

Competitive Programming Algorithms Boot-Camp

In March of 2016, I gave a series of algorithms lectures at the South Dakota School of Mines and Technology to prepare for the 2016 ACM ICPC World Finals. Each lecture was accompanied by code examples and competitive programming problems utilizing the algorithm in the solution. These lectures were recorded, and are now available online as training materials for other ICPC competitors.

The remainder of this document outlines the topics covered, including links to lecture videos. A selection of practice problems for each topic is also provided; these problems come from a variety of online judging systems. Lecture example code, when indicated, is available in the GitHub repo (<https://github.com/rmkrohn/AlgorithmsBootCamp>) in the corresponding topic directory.

These materials are intended to fill specific knowledge gaps for students already familiar with competitive programming, and not as an introduction. Topics were selected with a junior-level computer science student in mind; background in data structures and algorithms is strongly recommended. Some lectures build upon previous topics, so videos should be viewed in order when possible.

Rachel Krohn
krohnrm@gmail.com

Lecture topics are listed in order of original delivery. Alternatively, you may wish to work through them by area:

Data Structures

- Segment Trees
- Fenwick Trees
- Disjoint Set Union-Find

String Algorithms

- Knuth-Morris-Pratt (KMP) String Matching
- Suffix Arrays
- Aho-Corasick String Matching

Graph Algorithms

- Floyd-Warshall All-Pairs Shortest Path
- Ford-Fulkerson Max Flow

Computational Geometry

- Line Segment Any-Pair Intersect

1: Segment Trees

Covers basic space-efficient segment tree implementation for range sums before introducing a version utilizing lazy updates. Practice problems rely on segment trees to solve different types of range queries.

Lecture video: <https://youtu.be/Wnc6fvwpu7Q>

Example code: `segment.cpp`
`segmentLazy.cpp`

Practice problems:

Interval Product	https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=3977
Supercomputer	https://open.kattis.com/problems/supercomputer
Worst Weather Ever	https://open.kattis.com/problems/worstweather

2: Fenwick Trees

Covers the Fenwick Tree (or Binary Indexed Tree) data structure and their use in computing interval sums. Includes both basic conceptual discussion and implementation.

Lecture video: https://youtu.be/aYzZzNe_8nM

Example code: `fenwick.cpp` (and accompanying input file `fenwick.in`)

Practice problems:

Fenwick Tree	https://open.kattis.com/problems/fenwick
Turbo	https://open.kattis.com/problems/turbo
Jewel Heist	https://open.kattis.com/problems/jewelheist
Juggler	https://open.kattis.com/problems/juggler

3: Knuth-Morris-Pratt (KMP) String Matching

Covers the Knuth-Morris-Pratt string matching algorithm for linear-time string searches.

Lecture video: <https://youtu.be/GICIJ2ZZzk>

Example code: `kmp.cpp` (and accompanying input file `kmp.in`)

Practice problems:

String Matching	https://open.kattis.com/problems/stringmatching
Fibonacci Words	https://open.kattis.com/problems/fibonacci
Software Bugs	https://open.kattis.com/problems/bugs
Help!	https://open.kattis.com/problems/help2

4: Suffix Arrays

Covers suffix array construction in $O(n \lg(n))$ time and applications.

Lecture video: <https://youtu.be/3L0uTe3BJqs>

Example code: `suffixSorting.cpp` (and accompanying input file `suffixSorting.in`)

Practice problems:

Suffix Sorting	https://open.kattis.com/problems/suffixsorting
Repeated Substrings	https://open.kattis.com/problems/substrings
Dvaput	https://open.kattis.com/problems/dvaput
Stammering Aliens	https://open.kattis.com/problems/aliens
Life Forms	https://open.kattis.com/problems/lifeforms

5: Floyd-Warshall All-Pairs Shortest Path

Covers the Floyd-Warshall algorithm to compute the length of the shortest path between all pairs of vertices in a graph.

Lecture video: <https://youtu.be/jPAkyuxPShE>

Practice problems:

All Pairs Shortest Path <https://open.kattis.com/problems/allpairspath>

Arbitrage? <https://open.kattis.com/problems/arbitrage>

6: Ford-Fulkerson Max Flow

Covers the Ford-Fulkerson algorithm for finding max flow in a graph, the Edmonds-Karp implementation, and algorithm applications.

Lecture video: <https://youtu.be/jEqA1rxKpKQ>

Practice problems:

Maximum Flow <https://open.kattis.com/problems/maxflow>

Minimum Cut <https://open.kattis.com/problems/mincut>

The King of the North <https://open.kattis.com/problems/thewingofthenorth>

Avoiding the Apocalypse <https://open.kattis.com/problems/avoidingtheapocalypse>

Catering <https://open.kattis.com/problems/catering>

Tourist <https://open.kattis.com/problems/tourist>

7: Disjoint Set Union-Find

Covers the disjoint set union-find data structure, including find and union operations.

Lecture video: <https://youtu.be/ChbWDLVtLPk>

Example code: `unionfind.cpp` (and accompanying input file `unionfind.in`)

Practice problems:

Union-Find <https://open.kattis.com/problems/unionfind>

Almost Union-Find <https://open.kattis.com/problems/almostunionfind>

Virtual Friends <https://open.kattis.com/problems/virtualfriends>

8: Aho-Corasick String Matching

Covers the Aho-Corasick string matching algorithm for simultaneous matching of a dictionary of strings.

Lecture video: <https://youtu.be/o-MX0unBk8A>

Example code: `trie.cpp`
`multimatch.cpp` (and accompanying input file `multimatch.in`)

Practice problems:

String Multimatching <https://open.kattis.com/problems/stringmultimatching>

Password Suspects https://onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=3517

Obscene Words Filter <https://acm.timus.ru/problem.aspx?space=1&num=1269>

9: Line Segment Any-Pair Intersect

Covers the theory behind the Shamos-Hoey algorithm for the any-pair intersect problem and the extended Bentley-Ottmann algorithm for listing all intersections in a set of line segments.

Lecture video: https://youtu.be/r_an6j2ib3c

Practice problems:

Simple Polygon <https://open.kattis.com/problems/polygon>

Intersecting Rectangles <https://open.kattis.com/problems/intersectingrectangles>