

# CSCI-570: Analysis of Algorithms

Spring 2020

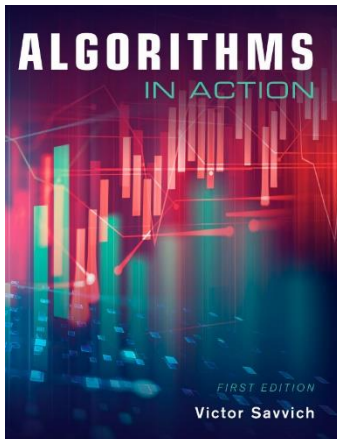
## Course Description

This course is about designing algorithms for computational problems, and how to think clearly about analyzing correctness and running time. The main goal of this course is to provide the intellectual tools needed for designing and analyzing your own algorithms for new problems you need to solve in the future. The course explores fundamental algorithm design techniques such as greedy, divide and conquer, dynamic programming, network flow, reduction, approximation, linear programming and randomization for efficient algorithm construction. The course describes Turing machines and explains what *NP*-completeness means with respect to possibilities for solving these problems efficiently.

## Learning Objectives:

- Understanding a variety of techniques for designing algorithms.
- Develop skills to reason about and prove properties of algorithms such as their correctness and running time.
- Design experiments to evaluate and compare different algorithm techniques on real-world problems
- Use approximation and linear programming to find near-optimal solutions for challenging problems.
- Use the concept of randomization to find efficient algorithms for challenging problems.
- Use the theory of *NP*-completeness to argue for the difficulty of some problems.

## Required Textbook:



*Algorithms in Action*, by V. Savvich, First Edition, 2019.

Purchase the textbook either from the university bookstore or from

the publisher at: <https://store.cognella.com/82372-1b-002>

## Optional books:

*Introduction to Algorithms*, by T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein

*Algorithms*, by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani.

*Algorithm Design*, by J. Kleinberg and E. Tardos

## Prerequisites:

Students in the class are expected to have a reasonable degree of mathematical sophistication, and to be familiar with the basic notions of algorithms and data structures, discrete mathematics, and probability. Undergraduate classes in these subjects should be sufficient. If you have no previous background in these I suggest a more thorough introduction such as “Mathematics for Computer Science”, by Eric Lehman, Thomson Leighton and Albert Meyer, Samurai Media Limited, 2017. The course does not emphasize nor require programming, just pseudocode to encourage students on conceptual understanding.

## Theory Homeworks:

- There will be five written theory assignments.
- The assignments should be submitted electronically via [Desire2learn](#).
- Theory assignments must be neatly written or typed.
- You may work in groups of 2-3. However, each person should hand-in their own writeup.
- Collaboration should be limited to high level talking about the problems, so that your writeup is written entirely by you and not copied from your partner.
- There are NO late days for assignments.

## Homework's Purpose:

Algorithms is a pivotal course in computer science studies. The course will require a significant amount of work on your part to follow what is taught in class and complete homework successfully. We stress that the homework is an essential part of your course work. We devote a fairly large amount of time for designing, writing, grading and explaining the homework, so that you can test yourselves and see how well you understand and implement the course's material.

## Exams:

- There will be two midterm exams.
- No makeup exams will be provided.
- If you missed the exam, you may be eligible for an IN grade for the course. The incomplete grade has to be completed within one year. However, in order to get an IN you have to have a valid cause. Please read the University policy on IN grade for more details.
- The exam solutions and grading rubric will always be posted.
- There will be a regrading session for each exam where you can discuss grading errors. A regrade is allowed only when there are clear and obvious grading errors. Grading errors are simple mistakes made on the part of the graders, and not differences in interpretation of a question or answer.

## Grading:

Final grades for the course will be determined by a curve. We will compute the letter grade cutoffs by setting the mean score to be equal a B.

Artifact	Weight	Date
Homework	<b>20%</b>	
First midterm exam	<b>40%</b>	Fri, Mar 13
Second midterm exam	<b>40%</b>	Fri, May 1

## Academic Integrity:

The USC Student Conduct Code prohibits plagiarism. All USC students are responsible for reading and following the Student Conduct Code, which appears on <https://policy.usc.edu/files/2018/07/SCampus-2018-19.pdf>.

In this course we encourage students to study together. This includes discussing general strategies to be used on individual assignments. However, all work submitted for the class is to be done individually. Some examples of what is not allowed by the conduct code: copying all or part of someone else's work (by hand or by looking at others' files, either secretly or if shown), and submitting it as your own; giving another student in the class a copy of your assignment solution; consulting with another student during an exam. If you have questions about what is allowed, please discuss it with the instructor.

## Schedule:

This schedule is meant as an outline. Depending on progress, material may be added or removed. Each lecture (and exam) is 2hrs and 20 mins long.

Date	Topics Covered
Jan. 13 – 17	<a href="#">Lecture 1</a> : Timing Analysis, Trees and Graphs, Proofs
Jan. 20 – 24	<a href="#">Lecture 2</a> : Amortized Data Structures and Their Analysis <b>HW1</b> (due Feb. 2)
Jan. 27 – 31	<a href="#">Lecture 3</a> : Binary and Binomial Heaps
Feb. 3 – 7	<a href="#">Lecture 4</a> : Greedy Algorithms <b>HW2</b> (due Feb. 16)
Feb. 10 – 14	<a href="#">Lecture 5</a> : More on Greedy, Master Theorem
Feb. 17 – 21	<a href="#">Lecture 6</a> : Divide-and-Conquer Algorithms <b>HW3</b> (due Mar. 8)
Feb. 24 – 28	<a href="#">Lecture 7</a> : Dynamic Programming
Mar. 2 – 6	<a href="#">Lecture 8</a> : Dynamic Programming
Mar. 9 – 13	Review for exam. <b>Exam – I</b> (on Friday at 5pm)
Mar. 16 – 20	<b>Spring Recess – no classes</b>
Mar. 23 – 27	<a href="#">Lecture 9</a> : Network Flow <b>HW4</b> (due Apr. 5)
Mar. 30 – Apr. 3	<a href="#">Lecture 10</a> : Flow Circulation
Apr. 6 – 10	<a href="#">Lecture 11</a> : Linear Programming <b>HW5</b> (due Apr. 19)
Apr. 13 – 17	<a href="#">Lecture 12</a> : NP-Completeness
Apr. 20 – 24	<a href="#">Lecture 13</a> : More on NP, Approximation Algorithms
Apr. 27 – May 1	Review for exam. <b>Exam – II</b> (on Friday at 5pm)

## Office Hours:

Mon	Tue	Wed	Thu	Fri
Liu Kuang: 9-11am, SSC 300	Wenxuan Zhou: 10am-12pm, RTH 314	Ta-Yang Wang: 9-11am, SAL Lab	Wenxuan Zhou: 10am-12pm, RTH 314	Mengxiao Zhang: 9-11 am, PHE 330
Ali Omrani: 10am-12pm, SGM 605	Di Huang: 1-3pm, SAL Lab	Ali Omrani: 10am-12pm, SGM 605	Mengxiao Zhang: 9-11 am, PHE 330	Xingyu Wei: 11am-1pm, SAL Lab
Di Huang: 3-5pm, SAL Lab	Daniel Link: 1-3pm, SAL 332	Xingyu Wei: 11am-1pm, SAL Lab		Mukul Puranik: 12-2pm MCB 1st floor Hall
He Jiang: 3-5pm, SAL Lab	Victor: 3 – 4:45 pm, SAL 242	Victor: 3 – 4:45 pm, SAL 242		