


# COMP 6901 -- Applied Algorithms Fall 2020

[Announcements](#) | [Course information](#) | [Assignments and tests](#) | [Lecture notes](#)



## Course information

The course information below is **very tentative!** The way the lectures/tests are done and the marking scheme, in particular, are likely to change depending on everybody's availability, enrolment, software licenses and other factors.

Instructor: [Antonina Kolokolova](#)

Instructor office hours: TBA. Also on Zoom.

Additionally, we will be using discussion boards on [Brightspace](#).

We will be using [Brightspace \(formerly: known as D2L\)](#) for assignment submissions, tests, grades, announcements, etc. The assignments, slides and videos of lectures will also be posted here. The Brightspace shell for our course should be available shortly to all registered students; if you cannot register/access Brightspace after the first week of classes, please let me know.

Textbook: There will be no official textbook for this course. We will mostly follow *J. Kleinberg, E. Tardos. Algorithm Design*, but you do not need to buy it.

Reference books:

- J. Kleinberg, E. Tardos. Algorithm Design.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms

We will also use other materials such as research papers.

See the [lecture notes from the previous run of this course](#) for more information.

**Marking scheme:** (tentative!): 4 assignments 15% each, and two tests 20% each. The last test is likely to be scheduled during the last week of classes.

**Description:** The goal of this course is to study both classical and advanced algorithm design techniques with emphasis on real-world applications. We will cover greedy algorithms, dynamic programming, backtracking, network flows as well as more advanced algorithms and techniques (time permitting). Time permitting, we will consider randomized, parallel and distributed algorithms, and/or streaming algorithms.

**Prerequisites:** This course assumes proficiency in the core subjects of computer science: programming, discrete math, and basic data structures/algorithms. In particular, I will assume that you can read and write proofs, know basic probability theory and combinatorics, know basic data structures and algorithm complexity analysis, can read and write pseudocode, and can program fluently in some programming language. Check our undergraduate courses COMP 1001, COMP 1002 and COMP 2002, and make sure you are comfortable with the material covered there; this should take care of most of the skills you need to succeed in COMP 6901.

**Policy on collaboration:** The work you submit must be your own. You may discuss problems from assignments with each other; however, you should prepare written solutions alone, and do not interact with anybody except the instructor while doing tests and online exercises. Plagiarism is a serious academic offense and will be dealt with accordingly. When in doubt, check [How to avoid Plagiarism](#)