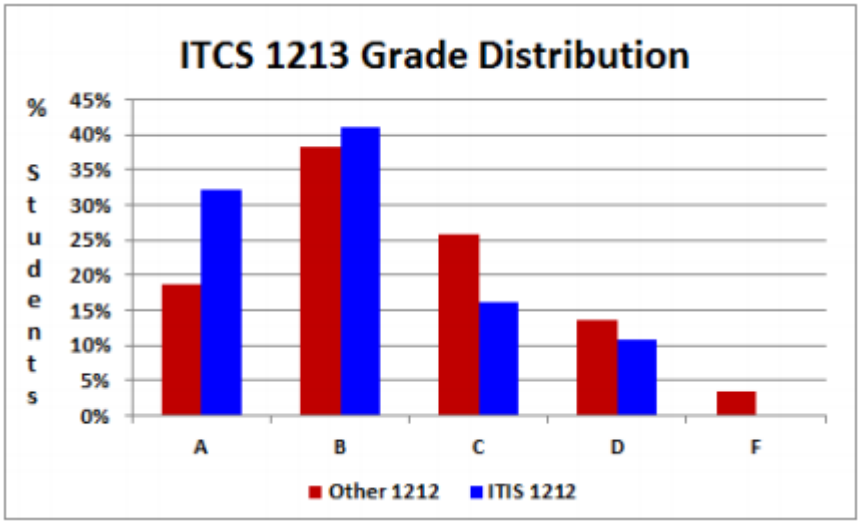

Figure 4 – Traditional ITCS 1212 vs flipped ITIS 1212.

2.5 Supplemental Materials

This section describes different supplementary materials for teaching by the flipped classroom model that are available in the internet for free. This are video lectures, quizzes and programming problems for algorithm learning from the best high education institutions. These materials are especially helpful for "homework" part of the flipped classroom model. For example, it will save a lot of time of teachers to use existing high quality video lectures instead of recording all by themselves.

Algorithms: Design and Analysis from Stanford University on Coursera

Course of algorithms offered by Stanford University on Coursera website covers wide range of algorithms and data structures that are usually taught in introductory courses to algorithms to undergraduates. Non exhausting list of algorithms and data structures covered is: different sorting algorithms (QuickSort, MergeSort, …), divide and conquer algorithms, greedy algorithms, dynamic programming, graphs and graph processing algorithms (breath-first search, depth-first search, shortest path, minimum spanning tree), data structures like heap, binary search trees, balanced BST, hash tables), exact and heuristic algorithms for NP-Complete problems. Each topic is explained through video lectures. In addition, subtitles are available for non-native English speakers to understand the content easier. Distinguishing element in this course is that most of the time instructor tries to explain main idea of algorithm, leaving programming language specifics and sometimes even some special cases in algorithms for students to figure out by themselves. This makes it easier to understand main

idea of algorithm and at the same time encourages students to work independently to complete algorithm. Course also provides 12 sets of theoretical problems and 12 sets of programming questions (requiring algorithm implementation by student) in order to evaluate how student learned the material. Besides, there is also included 2 final exam-s that both consist of 20 questions about covered material. All this resources listed above are great as homework part of the flipped classroom because they give opportunity to present new material to students and then to check their understanding with automated quizzes that does not require teacher's involvement for evaluation [15].

Algorithms from Princeton University on Coursera

Algorithms course from Princeton University on Coursera website contains also many useful supplementary materials for the flipped classes. It covers most of the topics that are covered in Stanford's course, however it is not doing emphasis on concepts like divide and conquer, greedy algorithms and dynamic programming. The biggest distinguishing trait of this course is that it provides not pseudocodes of algorithms, but instead it provides complete algorithms that are implemented in Java programming language. This approach is good for students that lack abstract thinking ability and understand algorithms better when they see working code of it in details, as opposed to teacher explaining just main idea of algorithm. This course also provides 24 sets of exercises and questions and 2 final exams that consists of exercises and questions too. Exercises most of the time require students to trace code of algorithms and write values of some variables after "running" certain lines of code as answer. This process of "running" code by one's own brain instead of by computer greatly helps students to fully understand how algorithm works. Another distinguishing element of this course is auto grader that is used to evaluate 10 programming assignments that the course offers. As opposed to many other courses where programming assignments are checked only with output on certain input, auto grader can check program for correctness and for time and memory efficiency. In addition, it checks this not only for final input and output, but also for individual methods and for some combinations of methods. All this makes auto grader powerful tool for automated evaluation of the programming assignments [1].

Introduction to Algorithms from Massachusetts Institute of Technology

Introduction to algorithms course from MIT on MIT's open courseware website is one of the most comprehensive supplementary resource for teaching introductory course of algorithms. At the same time it is one of the most comfortable one to use, because all resources in this

course are under creative commons license, can be downloaded with one clock as package and used by the teacher, as opposed to courses mentioned previously in this section that require students to register, get materials and submit assignments on Coursera website. Algorithm course from MIT containing all materials that was used to teach introduction to algorithms in MIT in fall semester of 2011. This include video lectures and recitation videos with notes in pdf, 7 problem sets with theoretical and programming questions, solutions to problem sets, 2 quizzes and one final exam with solutions. Video lectures explain main concepts of algorithms and data structures, sometimes even without writing pseudocode. Recitation videos explain some hard to understand topics in more detail, answering student's questions and are going through exercises. For further details, references to different chapters of algorithm books are provided. Concerning programming questions, Solutions for them must be written in Python, tests are provided students to be able to test correctness of solutions locally [16].

Algorithms from Khan Academy

Last supplementary resource for the flipped classes of algorithms that will be discussed here is algorithms course on KhanAcademy website. It is not as comprehensive as previous three resources mentioned before, but it has some elements that distinguishes it from others. It includes readings that explain material, quizzes and programming assignments as "challenge"-s and "project"-s. Challenges are type of assignment that can't be found in resources discussed before. This type of assignment requires student to implement algorithm that he just learned online in java script. Besides, page where student is typing exercise is very dynamic and gives lots of feedback, thus guiding student towards successful implementation and gives tips if student got stuck somewhere. Elements from this assignment can be seen in Figure 5 and Figure 6.

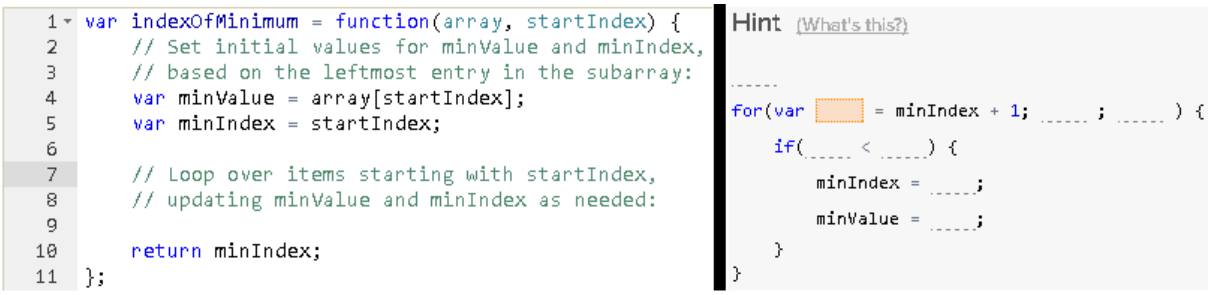

Figure 5 – Area to fill missing code fragment(left) and hint for it(right).

This type of assignment is very useful because sometimes after watching video lecture student may think he fully understood algorithm, when actually this is not the case. This assignment will check if student has any gaps in comprehending algorithm and if so will try to guide student by giving feedback and tips.

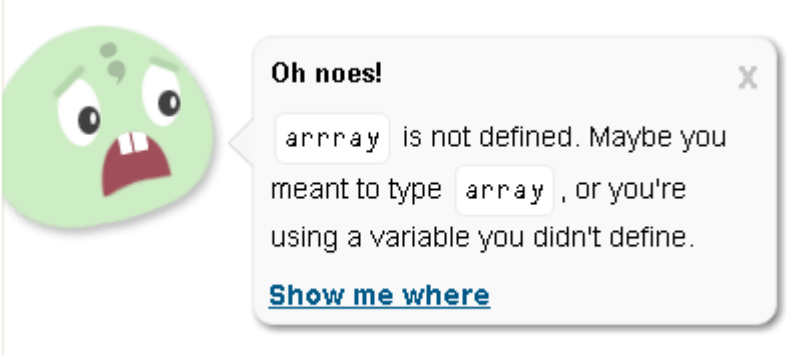

Figure 6 – Page for programming assignment giving tip to student.

At the moment there is only two "project" type of programming assignments. One of them, for example, asks student to visualize execution of selection sort. Since this web site is improving day by day, it is possible that they will add more exercises and materials to the course [17].

Chapter 3

3 Design of the pedagogical experience

Many research that have been done on the flipped classroom indicate positive results but at the same time state that further research is required. So we decided to put another brick on research around the flipped classroom by designing an experiment that would compare the flipped model of teaching to the traditional model. Furthermore, our hypothesis was that the flipped classroom would be an efficient way for teaching algorithms, it would make it easier for students to comprehend the subject and would increase their results. Thus, the designed experiment is specifically about teaching algorithms through the flipped classes. Its main objective is to check validity of the hypothesis mentioned above. Besides, it will collect students' opinions about the learning experience: if they agree that the flipped classroom facilitates their learning and if they prefer it over the traditional model. Subjects for the experiment should be students who have some background in the Java programming language and should know programming concepts such as recursion. Time period for running the experiment is around two weeks.

3.1 Description

In order to be able to show contrast in efficiency between the flipped and the traditional teaching methods the experiments require as subjects two groups of students. During the experiment, for eight days, one of this group of students will learn through the flipped classes, while second group will learn in a traditional way. Sequence of the steps in the experiment is as follows: first, both groups will write pre-test, that will be followed by different learning experiences for each group, where they will learn sorting algorithm called Mergesort. After 8

days since pre-test students will write a final test that will evaluate how they mastered new material. This process is visualized in Table 1.

Table 1 – Steps of the experiment.

Days	1	1	7	8
Flipped Class	Pre-test	Assign Flipped Homework	Flipped Class	Final Test
Traditional Class	Pre-test	Lecture, Practice, Assign Traditional Homework		Final Test

In The following days opinions of students that were part of the flipped classroom will be collected from an online form and in addition, their teacher will deliver field notes. Each of this stages will be described in detail in the following paragraphs.

Pre-test

As mentioned above, pre-test is the first step that both group goes through, on the first day of the experiment. Its function is to assess basic programming skills and problem solving abilities of the students. This information can be used to see if students from one group had better background and were better prepared than others. If this will be the case some adjustments on results of the final test will be needed in order to make correct conclusions from the results.

The test consists of three Java programming questions, each of them is worth one point. They ask students to fill in the body of a function. Questions are ordered by their complexity, with first one being very easy. Besides, test's main goal of assessing student's abilities, questions from the test are designed to bring students into thinking about sorting. For example, first questions makes student think about what it means formally that an array is sorted. Also, third question is about merge part of the Mergesort algorithm. Answering that programming question or even just thinking about possible solutions will help students to understand Mergesort easier when it will be presented on a lecture. All three questions of the test can be found in an appendix A.

Learning experience for the flipped class

After writing the pre-test, on the same day, the teacher of the flipped classroom will assign flipped homework to students from his group. Students will have six days for doing this

homework, that is, until they will have the flipped class. The homework consists of written instructions, video lecture, fragment from a book of algorithms, exercises and their solutions.

The written instructions are delivered as the word document, they are quite short, but they guide student through the homework, suggesting what to do first, second and so on. Also, instructions contain some advices for students, for example, about how can they use video lecture to get most from it. For more details see appendix b. Video lecture assigned as part of the homework is from an online course called Algorithms from the Coursera website (URL is provided to students), it explains the Mergesort algorithm [1]. For non-native English speakers subtitles are available in order to facilitate learning. Besides, for those students who prefer learning by reading, fragment of book of algorithms that explains same material is provided. The Book is called Algorithms (4^{th} edition) and author of it is the lecturer of the video lecture [2]. In case if this specific resources will not be possible to get, experiment can be done with any other, similar kind of resources that explain the Mergesort. The last element of the homework are exercises, they are delivered as the word document. Document contains two types of exercises, first one asks students to write code of the Mergesort and second one requires students to trace code of the Mergesort. Second exercise contains three variants of an input array. This exercises can be seen in detail in appendix c. The second exercise, like video lecture, is from course of algorithms from the Coursera website [1]. Besides, students also get the word document with solutions to the exercises in order to check if they were correct.

After doing the homework, on the seventh day of running the experiment, students from the flipped classroom have a class. Its duration is about three hours. The class starts by teacher asking questions to students in order to understand how well they comprehended new material, in this case the Mergesort. Further, students can ask questions or ask for explanation of some complex parts of a content to the teacher. Besides, if teacher will see the need he may run through all explanation of the Mergesort. After, students are given exercises in order to practice further. This also gives to the teacher an opportunity to observe and see how well students understood the content. This exercises are very similar to exercises that were included in the homework, namely one about tracing Mergesort. While students are working on the exercises, the teacher is walking from student to student, observing, guiding and helping them when needed. Students are also encouraged to do peer instruction, to help each other in process of understanding material. After this class students should be ready to write the final test.

Learning experience for the traditional classroom

After the pre-test, on the same day, students that belong to the traditional classroom will have a class. As the flipped class it will also last for three hours. During first hour teacher will present the Mergesort to students. After, students will have possibility to ask questions to the teacher around the topic and they will have a practical work where students will do same exercises that their peers from the flipped classroom. Meanwhile, the teacher will guide and help them. At the end of the class, teacher will give homework to students. As homework they will have to read and understand fragment from the book of algorithms about the Mergesort (same book that was mentioned in previous section) and also they will have to do same exercises that was part of the flipped homework. Students will have seven days to do the homework and to prepare for the final test.

Final test

Probably the most important element for evaluation is the final test. Just like pre-test, it is written by both groups. It consists of three questions about the Mergesort and its purpose is to give information about which group of students mastered the material better. First question asks students to complete code of Mergesort which is missing four lines. Second question is very similar to second exercise from the homework, it requires students to trace through Mergesort. Third question presents an initial array, same array in sorted order and four permutations of that array and asks students to indicate which two from these permutations can be an intermediate state while running Mergesort over the initial array. This question checks if students understand how Mergesort really works and that they have not just memorized how to solve some exercises. Just like in the pre-test, here also each question is worth one point, so maximal amount of points that can be earned in this test is three. Full version of the final test can be seen in the appendix d. Second and third questions are from course of algorithms from the Coursera website [1].

Questionnaire for students and field notes by the teacher

After taking the final test, students will be asked to fill questionnaire, result of which will be used to see what is their opinion about the flipped classroom and about related teaching elements. It is implemented in Google Forms and contains following 8 statements: I participated in the Flipped Classes pedagogical strategy, I believe I like flipped classes pedagogical approach more than traditional classes, I usually study the subject before being explained in class, I usually do exercises at home before being explained in class, I tried to do

flipped class homework before class, Studying before class helped me comprehend the material explained in class, Watching videos, reading books and doing exercises by myself is more productive than participating in class and Exercises given before class help doing practical work in class. Students can answer questions by 1 to 5 scale. 1, 2, 3, 4, 5 meaning totally disagree, disagree, don't agree nor disagree, agree, totally agree, respectively. One reason why students' opinions are important is that if they don't like the way of teaching, even if it will be inherently effective method, they will just not be motivated to study through it, thus, will have bad results.

Last method of evaluation used in the experiment is field notes by a teacher of the flipped classroom. In them the teacher should describe how the class went, how students were prepared after doing a flipped homework, if they understood class material easier as a result of watching video lectures or doing other elements of the homework. How active and engaged they were during the class. If time spared for presenting Mergesort was used fruitfully. Field notes are also important part of the evaluation, it may help researchers to see results from a different angle.

Chapter 4

4 Conducting Experiment

The experiment described in the previous chapter was applied in the Polytechnic Institute of Bragança. Subjects were students who were taking a course of Java Programming. In order to do experiment the group was split in two so that nine students became part of the flipped class and ten students became part of the traditional class. The experiment started at 22 April of 2016. Its duration was eight days, as planned in the previous chapter. In the next sections results of different methods of evaluations will be discussed.

4.1 Evaluation

Pre-test

Results from the pre-test can be seen in the Table 2. The table indicates that there was insignificant difference between average scores of students from the flipped class and students from the traditional class, 0.5 and 0.48, respectively, out of the maximum score which was three.

Table 2 – Results of the pre-test.

	Flipped Class	Traditional Class
	0	0.25
	1.35	0
	1	0
	0	0.7

	0.95	1
	0.25	0
	0.95	0,95
	0	0,95
Average	0.5	0.48
Standard Deviation	0.55	0.46

Standard deviation was little higher in case of flipped class (0.55) compared to traditional class (0.46). This results suggest that students background in the basic programming and problem solving skills were equal between two groups. However, students' skills varied little more in the flipped classroom, as opposed to the traditional classroom where students skills were more similar to each other.

Final test

Results from the final test can be seen in the Table 3. Students from the flipped classroom had slightly better results on average than students from the traditional classroom, 1.92 and 1.73, respectively. Difference is 6.5% of the maximal score which was three.

Table 3 –Results of the final-test.

	Flipped Class	Traditional Class
	2	0.8
	2.25	1.8
	0	2.5
	0.2	2
	3	2.5
	2.8	1.75
	3	1.2
		1
		2.5
		1.2
Average	1.92	1.73

Standard Deviation	1.29	0.65

Standard deviation was higher in flipped class (1.29), as opposed to traditional class (0.65). Besides fact that standard deviation of students from the flipped class was also higher on the pre-test, another reason for this difference on the final test can be nature of the flipped classes. Namely, it requires students to be more independent, to learn new material at home or at least get some idea about it, because teacher will only explain material briefly or not at all and will directly go to questions and exercises. Thus, those students who were generally not so good in the subject failed even more dramatically than they would in the traditional classroom. But on the other hand, students who were able to work independently received high scores, so that average score of the flipped class was still higher even though few students from it received zero or close to zero score.

Questionnaire

Results from the questionnaire can be seen in the Table 4. It is categorized them according to groups. This categorization is derived from the first question that stated "I participated in flipped classes". Other seven questions are listed in the table along with average scores. Unfortunately, only four students from the traditional class participated in questionnaire, however, everyone from the flipped class filled it.

Table 4 – Statistics from the questionnaire.

	Flipped Class	Traditional Class
I believe I like flipped classes pedagogical approach more than traditional classes	3.5	2.5
I usually study the subject before being explained in class	2.6	2
I usually do exercises at home before being explained in class	2.3	2
I tried to do flipped class homework before class	3.3	2.75
Studying before class helped me comprehend the material explained in class	3.6	3.5
Watching videos, reading books and doing exercises by myself is more productive than participating in class	2.4	2.5
Exercises given before class help doing practical work in class	3.6	3.5

Results worth mentioning are the following firstly, most of the students that belonged to the flipped class agree that they prefer flipped classes compared to traditional classes (3.5 average score). Also students from both groups agree that studying and doing exercises before class helps them understand material presented in a class, 3.6 was average score on this questions from students from the flipped class and 3.5 was average score from students from the traditional class. However, one negative but expected results was that more students disagree that they learn material more productively from reading and watching video lectures than from attending a class, 2.4 and 2.5 are average scores from the flipped and traditional classrooms, respectively.

Field notes

In this paragraph will be presented field notes delivered by the teacher of the flipped classroom. The teacher: "The class started by asking question if students have done a flipped homework. Unfortunately, very few of them had done it fully, but more than half of the students at least run through it briefly and had idea about what it was. This fact made possible for me to present the Mergesort algorithm faster than I would in an traditional classroom and also it made easier for students to comprehend the material. The flipped format gave me possibility to present the Mergesort in an more interactive way, students were asking questions and at the same time I was watching them to understand if everyone was catching up, sometimes asking about next step, to check if they remembered it from the homework and sometimes this was the case. Because presentation went faster, about 30-40 minutes, I was able to go to practical exercises sooner, these were exercises like ones from the homework, that required tracing the Mergesort. First, students were doing them individually, I was walking from student to student to check their progress, if someone was on the wrong path I was explaining related theory that was needed to solve that problem or when possible I was asking instructive questions for them to indicate mistake by themselves and go towards right direction for solving the exercise. Few times when I saw that some students had similar problem I explained theory related to it on the blackboard. Because it was impossible for me to give individual instructions to several students at the same time, I tried to encourage peer instruction and it was successful, students helped to each other. In conclusion, I think the flipped class went successfully, every student except one or two were engaged and understood the material quite well. Those who were not engaged were ones who haven't done homework and nor did they had good enough background knowledge.".

Chapter 5

5 Conclusions

Based on analyzing different research works on the flipped classroom and based on results of the final test, questionnaire and field notes from the experiment described in this document, it is believed that the flipped classroom model is an efficient way of teaching, outperforming the traditional model in some situations. Our experiment showed that it is suitable for teaching algorithms. Students from the group that were learning through the flipped classroom method had on average higher scores than students who were learning through the traditional method. Also, most of the students had positive opinions about the flipped classroom and its elements. Besides higher results and students' favorable opinions, another positive side of the flipped classroom is that it encourages student to become independent learners. This is very important nowadays, when the field of computer science is developing and changing very fast. This means, for example, students after four years since graduation may have to work on things they have never learned, thus one important and useful skill that he should take from high educational institution is ability to learn efficiently independently.

Besides this positive results, it is believed that flipped classes have even more potential. This potential is in fact that in time students can learn how to learn through them. Currently, most of the students have been learning how to learn by traditional methods almost all their life, they even have developed some habits for it. In order for flipped classes to reach their full potential students should develop similar kind of knowledge (how to learn), skills and habits specifically for it and this will increase efficiency of this teaching model. For this it is required the flipped classes to become more widespread, in such case students will have opportunity and sufficient time to learn how to learn specifically by this method.

The same can be said about research around the flipped classes, for more trustworthy results is needed more massive experiments, especially for longer time periods. For example, in our experiment that lasted for two weeks, students had very few time to understand how flipped classes work and to get used to them, thus, it is believed that if students had more time for this, difference in results of the flipped and traditional classrooms would been even bigger.

In summary, it is believed that the flipped classroom teaching model is an efficient method for teaching algorithms, slightly more efficient than traditional model. Besides, if it will become more widespread, its efficiency can increase because currently flipped classes are something new for students and they don't have enough skills or habits to use its full potential.

Bibliography

[1] Robert Sedgewick, Kevin Wayne. Algorithms course on coursera.org by Princeton University. https://www.coursera.org/learn/introduction-to-algorithms

[2] Robert Sedgewick, Kevin Wayne. Algorithms (4th Edition). 2011.

[3] Anabela Gomes, Antonio Jose Mendes. Learning to program - difficulties and solutions. International Conference on Engineering Education - ICEE 2007.

[4] Anabela Gomes, Lilian Carmo, Emilia Bigotte, Antonio Mendes. Mathematics and programming problem solving. 3rd E-Learning Conference. 2006.

[5] Jacob Lowell Bishop, Dr. Matthew A Verleger. The Flipped Classroom: A Survey of the Research. 120th ASEE annual Conference and Exposition. 2013.

[6] Jacqueline E. McLaughlin, Mary T. Roth, Dylan M. Glatt, Nastaran Gharkholonarehe, Christopher A. Davidson, LaToya M. Griffin, Denise A. Esserman, Russell J. Mumper. The Flipped Classroom: A Course Redesign to Foster Learning and Engagement in a Health Professions School. Academic Medicine, Vol. 89, No. 2. 2007.

[7] Celine Latulipe, N. Bruce Long, Carlos E. Seminario. Structuring Flipped Classes with Lightweight Teams and Gamification. 2015.

[8] Kathleen Fulton. Upside Down and Inside Out: Flip Your Classroom to Improve Student Learning. 2012.

[9] P.A. Cohen, B.J. Ebeling, and J.A. Kulik. A meta-analysis of outcome studies of visual-based instruction. Educational Technology Research and Development, 29(1):26–36, 1981.

[10] S.W. Bonham, D.L. Deardorff, and R.J. Beichner. Comparison of student performance using web and paperbased homework in college-level physics. Journal of Research in Science Teaching, 40(10):1050–1071, 2003.

[11] R.M. Felder and R. Brent. Designing and teaching courses to satisfy the ABET engineering criteria. Journal of Engineering Education, 92(1):7–25, 2003. ISSN 1069-4730.

[12] M.J. Lage, G.J. Platt, and M. Treglia. Inverting the classroom: A gateway to creating an inclusive learning environment. The Journal of Economic Education, 31(1):30–43, 2000.

[13] R. Toto and H. Nguyen. Flipping the work design in an industrial engineering course. In Frontiers in Education Conference, 2009. FIE 2009. 39th IEEE, pages 1–4. IEEE, 2009.

[14] Janet L. DeGrazia, John L. Falconer, Garret Nicodemus, and Will Medlin. Incorporating screencasts into chemical engineering courses. In Proceedings of the ASEE Annual Conference & Exposition, 2012.

[15] Tim Roughgarden. Algorithms: Design and Analysis course on coursera.org by Stanford University. https://www.coursera.org/learn/algorithm-design-analysis

[16] Erik Demaine, Srinivas Devadas. 6.006 Introduction to Algorithms. 2011. http://ocw.mit.edu

[17] Algorithms course from khanacademy.org.

https://www.khanacademy.org/computing/computer-science/algorithms

[18] P. Denny, A. Luxton-Reilly, and B. Simon. Evaluating a new exam question: Parsons problems. In Proceedings of the Fourth International Workshop on Computing Education Research. ACM, 2008.

Appendix A

1. Write function that will take as parameter **a** array of int-s, will check if it is sorted and will return true if it is and false if it isn't.

```
boolean isSorted(int[] a) {

}
```

2. Write function that will take as parameter **a** array of int-s, will find smallest element in it and after will swap its place with first element of an array.

```
void findMinAndSwap(int[] a) {

}
```

3. Write function that will take as parameters **a1** and **a2** sorted arrays of int-s and will return sorted array of int-s that will contain all data from both arrays.

```
int[] combineSortedArrays(int[] a1, int[] a2) {

}
```

Appendix B

1. Watch video lecture (with subtitles if needed), if explanation will be too fast you can pause and think about it, rewind, or watch whole lecture several times.
   If needed you can read about same content from textbook.
2. Do exercises.

Appendix C

1. Write the code of Mergesort.

2. Give the array that results immediately after the 7th call (and return) from merge() when top-down mergesorting the following array of size 12:

a) 42 41 51 76 99 28 73 83 36 89 77 67
b) 55 34 61 53 30 72 45 66 27 22 51 24
c) 91 95 29 38 61 12 89 23 78 88 58 96

Example: 74 26 42 47 11 29 76 80 50 91 57 12

Solution:
merge(0, 0, 1): 26 74 42 47 11 29 76 80 50 91 57 12
merge(0, 1, 2): 26 42 74 47 11 29 76 80 50 91 57 12
merge(3, 3, 4): 26 42 74 11 47 29 76 80 50 91 57 12
merge(3, 4, 5): 26 42 74 11 29 47 76 80 50 91 57 12
merge(0, 2, 5): 11 26 29 42 47 74 76 80 50 91 57 12
merge(6, 6, 7): 11 26 29 42 47 74 76 80 50 91 57 12
merge(6, 7, 8): 11 26 29 42 47 74 50 76 80 91 57 12

Appendix D

1. Complete Mergesort code.

```
public class Merge {
    private static void merge(Comparable[] a, Comparable[] aux, int
lo, int mid, int hi) {
        for (int k = lo; k <= hi; k++) {
            aux[k] = a[k];
        }
        int i = lo, j = mid+1;
        for (int k = lo; k <= hi; k++) {




        }
    }

    private static void sort(Comparable[] a, Comparable[] aux, int
lo, int hi) {
        if (hi <= lo) return;
        int mid = lo + (hi - lo) / 2;
        sort(a, aux, lo, mid);
        sort(a, aux, mid + 1, hi);
        merge(a, aux, lo, mid, hi);
    }

    public static void sort(Comparable[] a) {
        Comparable[] aux = new Comparable[a.length];
        sort(a, aux, 0, a.length-1);
    }

    private static boolean less(Comparable v, Comparable w) {
        return (v.compareTo(w) < 0);
    }
}
```

2. Give the array that results immediately after the 7th call (and return) from merge() when mergesorting the following array:

74 77 61 93 30 86 60 56 72 65 88 82

3. The column on the left contains an input array of 12 strings to be sorted; the column on the right contains the strings in sorted order; 2 of the other 4 columns contains the array at some intermediate step during mergesort. Identify those columns.

1.	2.	3.	4.	5.	6.
lust	bark	bark	bark	bark	bark
sand	gold	gold	ecru	lust	buff
bark	leaf	leaf	gold	pear	coal
pear	lust	lust	leaf	sand	ecru
gold	pear	pear	lust	ecru	flax
leaf	sand	sand	navy	gold	gold
ecru	ecru	ecru	pear	leaf	leaf
navy	flax	navy	sand	navy	lust
flax	navy	flax	buff	buff	navy
rose	buff	rose	coal	coal	pear
buff	coal	buff	flax	flax	rose
coal	rose	coal	rose	rose	sand