Video link

Script link

This is done on a google doc because the list of resources will probably eventually exceed the youtube description length limit. It will also likely be updated from time to time, so feel free to check back and/or suggest things to add.

Things directly mentioned in the video:

Codeforces blog about mentality:
Self-deception: maybe why you're still grey after practicing every day - Codeforces

Video with more details about organization:
Candidate Master in 1 Year - This Strategy Works Wonders

Videos about dealing with failure:
Competitive Programming: How to Learn from Contests
The Dark Side of Competitive Programming

Video and blog about creating logic:
Competitive Programming: How to Approach a Problem (as a red coder)
How to get started with solving Ad-Hoc tasks on codeforces

Practice sites - see below

Topics directly mentioned in the video:

0-999

[Vital]

- Brute force (trying every possibility)
- Sorting (your language's library function, you don't need to know the algorithms)
- Strings (just be familiar with them - they're essentially arrays of characters)
- Number theory - floor, ceil, modulo -
  Number Theory for Competitive Programming | Topic Stream 9
- Basic time complexity (you only need big-O)

[Helpful]

- Number theory - divisors, factorization -
  Number Theory for Competitive Programming | Topic Stream 9
- STL/your language's library (for example, https://cppreference.com)
- Binary search - Binary Search tutorial (C++ and Python)
- Two pointers

- Binary + bitwise stuff -
  Bitwise Operations for Competitive Programming | Topic Stream 8

1000-1199

[Vital]

- Brute force (trying every possibility)
- Sorting (your language's library function, you don't need to know the algorithms)
- Strings (just be familiar with them - they're essentially arrays of characters)
- Number theory - divisors, factorization, floor, ceil, modulo -
  Number Theory for Competitive Programming | Topic Stream 9
- STL/your language's library (for example, https://cppreference.com)
- Time complexity (you only need big-O)

[Helpful]

- Binary search - Binary Search tutorial (C++ and Python)
- Two pointers
- Binary + bitwise stuff -
  Bitwise Operations for Competitive Programming | Topic Stream 8
- Dynamic programming -
  Dynamic Programming lecture #1 - Fibonacci, iteration vs recursion /
  Dynamic Programming | Introduction
- Basic combinatorics -
  Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- Basic range queries, namely prefix sums

1200-1399

[Vital]

- Number theory - factorization, modular arithmetic, gcd/lcm, prime factor representation - Number Theory for Competitive Programming | Topic Stream 9
- STL/your language's library (for example, https://cppreference.com)
- Binary search - Binary Search tutorial (C++ and Python)
- Two pointers (rarely)
- Basic combinatorics -
  Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- Basic range queries
- Recursion - Recursion for DSA/CP : BEST way to Understand
- Dynamic programming -
  Dynamic Programming lecture #1 - Fibonacci, iteration vs recursion /
  Dynamic Programming | Introduction

[Helpful]

- Binary and bitwise stuff - Bitwise Operations for Competitive Programming | Topic Stream 8
- Graphs/trees
- DSU (alternately named, UFDS)
- Segment trees (as overkill) - Segment Tree Tutorial
- String algorithms (hashing)

1400-1599

[Vital]

- Number theory - factorization, modular arithmetic, gcd/lcm, prime factor representation - Number Theory for Competitive Programming | Topic Stream 9
- STL/your language's library (for example, https://cppreference.com)
- Binary search - Binary Search tutorial (C++ and Python), two pointers (rarely)
- Basic combinatorics - Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- Basic range queries
- Recursion - Recursion for DSA/CP : BEST way to Understand - and dynamic programming - Dynamic Programming lecture #1 - Fibonacci, iteration vs recur… / Dynamic Programming | Introduction
- Basic graphs/trees
- Proofs (not a topic, but worth mentioning)
- Constructives (all you can do is practice these)

[Helpful]

- Combinatorial techniques - Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- Probability/expected value (rarely) - Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- DSU (alternately named, UFDS)
- More advanced graph techniques (e.g. shortest paths/MST)
- Segment trees (as overkill) - Segment Tree Tutorial
- String algorithms (hashing)
- Basic game theory (a bit formal, but I learned from this one)

1600-1899

[Vital]

- Dynamic programming - Dynamic Programming lecture #1 - Fibonacci, iteration vs recursion / Dynamic Programming | Introduction
- Basic graphs/trees
- Number theory - factorization, modular arithmetic, prime factor representation - Number Theory for Competitive Programming | Topic Stream 9

- Basic combinatorics, probability, expected value - Complete Probability/Combinatorics Practice - Noob to Expert | Topic Stream 4
- Bitwise stuff - Bitwise Operations for Competitive Programming | Topic Stream 8
- Data structures: segment tree (Segment Tree Tutorial), STL (library)
- Game theory (relatively rare)
- Proofs (not a topic, but worth mentioning)
- Constructives (all you can do is practice these)

Sites for practice problems:

- Codeforces (problemset)
- CodeChef
- AtCoder (and kenkoooo to find problems)
- USACO
- ICPC (also check your regional site)
- CSES
- Codeforces (EDU)
- Mostafa sheet
- a2oj
- USACO training
- Timus

More TBA

More things

(in no particular order)

- Another roadmap, designed for rating 1000-2400
- USACO Guide - huge collection of resources and topics
- SecondThread - uploads guides for Codeforces rounds and other cool stuff
- Pavel Mavrin - channel with a ton of quality DSA stuff
- Errichto - high-quality tutorials and coverage of google/facebook contests
- Priyansh Agarwal - Codeforces & general stuff
- Utkarsh Gupta - cool DSA tutorials

More TBA