Syllabus

1. Course Description
    a. Title of a Course

    Mathematical Thinking in Computer Science

    b. Pre-requisites

    The Course is to be based on the acquisition of the following Courses:
    - Mathematical analysis;
    - General English.

    c. Course Type (compulsory, elective, optional)

    elective

    d. Abstract

    Mathematical thinking is crucial in all areas of computer science: algorithms, bioinformatics, computer graphics, data science, machine learning, etc. In this course, we will learn the most important tools used in discrete mathematics: induction, recursion, logic, invariants, examples, optimality. We will use these tools to answer typical programming questions like: How can we be certain a solution exists? Am I sure my program computes the optimal answer? Do each of these objects meet the given requirements? In the course, we use a try-this-before-we-explain-everything approach: you will be solving many interactive (and mobile friendly) puzzles that were carefully designed to allow you to invent many of the important ideas and concepts yourself.

2. Learning Objectives

    The objectives of this course are to learn the most important mathematical tools used in discrete mathematics: induction, recursion, logic, invariants, examples, optimality.

3. Learning Outcomes

    The student will know:

    - the basic the most important mathematical tools used in discrete mathematics: induction, recursion, logic, invariants, examples, optimality

    The student will be capable of:

    - Estimation the computational complexity of the problems and the amount of computational resources for their solution;
    - Adaptation new problematics, knowledge, scientific terminology and methodology, to possess the skills of independent learning;

    The student will get experience in:

    - Theoretical analysis of real problems related to mathematical concepts in computer science.

4. Course Plan

Section 1. *Making Convincing Arguments.*

Why some arguments are convincing and some are not? What makes an argument convincing? How to establish your argument in such a way that there is no possible room for doubt left? How mathematical thinking can help with this? In this week we will start digging into these

questions. We will see how a small remark or a simple observation can turn a seemingly non-trivial question into an obvious one. Through various examples we will observe a parallel between constructing a rigorous argument and mathematical reasoning.

***Section** 2. How to Find an Example?*

How can we be certain that an object with certain requirements exist? One way to show this, is to go through all objects and check whether at least one of them meets the requirements. However, in many cases, the search space is enormous. A computer may help, but some reasoning that narrows the search space is important both for computer search and for "bare hands" work. In this module, we will learn various techniques for showing that an object exists and that an object is optimal among all other objects. As usual, we'll practice solving many interactive puzzles. We'll show also some computer programs that help us to construct an example.

***Section** 3. Recursion and Induction.*

We'll discover two powerful methods of defining objects, proving concepts, and implementing programs — recursion and induction. These two methods are heavily used, in particular, in algorithms — for analysing correctness and running time of algorithms as well as for implementing efficient solutions. You will see that induction is as simple as falling dominos, but allows to make convincing arguments for arbitrarily large and complex problems by decomposing them and moving step by step. You will learn how famous Gauss unexpectedly solved his teacher's problem intended to keep him busy the whole lesson in just two minutes, and in the end you will be able to prove his formula using induction. You will be able to generalize scary arithmetic exercises and then solve them easily using induction.

***Section** 4. Logic*

We have already invoked mathematical logic when we discussed how to make convincing arguments by giving examples. This week we will turn mathematical logic full on. We will discuss its basic operations and rules. We will see how logic can play a crucial and indispensable role in creating convincing arguments. We will discuss how to construct a negation to the statement, and you will see how to win an argument by showing your opponent is wrong with just one example called counterexample!. We will see tricky and seemingly counterintuitive, but yet (an unintentional pun) logical aspects of mathematical logic. We will see one of the oldest approaches to making convincing arguments: Reductio ad Absurdum.

***Section** 5. Invariants*

"There are things that never change". Apart from being just a philosophical statement, this phrase turns out to be an important idea that can actually help. In this module we will see how it can help in problem solving. Things that do not change are called invariants in mathematics. They form an important tool of proving with numerous applications, including estimating running time of programs and algorithms. We will get some intuition of what they are, see how they can look like, and get some practice in using them.

5. Reading List

All materials in https://www.coursera.org/learn/what-is-a-proof

6. Grading System

The student's final assessment consists of the assessment of the exam A_{exam} and the accumulated assessment A_{acc} obtained on the platform https://www.coursera.org/learn/what-is-a-proof as follows:

$$A_{final} = (A_{acc} + A_{exam})/2$$

7. Guidelines for Knowledge Assessment

   Presented in https://www.coursera.org/learn/what-is-a-proof

8. Methods of Instruction

   The course is being studied on the online platform https://www.coursera.org/learn/what-is-a-proof

9. Special Equipment and Software Support (if required)

   Not required