**Topic Resources**

Teaching Material

Content:

[Topic 1: Introduction 2](#_t15qk4xdrl0)

[Topic 2: Data Structures + Libraries 3](#_j6h8emsmwry6)

[Topic 3: Problem Solving Paradigms: Searching, Sorting and Divide and Conquer. 4](#_kdkyy9x6wllt)

[Topic 4: Greedy Algorithms 5](#_atczbfbugui4)

[Topic 5: Dynamic Programming 6](#_bvr5pg6ltiva)

[Topic 6: Graphs 7](#_pscj7f3cfh2d)

[Topic 7: Mathematics and Number Theory 9](#_px0sx1s6chiy)

[Topic 8: Network Flow 10](#_m498g5vcxxh6)

[Topic 9: Computational Geometry and Geometry Algorithms 10](#_pud8lax9ay40)

[Topic 10: Strings 11](#_mue68u66dm7j)

[Topic 11: Combinatorics 12](#_6yscketc65qy)

[Topic 10: Shortest Path Algorithm 13](#_sjgabl11rcu)

[Problem Assessment: 14](#_le4avjrhwxil)

[Article on ICPC: 14](#_hpabm33lntqo)

[Ugly Codes and how to make them better? 14](#_hhl8xqjcame0)

[Notebooks 15](#_andskaxsa4yt)

[Problem Set Collection 15](#_2nrcmys3ksq7)

[Algorithm Collection: 15](#_hhe0ifpbhl5u)

[Problem Set Collection Topicwise for Practice and Tutorials: 15](#_iey9q9jnirgq)

# **Topic 1:** Introduction

**Problem Sets:**

1. [Carnegie Mellon University](https://dmoj.ca/contest/cmu15295s23c0)
2. [Stanford University](http://web.stanford.edu/class/cs97si/assn1.html)
3. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/01_introduction) (kattis)
4. [Reykjavik University warmup](https://github.com/SuprDewd/T-414-AFLV/tree/2016/00_warmup) (kattis)
5. [TAMU](https://tamu.kattis.com/courses/CSCE430/2021Spring)  (includes all assignments) (kattis)
6. [University of Porto](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps00.html) (kattis)
7. [University of Singapore - introduction](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/bvhuy8/problems) (kattis)
8. [University of Singapore - Ad Hoc](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/ybgxci/problems) (kattis)
9. [CS104c: Competitive Programming](https://www.cs.utexas.edu/users/downing/cs104c/Schedule.html) (codeforces) (includes all assignments)
10. [CS 290-CP1: Introduction to Competitive Programming](https://www.cs.purdue.edu/homes/ninghui/courses/CPi/cp1_2020_fall.html) (kattis + codeforces) (all assignments under additional problems)
11. [Introduction to Competitive Programming](https://www.cs.purdue.edu/homes/ninghui/courses/390_Spring20/20_cp2_lectures.html) (kattis + codeforces) (all assignments under additional problems)

**Slides:**

1. [University of Stanford](http://web.stanford.edu/class/cs97si/01-introduction.pdf)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/blob/2016/01_introduction/aflv_01_introduction.pdf)
3. [Illinois University - Introduction](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/course-introduction/) , [Illinois University - I/O](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/io/), [Illinois University - Ad Hoc](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/ad-hoc/)
4. [University of California](https://www.dropbox.com/sh/czhtmbko3xrrz20/AAD2xaSyTqRFYAS4hW_MVbU1a?dl=0) (video lectures included)
5. [UCF -how to use platforms](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/practice/Tutorial-ContestSites.pdf)
6. [ITMO - what are programming competitions](https://courses.edx.org/assets/courseware/v1/46e702c36f355de05d0be44754c47a1d/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/What_is_PC.pdf)
7. [ITMO - Rules of Programming Competitions](https://courses.edx.org/assets/courseware/v1/8b124777c3fd209005c01ca67777610e/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/Rules.pdf)
8. [ITMO - Benefits of Programming Competitions](https://courses.edx.org/assets/courseware/v1/e0456e1711cd7f9841e49f60895fcc8a/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/Benefits.pdf) , [ITMO - Programming Competitions](https://courses.edx.org/assets/courseware/v1/c17b79a7af996f4a47858c31f183b7fd/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/Algorythmic_PC.pdf)
9. [ITMO - Example Problem](https://courses.edx.org/assets/courseware/v1/62481dbc35538d21b8937924fc81b9cd/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/Problem_example.pdf)
10. [Why Should I Take Competitive Programming? | CS 491 CAP (Spring 2022)](https://uiuc-cs491cap.netlify.app/docs/getting-started/why/)

**Contests:**

**Outcome of the week and Notes:**

1. [Introduction to the Course - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week1)
2. [Solving Problems on Platforms](http://isaac.lsu.edu/class_2021_fall/solve.php)
3. [Categories of Problem](http://isaac.lsu.edu/class_2021_fall/categories.php)
4. [University of Porto](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week00.html)
5. [Teamwork in Programming Contests: 3 \* 1 = 4](http://isaac.lsu.edu/class_2021_spring/local/3n.html)
6. [I/O - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week2)

# 

# **Topic 2:** Data Structures + Libraries

**Problem Sets:**

1. [Assignment 3: Data Structures](http://web.stanford.edu/class/cs97si/assn3.html) - Stanford
2. [Reykjavik University part 1](https://github.com/SuprDewd/T-414-AFLV/tree/2016/02_data_structures) (kattis)
3. [Reykjavik University part 2](https://github.com/SuprDewd/T-414-AFLV/tree/2016/03_data_structures) (kattis)
4. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/problemsets/problem-set-1/)
5. [Sublinear complexity data structures (map, set, priority queue) - University of Porto](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps01.html)
6. [UOS](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/sozn8w/problems) (kattis)

**Slides:**

1. [Stanford University](http://web.stanford.edu/class/cs97si/03-data-structures.pdf)
2. [Reykjavik University part 1](https://github.com/SuprDewd/T-414-AFLV/blob/2016/02_data_structures/aflv_02_data_structures.pdf)
3. [Reykjavik University part 2](https://github.com/SuprDewd/T-414-AFLV/blob/2016/03_data_structures/aflv_03_data_structures.pdf)
4. [Illinois: Arrays and Vectors](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/arrays-and-vectors/)
5. [Illinois: Stacks and Queues](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/stacks-and-queues/)
6. [Purdue: Linked List](https://docs.google.com/presentation/d/1QBdcftkmxYXt0QNgyu36Mgvnpg0chA1QoJTxEzyinsc/edit#slide=id.g7e55042a2a_0_0)
7. [ITMO: Linear Data Structure](https://courses.edx.org/assets/courseware/v1/dfb2be8f6c147a1bbc04b7f8971f090e/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-02_w2.pdf)
8. [ITMO: Vector](https://courses.edx.org/assets/courseware/v1/1ee8696e49edb6f5a2aabda8943cfded/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-03_w2.pdf)
9. [ITMO: List](https://courses.edx.org/assets/courseware/v1/dbf6f3565661dbd66a24c62e221a9e23/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-04_w2.pdf)

**Contests:**

**Outcome of the week and Notes:**

1. [String Libraries - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week3)
2. [Big O Notation and Fundamental Data Structures - University of Porto](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week01.html)
3. [Good Blog Post Resources about Algorithm and Data Structures - Codeforces](https://codeforces.com/blog/entry/13529)
4. [CMU](https://scs.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=65e8e4e2-da84-44f6-ba1a-ac370031b0ae) (video on built in algorithms and data structures)

# 

# **Topic 3:** Problem Solving Paradigms: Searching, Sorting and Divide and Conquer.

**Problem Sets:**

1. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/04_problem_solving_paradigms) (kattis)
2. [UCF - Binary Search](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/team-contests/binsearch/practice/)
3. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps02.html> (codeforces)

**Slides:**

1. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/blob/2016/04_problem_solving_paradigms/aflv_04_problem_solving_paradigms.pdf)
2. [Illinois: Complete Search](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/complete-search/)
3. [Illinois: Divide and Conquer](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/divide-and-conquer/)
4. [UCF: Binary Search](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/BinSearchApps.pdf)
5. [ITMO: Sorting](https://courses.edx.org/assets/courseware/v1/411262858f4a22dd04b1ed1e135be9c6/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-01__5_.pdf)
6. [ITMO: Insertion Sort](https://courses.edx.org/assets/courseware/v1/0e689c8ea0d83bb074db6ba252b61f64/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-02__2_.pdf)
7. [ITMO: When to Sort?](https://courses.edx.org/assets/courseware/v1/024445187db9ec958d2ddf9cc6b4bf5c/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-03__2_.pdf)
8. [ITMO: Quick Sort](https://courses.edx.org/assets/courseware/v1/2d964caf9caa62020815c8f36e2f4759/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-04__2_.pdf) , [ITMO: Quick Sort Modification](https://courses.edx.org/assets/courseware/v1/f29491fb9883589551b680552c93290d/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-05__1_.pdf)
9. [ITMO: Merge Sort](https://courses.edx.org/assets/courseware/v1/70d5cd51a1e449d05270516cc3562869/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-06_new.pdf)
10. [ITMO: Priority Queue and Binary Heap](https://courses.edx.org/assets/courseware/v1/c6182dc6dc43431a6e03174ee0e5f03b/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/week_4_lecture_6.pdf)
11. [ITMO: Integer Sorting](https://courses.edx.org/assets/courseware/v1/690707b95dc9cadcc2d699d3e8414202/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-08__1_.pdf)
12. [ITMO: Guidelines for Sorting Libraries](https://courses.edx.org/assets/courseware/v1/6b07f577907fbfd59533964292f71a8f/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-09__1_.pdf)
13. [ITMO: Intro to Binary Search](https://courses.edx.org/assets/courseware/v1/162de5d3ee508557825323c4021715c9/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-10__3_.pdf) , [ITMO: Implementation of Binary Search](https://courses.edx.org/assets/courseware/v1/3699f1556ea3dbdbd408379476157b5e/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-11__1_.pdf)

**Contests:**

1. <https://codeforces.com/group/KIrM1Owd8u/contest/424909> (binary search) CMU

**Outcome of the week and Notes:**

1. [Sorting and Searching - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week4)
2. [Backtracking and Exhaustive Search - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week7)
3. [Study Materials for Sorting and Searching](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week02.html)
4. [University of Porto](https://contest.cs.cmu.edu/295/s21/lectures/binary-search.html) (binary search code)
5. [Lecture 03: Binary Search](https://scs.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=04034e38-7527-4adb-a43a-ac370031b0d9) (CMU Lecture video)

# **Topic 4:** Greedy Algorithms

**Problem Sets:**

1. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/05_greedy_algorithms) (kattis)
2. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/ind-contests/greedy/practice/)
3. [UOC](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/w4j2ec/problems) (kattis)

**Slides:**

1. [Princeton](https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/04GreedyAlgorithmsI.pdf)
2. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/greedy-algorithms/)
3. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Greedy-Contests.pdf)

**Contests:**

# 

# **Topic 5:** Dynamic Programming

**Problem Sets:**

1. [UCF 1](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/team-contests/dp1/practice/)
2. [UCF 2](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/team-contests/dp2/practice/)
3. [Porto I](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps05.html)
4. [Porto II](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps06.html)
5. [UOC](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/t7dueq/problems) (kattis)

**Slides:**

1. [Stanford University](http://web.stanford.edu/class/cs97si/04-dynamic-programming.pdf)
2. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/dynamic-programming-1/) , [Illinois: Advanced Dynamic Programming](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/advanced-dp/)
3. [Purdue University](https://docs.google.com/presentation/d/1cVtthENwZSp0yVWji8y5bFd6i0hFzRIsvgDrUMP-CdQ/edit#slide=id.g5f9d6f9215_1_130)
4. [Purdue University](https://docs.google.com/presentation/d/1tBi5sZrjec8tZNPr1ekpm98y6VPSNNPJnZnUoRlSaSk/edit)
5. [UCF DP1](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/COP4516-DP1.pdf)
6. [UCF DP2](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/COP4516-DP2.pdf)
7. [University of California](https://www.dropbox.com/sh/itg4cheg1dir3yp/AAAsiYtMGEFadKGyvMcgp0eQa?dl=0&preview=lecture2_dp.pdf)

**Contests:**

1. [Practice Contest](https://codeforces.com/group/KIrM1Owd8u/contest/267992)
2. <https://codeforces.com/group/KIrM1Owd8u/contest/426372>

**Videos:**

[CP1 Topic 3: Intro to Dynamic Programming - YouTube](https://www.youtube.com/playlist?list=PL-zAiSWFgxEDz3FB91YIPP-Fs3djtDqVY)

**Outcome of the week and Notes:**

1. [Basic Principles of Dynamic Programming - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week10)
2. [Concepts and Study Material - Porto](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week06.html)
3. [Minimum Spanning Trees, Shortest Distances notes - Stony Brooks](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week08.html)
4. [Introduction to Dynamic Programming - CMU](https://www.cs.cmu.edu/~15451-f21/lectures/lec08-dp1.pdf)
5. [Dijkstra Algorithm -CMU](https://www.cs.cmu.edu/~15451-f21/lectures/lec09-dp2.pdf)

# **Topic 6:** Graphs

**Problem Sets:**

1. [Stanford](http://web.stanford.edu/class/cs97si/assn6.html)
2. [Reykjavik University part 1](https://github.com/SuprDewd/T-414-AFLV/tree/2016/07_graphs_1) (kattis)
3. [Reykjavik University part 2](https://github.com/SuprDewd/T-414-AFLV/blob/2016/08_graphs_2/aflv_08_graphs_2.pdf) (kattis)
4. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/problemsets/problem-set-2/)
5. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/problemsets/problem-set-3/)
6. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/ind-contests/dfsbfstopsort/practice/)
7. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/ind-contests/distmst/practice/)
8. [Cumulative Sums and Fenwick Tree](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps04.html)
9. [Graphs I (DFS, BFS, topological sorting, articulation points, bridges, euler paths, ssc)](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps07.html)  (codeforces)
10. [Graphs II (distances: dijkstra, bellman-ford, floyd; MST: prim+kruskall; max flow)](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps08.html) (codeforces)
11. [University of Singapore - Graphs](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/yfausk/problems) (kattis)
12. [UOC](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/zr4cqc/problems) (kattis)

**Slides:**

1. [Stanford University](http://web.stanford.edu/class/cs97si/06-basic-graph-algorithms.pdf)
2. [Reykjavik University part 1](https://github.com/SuprDewd/T-414-AFLV/blob/2016/07_graphs_1/aflv_07_graphs_1.pdf)
3. [Reykjavik University part 2](https://github.com/SuprDewd/T-414-AFLV/tree/2016/08_graphs_2)
4. Trees: [Illinois: BST and Heap](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/bsts-and-heaps/), [Basic Graphs](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/basic-graphs/), [Traversals](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/traversals/), [Segment Trees](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/segment-trees/), [Fenwick Trees](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/fenwick-trees/), [Minimum Spanning Trees](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/minimum-spanning-trees/)
5. [for Programming Contests Quick Intro A general tree is a connected graph with n vertices and n-1 edges, with no cycles.](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/COP4516-TreeNotes.pdf)
6. [University Of California](https://www.dropbox.com/sh/8bzyqmsocz9pmv6/AADp5X7EIWabcqoedNp7iFaWa?dl=0&preview=lecture3_graph.pdf)
7. Graphs:

UCF:

[Unweighted Graph Algorithms: DFS, BFS and Topological Sort](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Graphs-I-Notes.pdf) , [Weighted Graph Algorithms: Minimum Spanning Tree, Shortest Distance](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Graphs-II-Notes.pdf)

Illinous:

[Graphs 2](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/graphs-2/), [Graphs 3](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/graphs-3/),

1. Union Find:

Purdue: [UF\_MST](https://docs.google.com/presentation/d/1tl1d_rc0ivlfa866AdXkqlMsSlYToBMTstLkOCMwfKk/edit#slide=id.g720fece285_0_154)

1. [ITMO: Intro to DFS](https://courses.edx.org/assets/courseware/v1/74535879a09b9f994cfc75c8965f0a87/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-03_week5.pdf)
2. [ITMO: Graphs](https://courses.edx.org/assets/courseware/v1/6ef0efe202bf2e3b12036f05711d381b/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-01_week5.pdf)
3. [ITMO: Graphs 2](https://courses.edx.org/assets/courseware/v1/0db1b9c8d7cd4dc4b0398af562c258fb/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-02_week5.pdf)
4. [ITMO: Topological Sort](https://courses.edx.org/assets/courseware/v1/2db72868d7738a548eef405460f35d46/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-05_week-05.pdf)
5. [ITMO: BFS](https://courses.edx.org/assets/courseware/v1/3292dde85fcc7abe492976538789e550/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-03_week6.pdf)

**Outcome of the week and Notes:**

1. <https://www3.cs.stonybrook.edu/~skiena/392/notes/week8>
2. <https://www3.cs.stonybrook.edu/~skiena/392/notes/week9>
3. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week03.html>
4. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week04.html>

# **Topic 7:** Mathematics and Number Theory

**Problem Sets:**

1. [Stanford](http://web.stanford.edu/class/cs97si/assn2.html)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/09_mathematics) (kattis)
3. [UOC](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/ci99z4/problems) (kattis)
4. [UOC](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/hbnspz/problems) (kattis)

**Slides:**

1. [Stanford University](http://web.stanford.edu/class/cs97si/02-mathematics.pdf)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/blob/2016/09_mathematics/aflv_10_mathematics.pdf)
3. [Illinois: Prime Numbers](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/prime-numbers/)
4. [Illinois: GCD](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/gcd/)
5. [Illinois: Games](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/games/)
6. [Illinois: Contest Strategy](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/contest-strategy/)

**Contests:**

**Outcome of the week:**

**Notes:**

1. [Summation, Libraries, Divisibility - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week5)
2. [Modular arithmetic and primality - Stony Brooks](https://www3.cs.stonybrook.edu/~skiena/392/notes/week6)

# **Topic 8:** Network Flow

**Problem Sets:**

1. [Stanford Assignment 8: Network Flow Problems](https://web.stanford.edu/class/cs97si/assn8.html)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/10_graphs_3_network_flow) (kattis)
3. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/team-contests/netflow/practice/)
4. [UOS](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/w4j2ec/problems) (kattis)

**Slides:**

1. [Network Flow Problems | CS 97SI | Stanford University](http://web.stanford.edu/class/cs97si/08-network-flow-problems.pdf)
2. [Princeton: max-flow and min-cut problems ‣ Ford–Fulkerson algorithm ‣ max-flow min-cut theorem ‣ capacity-scaling algorithm ‣](https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/07NetworkFlowI.pdf)

[‣ bipartite matching ‣ disjoint paths ‣ extensions to max flow ‣ survey design ‣ airline scheduling ‣ image segmenta](https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/07NetworkFlowII.pdf)

1. [Illinois: Network Flow](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/network-flow/)
2. [UCF: Network Flow](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/NetworkFlow.pdf)
3. [University of California](https://www.dropbox.com/sh/a1e8uczeh5qyp6c/AACrxcbTfqjMWFxuhc2hIelha?dl=0)

# **Topic 9:** Computational Geometry and Geometry Algorithms

**Problem Sets:**

1. [Stanford](http://web.stanford.edu/class/cs97si/assn9.html)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/12_geometry) (kattis)
3. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/team-contests/geo2d/practice/)
4. [UOS](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/pf96kz/problems) (kattis)

**Slides:**

1. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/blob/2016/12_geometry/aflv_12_geometry.pdf)
2. [Illinois: Lines n Vectors](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/points-lines-vectors/)
3. [Illinois: 2D Shapes](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/shapes/)
4. [Illinois: Convex Hull](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/convex-hull/)
5. [Illinois: The Bitmask Technique](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/bitmask-technique/)
6. [UCF](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Geometry-Nadeem.pdf)
7. [UCF - Geometry](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Geometry-UCFNotes.pdf)
8. [UCF - Computational Geometry](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Geometry-USACO.pdf)
9. [UCF - Geometry 2D](http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/notes/Geometry-2D.pdf)
10. [University of California](https://www.dropbox.com/sh/swm238wyi1km9jd/AAAVUjcAOTqzp650pzrgs1oua?dl=0)
11. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps10.html>

**Contests:**

**Outcome of the week and Notes:**

1. <https://www3.cs.stonybrook.edu/~skiena/392/notes/week11>
2. <https://www3.cs.stonybrook.edu/~skiena/392/notes/week12>
3. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week10.html>

# **Topic 10:** Strings

**Problem Sets:**

1. [Stanford Assignment 10: String Algorithms](http://web.stanford.edu/class/cs97si/assn10.html)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/tree/2016/11_strings) (kattis)
3. <https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/ps09.html> (codeforces)
4. [UOS](https://nus.kattis.com/courses/CS3233/CS3233_S2_AY2223/assignments/k9ykyk/problems) (kattis)

**Slides:**

1. [Stanford String Algorithms](http://web.stanford.edu/class/cs97si/10-string-algorithms.pdf)
2. [Reykjavik University](https://github.com/SuprDewd/T-414-AFLV/blob/2016/11_strings/aflv_11_strings.pdf)
3. [Illinois: String Matching](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/string-matching/), [Illinois: Strings n Dynamic Programming](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/strings-and-dynamic-programming/), <https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/suffix-trees/>
4. [University of California](https://www.dropbox.com/sh/ogkukvzmtpg6ij5/AAAfPDfTFCG7kKU7d9cVUj4ha?dl=0)

**Contests:**

**Outcome of the week and Notes:**

1. [Suffix and Prefix Trees](https://www.dcc.fc.up.pt/~pribeiro/aulas/pc2122/week09.html)

# **Topic 11:** Combinatorics

**Problem Sets:**

1. [Stanford](http://web.stanford.edu/class/cs97si/assn5.html)
2. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/problemsets/problem-set-5/)

**Slides:**

1. [Stanford University](http://web.stanford.edu/class/cs97si/05-combinatorial-games.pdf)
2. [Illinois Combinatorics - Fibonacci, Binomial, Catalan](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/combinatorics/)
3. [University of California](https://www.dropbox.com/sh/6tm41g6ghi006ba/AAAnuR8ZzHXNKJeves3FgTFLa?dl=0)

**Contests:**

**Outcome of the week:**

# **Topic 10:** Shortest Path Algorithm

**Problem Sets:**

1. [StanfordAssignment 7: Shortest Path Algorithms](http://web.stanford.edu/class/cs97si/assn7.html)
2. [Illinois](https://courses.engr.illinois.edu/cs491cap/fa2019/problemsets/problem-set-4/)

**Slides:**

1. [Stanford](http://web.stanford.edu/class/cs97si/07-shortest-path-algorithms.pdf)
2. [Single Source Shortest Paths](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/sssp/), [All Source Shortest Path](https://courses.engr.illinois.edu/cs491cap/fa2019/lectures/apsp/)
3. [How to Win Coding Competitions: Secrets of Champions Week 6: Algorithms on Graphs 2 Lecture 5: All Pairs Shortest Paths](https://courses.edx.org/assets/courseware/v1/5a9f5501087a06344deacf06d4376b8b/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-05_week6.pdf)
4. [ITMO: Single Source Shortest Path](https://courses.edx.org/assets/courseware/v1/229dc81e4984caf94b11fb5bf5cb8560/asset-v1:ITMOx+I2CPx+3T2017+type@asset+block/lecture-04_week6.pdf)

**Contests:**

**Contests:**

<https://github.com/SuprDewd/T-414-AFLV/tree/2016/05a_problem_session_1> topics: data structures, problem solving paradigms, greedy algorithms

<https://github.com/SuprDewd/T-414-AFLV/tree/2016/09a_problem_session_2>

Topics: graphs, mathematics, dynamic programming

# 

# 

# 

# **Problem Assessment:**

[Activity of Reading and Reviewing a Problem](https://www3.cs.stonybrook.edu/~skiena/392/reading.pdf)

[Attributes of a good problem set](http://isaac.lsu.edu/class_2021_fall/problem-set.php)

[Problem Package Format Specification](https://www.kattis.com/problem-package-format/)

[Writing Problem Section](https://cs.baylor.edu/~hamerly/courses/4144_22s/#handouts)

# 

# Article on ICPC:

<https://icpc.global/>

# Ugly Codes and how to make them better?

<https://www3.cs.stonybrook.edu/~skiena/392/notes/ugly-code>

# Notebooks

* [**CodeLibrary**](http://code-library.herokuapp.com/) **(Java and C++) |** [**Github**](https://github.com/indy256/codelibrary)
* [**Stanford Notebook**](https://github.com/jaehyunp/stanfordacm/raw/master/notebook.pdf) **(C++) |** [**Github**](https://github.com/jaehyunp/stanfordacm)
* [**spaghetti-source**](https://github.com/spaghetti-source/algorithm) **(C++)**

# Problem Set Collection

<https://cpbook.net/methodstosolve?oj=kattis&topic=all&quality=all> (kattis)

<https://purdue.kattis.com/courses/CS490-CP3> (kattis) (5 offerings)

<https://purdue.kattis.com/courses/CS290-CP1> (kattis) (4 offerings)

<https://purdue.kattis.com/courses/CS390-CP2> (kattis) (9 offerings)

<https://takeuforward.org/interview-experience/strivers-cp-sheet/>

# Algorithm Collection:

<https://cp-algorithms.com/#navigation>

Resource:

<https://github.com/lnishan/awesome-competitive-programming>

<https://codeforces.com/blog/entry/13529>

# Problem Set Collection Topicwise for Practice and Tutorials:

1. [Problem Sets](https://progvar.fun/problemsets)

This website contains problems for each topic from Codeforces, Codechef, Atocoder, UVa OJ, Spoj and Hacker Rank. This website has good content for beginners and experienced Competitive Programmers who want to solve problems based on a specific topic.

1. <https://codedigger1.tech/topicwise/practice>
2. <https://codeforces.com/blog/entry/57282> (Best Tutorials topicwise)
3. <https://codeforces.com/blog/entry/23054> (Includes: Coding Calendars, Coding Streams and Youtube Videos, Language Specifiers)