**Tim Roughgarden's Online Courses**

(To access live link: CTRL + right CLICK mouse)

**http://theory.stanford.edu/~tim/videos.html**

**MOOCs on Coursera**

[Algorithms Specialization](https://www.coursera.org/specializations/algorithms) based on Stanford's undergraduate algorithms course (CS161). Comprises four 4-week courses:

* + [**Part 1: Divide and Conquer, Sorting and Searching, and Randomized Algorithms**](https://www.coursera.org/learn/algorithms-divide-conquer)
    - Covers asymptotic ("Big-oh") notation, sorting and searching, divide and conquer (master method, integer and matrix multiplication, closest pair), and randomized algorithms (QuickSort, contraction algorithm for min cuts).
  + [**Part 2: Graph Search, Shortest Paths, and Data Structures**](https://www.coursera.org/learn/algorithms-graphs-data-structures)
    - Covers data structures (heaps, balanced search trees, hash tables, bloom filters), graph primitives (applications of breadth-first and depth-first search, connectivity, shortest paths), and their applications (ranging from deduplication to social network analysis).
  + [**Part 3: Greedy Algorithms, Minimum Spanning Trees, and Dynamic Programming**](https://www.coursera.org/learn/algorithms-greedy)
    - Covers greedy algorithms (scheduling, minimum spanning trees, clustering, Huffman codes) and dynamic programming (knapsack, sequence alignment, optimal search trees).
  + [**Part 4: Shortest Paths Revisited, NP-Complete Problems and What To Do About Them**](https://www.coursera.org/learn/algorithms-npcomplete)
    - Covers shortest paths (Bellman-Ford, Floyd-Warshall, Johnson), NP-completeness and what it means for the algorithm designer, and strategies for coping with computationally intractable problems (analysis of heuristics, local search).

These courses have appeared on various "top MOOCs of all time" lists, like [here](http://www.onlinecoursereport.com/the-50-most-popular-moocs-of-all-time/) and [here](https://www.class-central.com/collection/top-free-online-courses). New sessions begin every few weeks.

This specialization subsumes the following older versions:

* + [**Algorithms: Design and Analysis (Part I)**](https://www.coursera.org/learn/algorithm-design-analysis/).
    - A self-paced version of this course lives on through the [Stanford Laguanita](https://lagunita.stanford.edu/courses/course-v1:Engineering+Algorithms1+SelfPaced/about) platform.
    - A YouTube playlist of all the lecture videos is available, among other places, [here](https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V).
    - Slides are [here](http://theory.stanford.edu/~tim/mooc/algo1slides.zip).
  + [**Algorithms: Design and Analysis (Part II)**](https://www.coursera.org/learn/algorithm-design-analysis-2/). Slides are [here](http://theory.stanford.edu/~tim/mooc/Algo2Typed.zip).

**Stanford lectures on YouTube**

* [20 Video Lectures on the Design and Analysis of Algorithms](http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms), covering most of the above Coursera MOOCs, for those of you who prefer blackboard lectures (from Stanford's CS161, Winter 2011).
  + Warning/apology: the audio is suboptimal on a few segements of these lectures.
* [20 Video Lectures from A Second Course in Algorithms](https://www.youtube.com/playlist?list=PLEGCF-WLh2RJh2yDxlJJjnKswWdoO8gAc) (Stanford's CS261, Winter 2016). Lecture notes are [here](http://theory.stanford.edu/~tim/w16/w16.html)
  + Topics include: algorithms for maximum flow and minimum cuts (augmenting paths, push-relabel etc.); bipartite matching and generalizations; linear programming; duality; online regret-minimization and the multiplicative weights algorithm; online algorithms; design and analysis of approximation algorithms; the traveling salesman problem; essential tools for the analysis of randomized algorithms; beating brute-force search for NP-hard problems.
* [20 Video Lectures on Algorithmic Game Theory](https://www.youtube.com/playlist?list=PLEGCF-WLh2RJBqmxvZ0_ie-mleCFhi2N4) (from Stanford's CS364A, Fall 2013). Lecture notes are [here](http://theory.stanford.edu/~tim/f13/f13.pdf)
  + Topics include: auction and mechanism design (Vickrey and Myerson auctions, VCG mechanism, etc.); the "price of anarchy" (selfish routing, smoothness, etc.); learning in games (no-regret algorithms, etc.); complexity of Nash equilibria (PLS- and PPAD-completeness). Case studies include: keyword search auctions, combinatorial auctions for spectrum, kidney exchange, and communication network management.
* [20 Video Lectures on Advanced Mechanism Design](https://www.youtube.com/playlist?list=PLEGCF-WLh2RI77PL4gwLld_OU9Zh3TCX9) (from Stanford's CS364B, Winter 2014). Lecture notes are [here](http://theory.stanford.edu/~tim/w14/w14.html)
  + Topics include: combinatorial auctions; Walrasian equilibria and the gross substitutes condition; ascending and ex post implementations; truthful approximation mechanisms; black-box reductions; simple auctions and equilibrium guarantees for them (Bayes-Nash smoothness, etc.); Border's theorem; multi-parameter revenue-maximization.
* [20 Video Lectures on Beyond Worst-Case Analysis](https://www.youtube.com/playlist?list=PLEGCF-WLh2RL8jsZpaf2tLHa5LotFEt5b) (from Stanford's CS264, Fall 2014). Lecture notes are [here](http://theory.stanford.edu/~tim/f14/f14.html)
  + Topics include: instance optimality; parameterized analysis and condition numbers; stable clustering; models of data (pseudorandomness, locality, diffuse adversaries, etc.); distributional models; smoothed analysis; planted and semi-random graph models; exact recovery (compressive sensing, LP decoding). Motivating problems drawn from online algorithms, machine learning, computational geometry, graph partitioning, linear programming, hashing, etc.

[Home](http://theory.stanford.edu/~tim/)