## [**Foundation**](https://www.codechef.com/certification/data-structures-and-algorithms/prepare#)

## **Syllabus:**

The syllabus for each level is mentioned below:

* Basic Data Structures: Arrays, Strings, Stacks, Queues, Asymptotic analysis (Big-O notation), Basic math operations (addition, subtraction, multiplication, division, exponentiation)
* Sqrt(n) primality testing, Euclid’s GCD Algorithm
* Basic Recursion, Greedy Algorithms, Basic Dynamic Programming
* Naive string searching, O(n logn) Sorting, Binary Searching

## **Learning Resources:**

* **Asymptotic analysis (Big-O notation)**
  + Basic
    - youtube.com - [Time complexity of a computer program](https://www.youtube.com/watch?v=V42FBiohc6c&list=PL2_aWCzGMAwI9HK8YPVBjElbLbI3ufctn)
    - youtube.com - [Big-O notation in 5 minutes - The basics](https://www.youtube.com/watch?v=__vX2sjlpXU)
    - youtube.com - [Definition Of Big O Notation - Intro to Theoretical Computer Science](https://www.youtube.com/watch?v=i1F_Uu0bYCc)
    - youtube.com - [Algorithms Lecture 1 -- Introduction to asymptotic notations](https://www.youtube.com/watch?v=aGjL7YXI31Q)
    - iarcs.org.in - [Measuring the efficiency of algorithms](https://www.iarcs.org.in/inoi/online-study-material/topics/efficiency.php)
    - interactivepython.org - [Particularly for Big-O notation](http://interactivepython.org/courselib/static/pythonds/AlgorithmAnalysis/toctree.html)
  + Advanced
    - rob-bell.net - [A beginner's guide to Big O notation](https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/)
    - youtube.com - [Big O Notation, Gayle Laakman McDowell](https://www.youtube.com/watch?v=v4cd1O4zkGw)
    - web.mit.edu - [Big O notation](http://web.mit.edu/16.070/www/lecture/big_o.pdf)
    - youtube.com - [Time and space complexity analysis of recursive programs - using factorial](https://www.youtube.com/watch?v=ncpTxqK35PI)
    - [A very nice tutorial with examples](https://web.archive.org/web/20171215122943/http://eniac.cs.qc.cuny.edu/andrew/csci700/lecture2.pdf)
  + Practice Problems
    - Check some MCQs on space and time complexity [here](https://discuss.codechef.com/questions/122289/multiple-choice-questions-related-to-testing-knowledge-about-time-and-space-complexity-of-a-program).
    - You can see some problems with solutions here: [Time complexity of an algorithm](http://www.iitk.ac.in/esc101/08Jul/lecnotes/practise_sol.pdf)
* **Arrays**
  + Resources
    - codechef.com - [Data Structure Tutorial: Array](https://discuss.codechef.com/questions/87915/data-structure-tutorial-array)
    - cs.cmu.edu - [Arrays](https://www.cs.cmu.edu/~rjsimmon/15122-f14/lec/04-arrays.pdf)
    - geeksforgeeks.org - [Arrays Data Structure](http://www.geeksforgeeks.org/array-data-structure/)
  + Practice Problems
    - codechef.com - [LECANDY](https://www.codechef.com/problems/LECANDY), [editorial](https://discuss.codechef.com/questions/1108/lecandy-editorial)
    - codechef.com - [CNOTE](https://www.codechef.com/problems/CNOTE), [editorial](https://discuss.codechef.com/questions/65992/cnote-editorial) ;
    - codechef.com - [SALARY](https://www.codechef.com/problems/SALARY), [editorial](https://discuss.codechef.com/questions/5144/salary-editorial)
    - codechef.com - [CHN15A](https://www.codechef.com/problems/CHN15A), [editorial](https://discuss.codechef.com/questions/77487/chn15a-editorial)
    - codechef.com - [RAINBOWA](https://www.codechef.com/problems/RAINBOWA), [editorial](https://discuss.codechef.com/questions/107967/rainbowa-editorial)
    - codechef.com - [FRGTNLNG](https://www.codechef.com/problems/FRGTNLNG), [editorial](https://discuss.codechef.com/questions/75211/frgtnlng-editorial)
    - codechef.com - [COPS](https://www.codechef.com/problems/COPS), [editorial](https://discuss.codechef.com/questions/72499/cops-editorial)
* **Strings**
  + Resources
    - tutorialspoint.com - [C++ strings](https://www.tutorialspoint.com/cplusplus/cpp_strings.htm)
    - guru99.com - [Java strings](https://www.guru99.com/java-strings.html)
    - docs.python.org - [Python strings](https://docs.python.org/2/library/string.html)
    - tutorialspoint.com - [Python strings](https://www.tutorialspoint.com/python/python_strings.htm)
    - geeksforgeeks.org - [Many string questions](http://www.geeksforgeeks.org/string-data-structure/)
  + Practice Problems
    - codechef.com - [CSUB](https://www.codechef.com/problems/CSUB), [editorial](https://discuss.codechef.com/questions/47237/csub-editorial)
    - codechef.com - [LAPIN](https://www.codechef.com/problems/LAPIN), [editorial](https://discuss.codechef.com/questions/13991/lapin-editorial)
* **Stack and Queue**
  + Resources
    - geeksforgeeks.org - [Stack Data Structure](http://www.geeksforgeeks.org/stack-data-structure/)
    - geeksforgeeks.org - [Introduction and Array Implementation](http://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/)
    - tutorialspoint.com - [Data Structures Algorithms](https://www.tutorialspoint.com/data_structures_algorithms/stack_algorithm.htm)
    - cs.cmu.edu - [Stacks](https://www.cs.cmu.edu/~wlovas/15122-r11/lectures/10-stacks.pdf)
    - cs.cmu.edu - [Stacks and Queues](https://www.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html)
    - cs.cmu.edu - [Stacks and Queues](https://www.cs.cmu.edu/~rjsimmon/15122-s13/09-queuestack.pdf)
  + Practice Problems
    - spoj.com - [JNEXT](http://www.spoj.com/problems/JNEXT/)
    - spoj.com - [STPAR](http://www.spoj.com/problems/STPAR/)
    - spoj.com - [ONP](http://www.spoj.com/problems/ONP/)
    - codechef.com - [COMPILER](https://www.codechef.com/problems/COMPILER)
    - spoj.com - [MMASS](http://www.spoj.com/problems/MMASS/)
    - spoj.com - [HISTOGRA](http://www.spoj.com/problems/HISTOGRA/)
    - codeforces.com - [D. Maximum Xor Secondary](http://codeforces.com/problemset/problem/281/D)
    - spoj.com - [ANARC09A](http://www.spoj.com/problems/ANARC09A/)
    - codeforces.com - [C. Minimal string](http://codeforces.com/contest/797/problem/C)
    - codeforces.com - [B. Alternating Current](http://codeforces.com/contest/343/problem/B)
    - codeforces.com - [C. Longest Regular Bracket Sequence](http://codeforces.com/contest/5/problem/C)
* **Basic math operations (addition, subtraction, multiplication, division, exponentiation)**
  + codechef.com - [A tutorial on Fast Modulo Multiplication](https://discuss.codechef.com/questions/20451/a-tutorial-on-fast-modulo-multiplication-exponential-squaring)
* **Euclid’s GCD Algorithm**
  + Resources
    - youtube.com - [Mycodeschool video](https://www.youtube.com/watch?v=7HCd074v8g8)
    - khanacademy.org - [The Euclidean Algorithm](https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm)
    - geeksforgeeks.org - [Example program to find gcd in c++:](http://www.geeksforgeeks.org/c-program-find-gcd-hcf-two-numbers/)
* **Prime Numbers, divisibility of numbers**
  + Resources:
    - Only O(sqrt(n)) algorithm for finding whether a number is a prime, factorization of a number.
    - [Finding prime factors by taking the square root](https://math.stackexchange.com/questions/1039519/finding-prime-factors-by-taking-the-square-root/1039525#1039525)
  + Practice Problems:
    - community.topcoder.com - [DivisorInc](https://community.topcoder.com/stat?c=problem_statement&pm=6186&rd=9823)
    - community.topcoder.com - [Prime Polynom](https://community.topcoder.com/stat?c=problem_statement&pm=4475&rd=8012)
    - community.topcoder.com - [Prime Anagrams](https://community.topcoder.com/stat?c=problem_statement&pm=3458&rd=5869)
    - community.topcoder.com - [Refactoring](https://community.topcoder.com/stat?c=problem_statement&pm=2986&rd=5862)
* **Basic Recursion**
  + Resources
    - topcoder.com - [An Introduction to Recursion, Part 1](https://www.topcoder.com/community/data-science/data-science-tutorials/an-introduction-to-recursion-part-1/)
    - topcoder.com - [An Introduction to Recursion: Part 2](https://www.topcoder.com/community/data-science/data-science-tutorials/an-introduction-to-recursion-part-2/)
    - geeksforgeeks.org - [Recursion](http://www.geeksforgeeks.org/recursion/) ;(along with questions)
    - web.mit.edu - [Recursion](http://web.mit.edu/6.005/www/fa15/classes/10-recursion/)
    - csee.umbc.edu - [Recursion](https://www.csee.umbc.edu/~chang/cs202.f98/readings/recursion.html) ;(Examples with exercises)
    - loveforprogramming.quora.com - [Backtracking, Memoization & Dynamic Programming](https://loveforprogramming.quora.com/Backtracking-Memoization-Dynamic-Programming)
    - byte-by-byte - [Recursion for Coding Interviews](https://www.byte-by-byte.com/recursion/)
  + Practice Problems
    - codechef.com - [NOKIA](https://www.codechef.com/problems/NOKIA), [editorial](https://discuss.codechef.com/questions/1280/nokia-editorial)
    - codechef.com - [TRISQ](https://www.codechef.com/problems/TRISQ), [editorial](https://discuss.codechef.com/questions/64151/trisq-editorial)
    - codechef.com - [LFSTACK](https://www.codechef.com/problems/LFSTACK), [editorial](https://discuss.codechef.com/questions/84364/lfstack-editorial)
    - codechef.com - [FICE](https://www.codechef.com/problems/FICE), [editorial](https://discuss.codechef.com/questions/85839/fice-editorial)
* **Greedy Algorithms**
  + Resources
    - iarcs.org.in - [Greedy Algorithms](https://www.iarcs.org.in/inoi/online-study-material/topics/greedy.php)
    - iarcs.org.in - [Greedy Algorithms](https://www.iarcs.org.in/inoi/online-study-material/problems/buffalos-soln.php#solution)
    - topcoder.com - [Greedy Algorithms](https://www.topcoder.com/community/data-science/data-science-tutorials/greedy-is-good/)
    - [Greedy Algorithms](http://jeffe.cs.illinois.edu/teaching/algorithms/book/04-greedy.pdf)
  + Practice Problems
    - codechef.com - [TACHSTCK](https://www.codechef.com/problems/TACHSTCK), [editorial](https://discuss.codechef.com/questions/18267/tachstck-editorial)
    - codechef.com - [CIELRCPT](https://www.codechef.com/problems/CIELRCPT), [editorial](https://discuss.codechef.com/questions/1748/cielrcpt-editorial)
    - codechef.com - [MAXDIFF](https://www.codechef.com/problems/MAXDIFF), [editorial](https://discuss.codechef.com/questions/8114/maxdiff-editorial)
    - codechef.com - [CHEFST](https://www.codechef.com/problems/CHEFST), [editorial](https://discuss.codechef.com/questions/77629/chefst-editorial)
    - codechef.com - [CAKEDOOM](https://www.codechef.com/problems/CAKEDOOM), [editorial](https://discuss.codechef.com/questions/1119/cakedoom-editorial)
    - codechef.com - [CLETAB](https://www.codechef.com/problems/CLETAB), [editorial](https://discuss.codechef.com/questions/49342/cletab-editorial)
    - codechef.com - [TADELIVE](https://www.codechef.com/problems/TADELIVE), [editorial](https://discuss.codechef.com/questions/60005/tadelive-editorial)
    - codechef.com - [MANYCHEF](https://www.codechef.com/problems/MANYCHEF), [editorial](https://discuss.codechef.com/questions/5606/manychef-editorial)
    - codechef.com - [MMPROD](https://www.codechef.com/problems/MMPROD), [editorial](https://discuss.codechef.com/questions/82151/mmprod-editorial)
    - codechef.com - [CHEFTMA](https://www.codechef.com/problems/CHEFTMA), [editorial](https://discuss.codechef.com/questions/78212/cheftma-editorial)
    - codechef.com - [STICKS](https://www.codechef.com/problems/STICKS), [editorial](https://discuss.codechef.com/questions/82568/sticks-editorial)
    - spoj.com - [BAISED](http://www.spoj.com/problems/BAISED/)
    - spoj.com - [BALIFE](http://www.spoj.com/problems/BALIFE/)
    - spoj.com - [GCJ101BB](http://www.spoj.com/problems/GCJ101BB/)
    - codechef.com - [FGFS](https://www.codechef.com/problems/FGFS)
    - codechef.com - [KNPSK](https://www.codechef.com/problems/KNPSK)
    - codechef.com - [LEMUSIC](https://www.codechef.com/problems/LEMUSIC)
    - spoj.com - [ARRANGE](http://www.spoj.com/problems/ARRANGE/)
    - spoj.com - [FASHION](http://www.spoj.com/problems/FASHION/)
* **Dynamic programming (Basic DP)**
  + Resources
    - medium.freecodecamp.org - [Demystifying Dynamic Programming](https://medium.freecodecamp.org/demystifying-dynamic-programming-3efafb8d4296)
    - iarcs.org.in - [Dynamic Programming - Tiling](https://www.iarcs.org.in/inoi/online-study-material/topics/dp-tiling.php)
    - topcoder.com - [Dynamic Programming – From Novice to Advanced](https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/)
    - illinois.edu - [Dynamic Programming](http://jeffe.cs.illinois.edu/teaching/algorithms/book/03-dynprog.pdf) ;(Exercises are recommended)
    - codechef.com - [Dynamic Programming](https://www.codechef.com/wiki/tutorial-dynamic-programming)
    - geeksforgeeks.org - [Dynamic Programming](http://www.geeksforgeeks.org/dynamic-programming/) ;(Contains a lot of practice sessions)
    - MIT OCW (Contains some Advanced topics as well)
      * [Dynamic Programming I](https://www.youtube.com/watch?v=OQ5jsbhAv_M)
      * [Dynamic Programming II](https://www.youtube.com/watch?v=ENyox7kNKeY)
      * [Dynamic Programming III](https://www.youtube.com/watch?v=ocZMDMZwhCY)
      * [Dynamic Programming IV](https://www.youtube.com/watch?v=tp4_UXaVyx8)
  + Practice Problems
    - codechef.com - [ALTARAY](https://www.codechef.com/problems/ALTARAY), [editorial](https://discuss.codechef.com/questions/80137/altaray-editorial)
    - codechef.com - [DELISH](https://www.codechef.com/problems/DELISH), [editorial](https://discuss.codechef.com/questions/13727/delish-editorial)
    - codechef.com - [DBOY](https://www.codechef.com/problems/DBOY), [editorial](https://discuss.codechef.com/questions/4443/dboy-editorial)
    - codechef.com - [XORSUB](https://www.codechef.com/problems/XORSUB), [editorial](https://discuss.codechef.com/questions/58422/xorsub-editorial)
    - codechef.com - [GRID](https://www.codechef.com/problems/GRID), [editorial](https://discuss.codechef.com/questions/51454/grid-editorial)
    - codechef.com - [TADELIVE](https://www.codechef.com/problems/TADELIVE), [editorial](https://discuss.codechef.com/questions/60005/tadelive-editorial)
    - codechef.com - [FROGV](https://www.codechef.com/problems/FROGV), [editorial](https://discuss.codechef.com/questions/47239/frogv-editorial)
    - codechef.com - [MATRIX2](https://www.codechef.com/problems/MATRIX2), [editorial](https://discuss.codechef.com/questions/22491/matrix2-editorial)
    - codechef.com - [AMSGAME2](https://www.codechef.com/problems/AMSGAME2), [editorial](https://discuss.codechef.com/questions/10495/amsgame2-editorial)
    - spoj.com - [MDOLLS](http://www.spoj.com/problems/MDOLLS/)
    - spoj.com - [MSTICK](http://www.spoj.com/problems/MSTICK/)
    - spoj.com - [MCARDS](http://www.spoj.com/problems/MCARDS/)
    - spoj.com - [MIXTURES](http://www.spoj.com/problems/MIXTURES/)
    - spoj.com - [SAMER08D](https://www.spoj.pl/problems/SAMER08D/)
    - spoj.com - [AIBOHP](https://www.spoj.pl/problems/AIBOHP/)
* **Naive string searching**
  + Resources
    - geeksforgeeks.org - [Naive Pattern Searching](http://www.geeksforgeeks.org/searching-for-patterns-set-1-naive-pattern-searching/)
* **Sorting**
  + [khanacademy.org](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/a/sorting)
  + [visualgo.net](https://visualgo.net/en/sorting?slide=1)
  + [iarcs.org.in](https://www.iarcs.org.in/inoi/online-study-material/topics/sorting.php)
  + Merge sort
    - youtube.com - [Merge sort algorithm](https://www.youtube.com/watch?v=TzeBrDU-JaY)
    - Practice Problems  
      codechef.com -[MRGSRT](https://www.codechef.com/problems/MRGSRT)
  + Quick sort
    - youtube.com - [Quicksort algorithm](https://www.youtube.com/watch?v=COk73cpQbFQ)
    - Practice Problems  
      codechef.com -[TSORT](https://www.codechef.com/problems/TSORT)
  + Counting sort
    - geeksforgeeks.org - [Counting Sort](http://www.geeksforgeeks.org/counting-sort/)
    - Practice Problems
      * codechef.com - [TACHSTCK](https://www.codechef.com/problems/TACHSTCK), [editorial](https://discuss.codechef.com/questions/18267/tachstck-editorial)
      * codechef.com - [STICKS](https://www.codechef.com/problems/STICKS), [editorial](https://discuss.codechef.com/questions/82568/sticks-editorial)
* **Binary Search**
  + Resources
    - [topcoder.com](https://www.topcoder.com/community/data-science/data-science-tutorials/binary-search/) (Try solving problems of Simple and Moderate level as mentioned in the end of the link)
    - [codechef.com](https://www.codechef.com/wiki/tutorial-binary-search)
    - [usfca.edu](https://www.cs.usfca.edu/~galles/visualization/Search.html)
    - [khanacademy.org](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search)
  + Detailed Theoretical analysis
    - [cmu.edu](https://www.cs.cmu.edu/~fp/courses/15122-f10/lectures/03-binsearch.pdf) (A theoretical analysis)
  + Problems
    - geeksforgeeks.org - [Binary Search](http://www.geeksforgeeks.org/binary-search) (Contains some solved problems)
    - codechef.com - [STRSUB](https://www.codechef.com/problems/STRSUB), [editorial](https://discuss.codechef.com/questions/66064/strsub-editorial)
    - codechef.com - [ASHIGIFT](https://www.codechef.com/problems/ASHIGIFT), [editorial](https://discuss.codechef.com/questions/66867/ashigift-editorial)
    - codechef.com - [STACKS](https://www.codechef.com/problems/STACKS), [editorial](https://discuss.codechef.com/questions/75205/stacks-editorial)
    - codechef.com - [DIVSET](https://www.codechef.com/problems/DIVSET), [editorial](https://discuss.codechef.com/questions/107068/divset-editorial)
    - codechef.com - [LOWSUM](https://www.codechef.com/problems/LOWSUM), [editorial](https://discuss.codechef.com/questions/29659/lowsum-editorial)
    - codechef.com - [SNTEMPLE](https://www.codechef.com/problems/SNTEMPLE), [editorial](https://discuss.codechef.com/questions/99456/sntemple-editorial)
    - codechef.com - [SNAKEEAT](https://www.codechef.com/problems/SNAKEEAT), [editorial](https://discuss.codechef.com/questions/98802/snakeeat-editorial)
    - codechef.com - [SCHEDULE,](https://www.codechef.com/problems/SCHEDULE) [editorial](https://discuss.codechef.com/questions/92702/schedule-editorial)
    - codechef.com - [RIGHTTRI](https://www.codechef.com/problems/RIGHTTRI), [editorial](https://discuss.codechef.com/questions/82375/righttri-editorial)
    - codechef.com - [FORESTGA](https://www.codechef.com/problems/FORESTGA), [editorial](https://discuss.codechef.com/questions/81382/forestga-editorial)
    - codechef.com - [CHEFHCK2](https://www.codechef.com/problems/CHEFHCK2),[editorial](https://discuss.codechef.com/questions/6650/chefhck2-editorial)
    - spoj.com - [ABCDEF](http://www.spoj.com/problems/ABCDEF)
    - spoj.com - [NOTATRI](http://www.spoj.com/problems/NOTATRI)
    - spoj.com - [SCALE](http://www.spoj.com/problems/SCALE)
    - spoj.com - [SUMFOUR](http://www.spoj.com/problems/SUMFOUR)
    - spoj.com - [SUBSUMS](http://www.spoj.com/problems/SUBSUMS)
    - spoj.com - [ANARC05B](http://www.spoj.com/problems/ANARC05B)
    - spoj.com - [RENT](http://www.spoj.com/problems/RENT)
    - spoj.com - [PIE](http://www.spoj.com/problems/PIE)
    - spoj.com - [MKUHAR](http://www.spoj.com/problems/MKUHAR)
    - spoj.com - [SVADA](http://www.spoj.com/problems/SVADA)
    - spoj.com - [SUBS](http://www.spoj.com/problems/SUBS)
* Past Test - <https://www.codechef.com/FLPAST01>
* Test 1 - [codechef.com/FLMOCK01](https://www.codechef.com/FLMOCK01)
* Test 2 - [codechef.com/FLMOCK02](https://www.codechef.com/FLMOCK02)
* Test 3 - [codechef.com/FLMOCK03](https://www.codechef.com/FLMOCK03)
* Test 4 - [codechef.com/FLMOCK04](https://www.codechef.com/FLMOCK04)

## 

# **Advanced**

This level is intended to test that the candidate has a very good grasp of algorithms and data structures, and can solve most problems that arise in practice.

## **Syllabus:**

* Heaps (priority queue)
* Disjoint Set Union
* Segment Trees
* Binary Index Tree (Fenwick tree)
* Trees (traversals, tree dynamic programming)
* Finding Lowest Common Ancestors (O(log N) solution where N is number of nodes).
* Graph Algorithms:
  + Finding connected components and transitive closures.
  + Shortest-path algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall)
  + Minimum spanning tree (Prim and Kruskal algorithms)
  + Biconnectivity in undirected graphs (bridges, articulation points)
  + Strongly connected components in directed graphs
  + Topological Sorting
  + Euler path, tour/cycle.
* Modular arithmetic including division, inverse
* Amortized Analysis
* Divide and Conquer
* Advanced Dynamic Programming problems (excluding the dp optimizations which are added in expert level)
* Sieve of Eratosthenes

**Resources**

* **Heaps (priority queue)**
  + Resources
    - [cs.cmu.edu](https://www.cs.cmu.edu/~wlovas/15122-r11/lectures/15-priorqs.pdf)
    - [eecs.wsu.edu](http://www.eecs.wsu.edu/~ananth/CptS223/Lectures/heaps.pdf)
    - [geeksforgeeks.org](http://www.geeksforgeeks.org/binary-heap/)
    - [visualgo.net](https://visualgo.net/en/heap)
    - [iarcs.org.in](https://www.iarcs.org.in/inoi/online-study-material/topics/heaps.php)
  + Practice Problems
    - codechef.com - [IPCTRAIN](https://www.codechef.com/JULY17/problems/IPCTRAIN), [editorial](https://discuss.codechef.com/questions/105180/ipctrain-editorial)
    - codechef.com - [ANUMLA](https://www.codechef.com/problems/ANUMLA), [editorial](https://discuss.codechef.com/questions/53529/anumla-editorial)
    - codechef.com - [KSUBSUM](https://www.codechef.com/problems/KSUBSUM), [editorial](https://discuss.codechef.com/questions/4018/ksubsum-editorial)
    - codechef.com - [RRATING](https://www.codechef.com/problems/RRATING), [editorial](https://discuss.codechef.com/questions/1585/rrating-editorial)
    - codechef.com - [TSECJ05](https://www.codechef.com/problems/TSECJ05), [editorial](https://discuss.codechef.com/questions/103875/chef-and-his-software-tsecj05-editorial)
    - spoj.com - [WEIRDFN](http://www.spoj.com/problems/WEIRDFN/)
    - codechef.com - [CAPIMOVE](https://www.codechef.com/problems/CAPIMOVE), [editorial](https://discuss.codechef.com/questions/90266/capimove-editorial)
    - spoj.com - [RMID2](http://www.spoj.com/problems/RMID2/)
    - spoj.com - [LAZYPROG](http://www.spoj.com/problems/LAZYPROG/)
    - spoj.com - [EXPEDI](http://www.spoj.com/problems/EXPEDI/)
    - [acm.timus.ru](http://acm.timus.ru/problem.aspx?space=1&num=1306)
    - baylor.edu - [Maze Checking and Visualization](https://icpcarchive.ecs.baylor.edu/index.php?option=onlinejudge&page=show_problem&problem=122)
    - codechef.com - [MOSTDIST](https://www.codechef.com/problems/MOSTDIST), [editorial](https://discuss.codechef.com/questions/56239/mostdist-editorial)
    - hackerearth - [Break the Door](https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/break-the-door/)
* **Disjoint Set Union**
  + Resources
    - [topcoder.com](https://www.topcoder.com/community/data-science/data-science-tutorials/disjoint-set-data-structures/)
    - [harvard.edu](http://people.cs.georgetown.edu/jthaler/ANLY550/lec6.pdf)
    - [ucdavis.edu](http://web.cs.ucdavis.edu/~amenta/w10/trevisanNotes.pdf)
    - [visualgo.net](https://visualgo.net/en/ufds)
    - [hackerearth.com](https://www.hackerearth.com/practice/data-structures/disjoint-data-strutures/basics-of-disjoint-data-structures/tutorial/)
  + Practice Problems
    - codechef.com - [GALACTIK](https://www.codechef.com/problems/GALACTIK), [editorial](https://discuss.codechef.com/questions/18004/galactik-editorial)
    - codechef.com - [DISHOWN](https://www.codechef.com/problems/DISHOWN), [editorial](https://discuss.codechef.com/questions/47241/dishown-editorial)
    - codechef.com - [JABO](https://www.codechef.com/problems/JABO), [editorial](https://discuss.codechef.com/questions/3710/jabo-editorial)
    - codechef.com - [PARITREE](https://www.codechef.com/problems/PARITREE), [editorial](https://discuss.codechef.com/questions/79920/paritree-editorial)
    - codechef.com - [FILLMTR](https://www.codechef.com/problems/FILLMTR), [editorial](https://discuss.codechef.com/questions/109357/fillmtr-editorial)
    - [B. Mike and Feet](http://codeforces.com/problemset/problem/547/B)
    - [D. Quantity of Strings](http://codeforces.com/contest/151/problem/D)
    - codechef.com - [SETELE](https://www.codechef.com/problems/SETELE), [editorial](https://discuss.codechef.com/questions/87909/setele-editorial)
    - codechef.com - [MAZE](https://www.codechef.com/problems/MAZE), [editorial](https://discuss.codechef.com/questions/81591/maze-editorial)
    - codechef.com - [MAGICSTR](https://www.codechef.com/problems/MAGICSTR), [editorial](https://discuss.codechef.com/questions/77347/magical-strings-editorial)
    - codechef.com - [MTRWY](https://www.codechef.com/problems/MTRWY), [editorial](https://discuss.codechef.com/questions/66031/mtrwy-editorial)
    - codechef.com - [BIGOF01](https://www.codechef.com/problems/BIGOF01), [editorial](https://discuss.codechef.com/questions/68454/bigof01-editorial)
    - codechef.com - [FIRESC](https://www.codechef.com/problems/FIRESC), [editorial](https://discuss.codechef.com/questions/7269/firesc-editorial)
* **Segment Trees**
  + Resources
    - [wcipeg.com](http://wcipeg.com/wiki/Segment_tree)
    - [topcoder.com](https://www.topcoder.com/community/data-science/data-science-tutorials/range-minimum-query-and-lowest-common-ancestor/#Segment_Trees)
    - [kartikkukreja.wordpress.com](https://kartikkukreja.wordpress.com/2014/11/09/a-simple-approach-to-segment-trees/)
    - [visualgo.net](https://visualgo.net/en/segmenttree)
    - [iarcs.org.in](https://www.iarcs.org.in/inoi/online-study-material/topics/segment-tree.php)
  + Practice Problems
    - spoj.com - [GSS1](http://www.spoj.pl/problems/GSS1/)
    - spoj.com - [GSS2](http://www.spoj.pl/problems/GSS2/)
    - codeforces.com - [Classic Segment Tree](http://codeforces.com/blog/entry/15890) (Expert Level)
    - spoj.com - [IOPC1207](http://www.spoj.pl/problems/IOPC1207/)
    - spoj.com - [ORDERSET](http://www.spoj.com/problems/ORDERSET/)
    - spoj.com - [HELPR2D2](http://www.spoj.com/problems/HELPR2D2/)
    - spoj.com - [ANDROUND](http://www.spoj.com/problems/ANDROUND/)
    - spoj.com - [HEAPULM](http://www.spoj.com/problems/HEAPULM/)
    - spoj.com - [NICEDAY](http://www.spoj.com/problems/NICEDAY/)
    - spoj.com - [YODANESS](http://www.spoj.com/problems/YODANESS/)
    - spoj.com - [DQUERY](http://www.spoj.pl/problems/DQUERY/)
    - spoj.com - [KQUERY](http://www.spoj.pl/problems/KQUERY/)
    - spoj.com - [FREQUENT](http://www.spoj.pl/problems/FREQUENT/)
    - spoj.com - [GSS3](http://www.spoj.pl/problems/GSS3/)
    - spoj.com - [GSS4](http://www.spoj.pl/problems/GSS4)
    - spoj.com - [GSS5](http://www.spoj.pl/problems/GSS5/)
    - spoj.com - [KGSS](http://www.spoj.pl/problems/KGSS/)
    - spoj.com - [HELPR2D2](http://www.spoj.pl/problems/HELPR2D2/)
    - spoj.com - [BRCKTS](http://www.spoj.pl/problems/BRCKTS/)
    - spoj.com - [CTRICK](http://www.spoj.pl/problems/CTRICK/)
    - spoj.com - [MATSUM](http://www.spoj.pl/problems/MATSUM/)
    - spoj.com - [RATING](http://www.spoj.pl/problems/RATING/)
    - spoj.com - [RRSCHED](http://www.spoj.pl/problems/RRSCHED/)
    - spoj.com - [SUPPER](http://www.spoj.pl/problems/SUPPER/)
    - spoj.com - [ORDERS](http://www.spoj.pl/problems/ORDERS/)
    - codechef.com - [LEBOBBLE](https://www.codechef.com/problems/LEBOBBLE)
    - codechef.com - [QUERY](https://www.codechef.com/problems/QUERY)
    - spoj.com - [TEMPLEQ](http://www.spoj.com/problems/TEMPLEQ)
    - spoj.com - [DISUBSTR](http://www.spoj.com/problems/DISUBSTR/)
    - spoj.com - [QTREE](http://www.spoj.pl/problems/QTREE/)
    - spoj.com - [QTREE2](http://www.spoj.pl/problems/QTREE2/)
    - spoj.com - [QTREE3](http://www.spoj.pl/problems/QTREE3/)
    - spoj.com - [QTREE4](http://www.spoj.com/problems/QTREE4/)
    - spoj.com - [QTREE5](http://www.spoj.com/problems/QTREE5/)
  + Problems on segment tree with lazy propagation
    - spoj.com - [HORRIBLE](http://www.spoj.com/problems/HORRIBLE) (must do basic lazy propagation problem)
    - spoj.com - [LITE](http://www.spoj.com/problems/LITE) (a nice lazy propagation problem)
    - spoj.com - [MULTQ3](http://www.spoj.com/problems/MULTQ3/) (another nice lazy propagation problem)
    - codechef.com - [CHEFD](https://www.codechef.com/problems/CHEFD)
    - codechef.com - [FUNAGP](https://www.codechef.com/problems/FUNAGP) (a difficult lazy propagation problem.)
    - [RPAR](http://www.spoj.com/problems/RPAR/) (a difficult and nice lazy propagation)
    - codechef.com - [ADDMUL](https://www.codechef.com/problems/ADDMUL)
    - spoj.com - [SEGSQRSS](http://www.spoj.com/problems/SEGSQRSS/) (a difficult lazy propagation problem)
    - spoj.com - [KGSS](http://www.spoj.com/problems/KGSS/)
    - codeforces.com - [C. Circular RMQ](http://codeforces.com/contest/52/problem/C)
    - codeforces.com - [E. Lucky Queries](http://codeforces.com/contest/145/problem/E) (must do hard problem on lazy propagation)
    - codeforces.com - [E. A Simple Task](http://codeforces.com/contest/558/problem/E)
    - codeforces.com - [C. DZY Loves Fibonacci Numbers](http://codeforces.com/contest/446/problem/C) (important problem to do, introduces some nice properties over lazy propagation)
    - codeforces.com - [D. The Child and Sequence](http://codeforces.com/contest/438/problem/D)
    - codeforces.com - [E. Lucky Array](http://codeforces.com/contest/121/problem/E)
* **Binary Index Tree (Fenwick tree)**
  + Resources
    - [topcoder.com](https://www.topcoder.com/community/data-science/data-science-tutorials/binary-indexed-trees/)
    - [iarcs.org.in](https://www.iarcs.org.in/inoi/online-study-material/topics/binary-index-tree.php)
    - [visualgo.net](https://visualgo.net/en/fenwicktree)
  + Practice Problems:  
    Please solve the problems mentioned in the above segment tree practice problems section. Note that usually, it's difficult to do range updates in binary indexed trees. Mostly, it is used for for range query and point update. However, you can check the following article for checking how some simple specific kind of range updates can be performed on binary indexed tree (http://petr-mitrichev.blogspot.in/2013/05/fenwick-tree-range-updates.html). Note that range updates on BIT is not a part of the syllabus.
    - spoj.com - [INVCNT](http://www.spoj.com/problems/INVCNT/)
    - spoj.com - [TRIPINV](http://www.spoj.com/problems/TRIPINV/)
* **Trees (traversals)**
  + Resources
    - [slideshare.net](https://www.slideshare.net/ecomputernotes/traversal-of-a-binary-tree-10682319)
    - [iarcs.org.in](https://www.iarcs.org.in/inoi/online-study-material/topics/dp-trees.php)
    - [berkeley.edu](https://people.eecs.berkeley.edu/~vazirani/s99cs170/notes/dynamic2.pdf)
  + Practice Problems
    - spoj.com - [TREEORD](http://www.spoj.com/problems/TREEORD/)
* **Finding Lowest Common Ancestors (O(log N) solution where N is number of nodes)**
  + Resources
    - [topcoder.com](https://www.topcoder.com/community/data-science/data-science-tutorials/range-minimum-query-and-lowest-common-ancestor/)
* **Depth First Search, Breadth First Search (Finding connected components and transitive closures)**
  + Resources
    - geeksforgeeks.org - [Connected Components in an undirected graph](http://www.geeksforgeeks.org/connected-components-in-an-undirected-graph/)
    - geeksforgeeks.org - [Transitive closure of a graph](http://www.geeksforgeeks.org/transitive-closure-of-a-graph/)
    - geeksforgeeks.org - [Depth First Traversal or DFS for a Graph](http://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/)
    - iarcs.org.in - [Basic Graph Algorithms](https://www.iarcs.org.in/inoi/online-study-material/topics/graphs.php)
    - visualgo.net - [Graph Traversal](https://visualgo.net/en/dfsbfs)
    - harvard.edu - [Breadth-First Search](http://people.cs.georgetown.edu/jthaler/ANLY550/lec4.pdf)
  + Practice Problems
    - codechef.com - [FIRESC](https://www.codechef.com/problems/FIRESC), [editorial](https://discuss.codechef.com/questions/7269/firesc-editorial)
    - spoj.com - [BUGLIFE](http://www.spoj.com/problems/BUGLIFE/)
    - spoj.com - [CAM5](http://www.spoj.com/problems/CAM5/)
    - spoj.com - [GCPC11J](http://www.spoj.com/problems/GCPC11J/)
    - spoj.com - [KFSTB](http://www.spoj.com/problems/KFSTB/)
    - spoj.com - [PT07Y](http://www.spoj.com/problems/PT07Y/)
    - spoj.com - [PT07Z](http://www.spoj.com/problems/PT07Z/)
    - spoj.com - [LABYR1](http://www.spoj.com/problems/LABYR1/)
    - spoj.com - [PARADOX](http://www.spoj.com/problems/PARADOX/)
    - spoj.com - [PPATH](http://www.spoj.com/problems/PPATH/) ;(must do bfs problem)
    - spoj.com - [ELEVTRBL](http://www.spoj.com/problems/ELEVTRBL/) (bfs)
    - spoj.com - [QUEEN](http://www.spoj.com/problems/QUEEN/) (bfs)
    - spoj.com - [SSORT](http://www.spoj.com/problems/SSORT/) ;(cycles in a graph)
    - spoj.com - [ROBOTGRI](http://www.spoj.com/problems/ROBOTGRI/) ;(bfs)
* **Flood Fill:**
  + Resources
    - Hackerearth - [FloodFill](https://www.hackerearth.com/practice/algorithms/graphs/flood-fill-algorithm/tutorial/)
  + Practice Problems
    - Leetcode - [Floodfill](https://leetcode.com/problems/flood-fill/)
    - Hackerearth - [Logo Hunting](https://www.hackerearth.com/practice/algorithms/graphs/flood-fill-algorithm/practice-problems/algorithm/logo-hunting-2d4068ca/)
* **Shortest-path algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall)**
  + Resources
    - geeksforgeeks.org - [Dijkstra’s shortest path algorithm](http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/)
    - Iarcs.org.in - [Shortest paths](https://www.iarcs.org.in/inoi/online-study-material/topics/shortest-paths.php)
    - Visualgo.net - [Single-Source Shortest Paths (SSSP)](https://visualgo.net/en/sssp)
  + Practice Problems
    - codechef.com - [DIGJUMP](https://www.codechef.com/problems/DIGJUMP), [editorial](https://discuss.codechef.com/questions/44800/digjump-editorial)
    - codechef.com - [AMR14B](https://www.codechef.com/AMR14ROS/problems/AMR14B), [editorial](https://discuss.codechef.com/questions/61701/amr14b-editorial)
    - codechef.com - [INSQ15\_F](https://www.codechef.com/problems/INSQ15_F), [editorial](https://discuss.codechef.com/questions/74790/insq15_f-editorial)
    - codechef.com - [SPSHORT](https://www.codechef.com/problems/SPSHORT), [editorial](https://discuss.codechef.com/questions/64224/spshort-editorial) (slightly difficult dijkstra's problem.)
    - codechef.com - [RIVPILE](https://www.codechef.com/problems/RIVPILE), [editorial](https://discuss.codechef.com/questions/18188/rivpile-editorial)
    - spoj.com - [SHPATH](http://www.spoj.com/problems/SHPATH/)
    - spoj.com - [TRAFFICN](http://www.spoj.com/problems/TRAFFICN/)
    - spoj.com - [SAMER08A](http://www.spoj.com/problems/SAMER08A/)
    - spoj.com - [MICEMAZE](http://www.spoj.com/problems/MICEMAZE/)
    - spoj.com - [TRVCOST](http://www.spoj.com/problems/TRVCOST/)
    - codechef.com - [PAIRCLST](https://www.codechef.com/problems/PAIRCLST), [editorial](https://discuss.codechef.com/questions/79923/pairclst-editorial)
* **Bellman Ford Algorithm**
  + Resources
    - geeksforgeeks.org - [Dynamic Programming - Bellman–Ford Algorithm](http://www.geeksforgeeks.org/dynamic-programming-set-23-bellman-ford-algorithm/)
    - compprog.wordpress.com - ;[One Source Shortest Path - Bellman-Ford Algorithm](https://compprog.wordpress.com/2007/11/29/one-source-shortest-path-the-bellman-ford-algorithm/)
  + Practice Problem
    - community.topcoder.com - [PeopleYouMayKnow](https://community.topcoder.com/stat?c=problem_statement&pm=10580)
    - codeforces.com - [D. Robot Control](http://codeforces.com/problemset/problem/346/D)
    - spoj.com - [ARBITRAG - Arbitrage](http://www.spoj.com/problems/ARBITRAG/) ;(Floyd Warshall)
    - community.topcoder.com - [NetworkSecurity](http://community.topcoder.com/stat?c=problem_statement&pm=10736) ;(Floyd Warshall)
* **Minimum spanning tree (Prim and Kruskal algorithms)**
  + Resources
    - algs4.cs.princeton.edu - [Minimum Spanning Trees](http://algs4.cs.princeton.edu/lectures/43MinimumSpanningTrees.pdf)
    - iarcs.org.in - [Spanning trees](https://www.iarcs.org.in/inoi/online-study-material/topics/spanning-trees.php)
    - visualgo.net - [Spanning Tree](https://visualgo.net/en/mst)
  + Practice Problem
    - spoj.com - [MST](http://www.spoj.com/problems/MST/)
    - spoj.com - [NITTROAD](http://www.spoj.com/problems/NITTROAD/)
    - spoj.com - [BLINNET](http://www.spoj.com/problems/BLINNET/)
    - spoj.com - [CSTREET](http://www.spoj.com/problems/CSTREET/)
    - spoj.com - [HIGHWAYS](http://www.spoj.com/problems/HIGHWAYS/)
    - spoj.com - [IITWPC4I](http://www.spoj.com/problems/IITWPC4I/)
    - codechef.com - [MSTQS](https://www.codechef.com/problems/MSTQS), [editorial](https://discuss.codechef.com/questions/89653/mstqs-editorial)
    - codechef.com - [CHEFGAME](https://www.codechef.com/problems/CHEFGAME), [editorial](https://discuss.codechef.com/questions/8119/chefgame-editorial)
    - codechef.com - [GALACTIK](https://www.codechef.com/problems/GALACTIK), [editorial](https://discuss.codechef.com/questions/18004/galactik-editorial)
    - codechef.com - [GOOGOL03](https://www.codechef.com/problems/GOOGOL03), [editorial](https://discuss.codechef.com/questions/70187/googol03-editorial)
    - spoj.com - [KOICOST](http://www.spoj.com/problems/KOICOST/)
* **Biconnectivity in undirected graphs (bridges, articulation points)**
  + Resources
    - e-maxx-eng.appspot.com - [Finding Bridges in a Graph](https://e-maxx-eng.appspot.com/graph/bridge-searching.html)
    - iarcs.org.in - [Articulation Points](https://www.iarcs.org.in/inoi/online-study-material/topics/articulation-points.php)
    - pisces.ck.tp.edu.tw - [Articulation Points](http://pisces.ck.tp.edu.tw/~peng/index.php?action=showfile&file=fb1f19a9be617037cb419c5d454b184bead47e243)
    - Hackerearth - [Articulation Points and Bridges](https://www.hackerearth.com/practice/algorithms/graphs/articulation-points-and-bridges/tutorial/)
    - Hackerearth - [Biconnected Components](https://www.hackerearth.com/practice/algorithms/graphs/biconnected-components/tutorial/)
  + Practice Problem
    - uva.onlinejudge.org - [Network](https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=251)
    - icpcarchive.ecs.baylor.edu - [Building Bridges](https://icpcarchive.ecs.baylor.edu/index.php?option=onlinejudge&page=show_problem&problem=722)
    - uva.onlinejudge.org - [Tourist Guide](https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1140)
    - acm.tju.edu.cn - [Network](http://acm.tju.edu.cn/toj/showp1026.html)
    - spoj.com - [EC\_P - Critical Edges](http://www.spoj.com/problems/EC_P/)
    - spoj.com - [SUBMERGE - Submerging Islands](http://www.spoj.com/problems/SUBMERGE/)
    - spoj.com - [POLQUERY - Police Query](http://www.spoj.com/problems/POLQUERY/)
    - codeforces.com - [A. Cutting Figure](http://codeforces.com/problemset/problem/193/A)
    - Hackerearth - [Hank & Guests](https://www.hackerearth.com/practice/algorithms/graphs/biconnected-components/practice-problems/algorithm/gaviria-and-guests/)
* **Strongly connected components in directed graphs**
  + Resources
    - iarcs.org.in - [Strongly connected components](https://www.iarcs.org.in/inoi/online-study-material/topics/scc.php)
    - theory.stanford.edu - [Strongly Connected Components](http://theory.stanford.edu/~tim/w11/l/scc.pdf)
  + Practice Problem
    - spoj.com - [ANTTT](http://www.spoj.com/problems/ANTTT/)
    - spoj.com - [CAPCITY](http://www.spoj.com/problems/CAPCITY/)
    - spoj.com - [SUBMERGE](http://www.spoj.com/problems/SUBMERGE/)
    - codechef.com - [MCO16405](https://www.codechef.com/problems/MCO16405), [editorial](https://discuss.codechef.com/questions/91585/mco16405-editorial)
    - spoj.com - [BOTTOM](http://www.spoj.com/problems/BOTTOM/)
    - spoj.com - [BREAK](http://www.spoj.com/problems/BREAK/)
    - community.topcoder.com - [Marble Collection Game](https://community.topcoder.com/stat?c=problem_statement&pm=8488&rd=11125)
    - hackerearth - [A Walk to Remember](https://www.hackerearth.com/practice/algorithms/graphs/strongly-connected-components/practice-problems/algorithm/a-walk-to-remember-qualifier2/)
* **Topological Sorting**
  + Resources
    - geeksforgeeks.org - [Topological Sorting](http://www.geeksforgeeks.org/topological-sorting/)
    - Hackerearth - [Topological Sort](https://www.hackerearth.com/practice/algorithms/graphs/topological-sort/tutorial/)
  + Practice Problem
    - Hackerearth - [Lonely Island](https://www.hackerearth.com/practice/algorithms/graphs/topological-sort/practice-problems/algorithm/lonelyisland-49054110/)
    - spoj.com - [TOPOSORT](http://www.spoj.com/problems/TOPOSORT/) ;
    - codeforces.com - [C. Fox And Names](http://codeforces.com/contest/510/problem/C) ;
    - codechef.com - [RRDAG](https://www.codechef.com/problems/RRDAG), [editorial](https://discuss.codechef.com/questions/47983/rrdag-editorial)
    - spoj.com - [RPLA](http://www.spoj.com/problems/RPLA/)
    - codechef.com - [CL16BF](https://www.codechef.com/problems/CL16BF) (topological sort with dp), [editorial](https://discuss.codechef.com/questions/85994/cl16bf-editorial)
    - spoj.com - [MAKETREE](http://www.spoj.com/problems/MAKETREE/)
* **Euler path, tour/cycle.**
  + Resources
    - math.ku.edu - [Euler Paths and Euler Circuits](https://www.math.ku.edu/~jmartin/courses/math105-F11/Lectures/chapter5-part2.pdf)
    - hackerearth - [Hamiltonian Path](https://www.hackerearth.com/practice/algorithms/graphs/hamiltonian-path/tutorial/)
  + Practice Problem
    - spoj.com - [WORDS1](http://www.spoj.com/problems/WORDS1/)
    - codechef.com - [CHEFPASS](https://www.codechef.com/problems/CHEFPASS), [editorial](https://discuss.codechef.com/questions/2096/chefpass-editorial)
    - codechef.com - [TOURISTS](https://www.codechef.com/problems/TOURISTS), [editorial](https://discuss.codechef.com/questions/90320/tourists-editorial)
    - codeforces.com - [D. New Year Santa Network](http://codeforces.com/contest/500/problem/D)
    - [B. Strongly Connected City](http://codeforces.com/contest/475/problem/B)
    - codechef.com - [PEOPLOVE](https://www.codechef.com/problems/PEOPLOVE)
    - codeforces.com - [D. Tanya and Password](http://codeforces.com/contest/508/problem/D)
    - codeforces.com - [E. One-Way Reform](http://codeforces.com/contest/723/problem/E)
    - spoj.com - [GCPC11C](http://www.spoj.com/problems/GCPC11C/)
    - spoj.com - [MAKETREE](http://www.spoj.com/problems/MAKETREE/)
    - Hackerearth - [Fredo and his Birthday Gift](https://www.hackerearth.com/practice/algorithms/graphs/hamiltonian-path/practice-problems/algorithm/fredo-and-his-birthday-gift-4/)
* **Modular arithmetic including division, inverse**
  + Resources
    - codechef.com - [Fast Modulo Multiplication (Exponential Squaring)](https://discuss.codechef.com/questions/20451/a-tutorial-on-fast-modulo-multiplication-exponential-squaring)
    - codechef.com - [Best known algos for calculating nCr % M](https://discuss.codechef.com/questions/3869/best-known-algos-for-calculating-ncr-m) ;(only for expert level)
* **Amortized Analysis**
  + Resources
    - ocw.mit.edu - [Amortized Analysis](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2012/lecture-notes/MIT6_046JS12_lec11.pdf)
    - wikipedia.org - [Amortized Analysis](https://en.wikipedia.org/wiki/Amortized_analysis)
    - iiitdm.ac.in - [Amortized Analysis](http://www.iiitdm.ac.in/old/Faculty_Teaching/Sadagopan/pdf/ADSA/amortized-analysis.pdf)
* **Divide and Conquer**
  + Resources
    - cs.cmu.edu - [Divide-and-Conquer and Recurrences](http://www.cs.cmu.edu/afs/cs/academic/class/15210-s12/www/lectures/lecture02.pdf)
    - geeksforgeeks.org - [Divide-and-Conquer](http://www.geeksforgeeks.org/category/divide-and-conquer/)
  + Practice Problem
    - codechef.com - [MRGSRT](https://www.codechef.com/problems/MRGSRT), [editorial](https://discuss.codechef.com/questions/70773/mrgsrt-editorial)
    - spoj.com - [HISTOGRA](http://www.spoj.com/problems/HISTOGRA/)
    - codechef.com - [TASTYD](https://www.codechef.com/problems/TASTYD), [editorial](https://discuss.codechef.com/questions/15957/tastyd-editorial)
    - codechef.com - [RESTPERM](https://www.codechef.com/problems/RESTPERM), [editorial](https://discuss.codechef.com/questions/86763/restperm-editorial)
    - codechef.com - [ACM14KP1](https://www.codechef.com/problems/ACM14KP1), [editorial](https://discuss.codechef.com/questions/56980/acm14kp1-editorial)
* **Advanced Dynamic Programming** problems (excluding the dp optimizations which are added in expert level, Please go through the basic DP resources )
  + Resources
    - apps.topcoder.com - [Commonly used DP state domains](http://apps.topcoder.com/forums/?module=Thread&threadID=697369&start=0)
    - apps.topcoder.com - [Introducing Dynamic Programming](http://apps.topcoder.com/forums/?module=Thread&threadID=700080&start=0)
    - apps.topcoder.com - [Optimizing DP solution](http://apps.topcoder.com/forums/?module=Thread&threadID=697925&start=0)
    - codeforces.com - [DP over Subsets and Paths](http://codeforces.com/blog/entry/337)
    - hackerearth - [2D DP](https://www.hackerearth.com/practice/algorithms/dynamic-programming/2-dimensional/tutorial/)
  + Problems for Advanced DP
    - spoj.com - [HIST2](http://www.spoj.com/problems/HIST2/) ;(dp bitmask)
    - spoj.com - [LAZYCOWS](http://www.spoj.com/problems/LAZYCOWS/) ;(dp bitmask)
    - spoj.com - [TRSTAGE](http://www.spoj.com/problems/TRSTAGE/) ;(dp bitmask)
    - spoj.com - [MARTIAN](http://www.spoj.pl/problems/MARTIAN/)
    - spoj.com - [SQRBR](http://www.spoj.pl/problems/SQRBR/)
    - spoj.com - [ACMAKER](http://www.spoj.pl/problems/ACMAKER/)
    - spoj.com - [AEROLITE](http://www.spoj.pl/problems/AEROLITE/)
    - spoj.com - [BACKPACK](https://www.spoj.pl/problems/BACKPACK/)
    - spoj.com - [COURIER](http://www.spoj.pl/problems/COURIER/)
    - spoj.com - [DP](http://www.spoj.pl/problems/DP/)
    - spoj.com - [EDIST](http://www.spoj.pl/problems/EDIST/)
    - spoj.com - [KRECT](http://www.spoj.pl/problems/KRECT/)
    - spoj.com - [GNY07H](http://www.spoj.pl/problems/GNY07H/)
    - spoj.com - [LISA](http://www.spoj.pl/problems/LISA/)
    - spoj.com - [MINUS](http://www.spoj.pl/problems/MINUS/)
    - spoj.com - [NAJKRACI](http://www.spoj.pl/problems/NAJKRACI/)
    - spoj.com - [PHIDIAS](http://www.spoj.pl/problems/PHIDIAS/)
    - spoj.com - [PIGBANK](http://www.spoj.pl/problems/PIGBANK/)
    - spoj.com - [PT07X](http://www.spoj.pl/problems/PT07X/)
    - spoj.com - [VOCV](http://www.spoj.pl/problems/VOCV/)
    - spoj.com - [TOURIST](http://www.spoj.pl/problems/TOURIST/)
    - spoj.com - [MKBUDGET](http://www.spoj.pl/problems/MKBUDGET)
    - spoj.com - [MMAXPER](http://www.spoj.pl/problems/MMAXPER)
    - spoj.com - [ANARC07G](http://www.spoj.pl/problems/ANARC07G)
    - spoj.com - [MENU](http://www.spoj.pl/problems/MENU)
    - spoj.com - [RENT](http://www.spoj.com/problems/RENT/) ;(dp with segment tree/BIT)
    - spoj.com - [INCSEQ](http://www.spoj.com/problems/INCSEQ/) ;(dp with segment tree/BIT)
    - spoj.com - [INCDSEQ](http://www.spoj.com/problems/INCDSEQ/) ;(dp with segment tree/BIT)
    - You can solve some advanced problems from
    - codeforces.com - [Dynamic Programming Type](http://codeforces.com/blog/entry/325)
* **Sieve of Eratosthenes**
  + Resources:
    - codechef.com - [Sieve Methods](https://discuss.codechef.com/questions/54846/please-proofread-the-atricle-for-tutorial-section-on-sieve-methods)
  + Practice Problems
    - spoj.com - [TDKPRIME](http://www.spoj.com/problems/TDKPRIME/)
    - spoj.com - [TDPRIMES](http://www.spoj.com/problems/TDPRIMES/)
    - spoj.com - [ODDDIV](http://www.spoj.com/problems/ODDDIV/) ;(sieve + binary search)
    - spoj.com - [NDIVPHI](http://www.spoj.com/problems/NDIVPHI/) ;O(N) prime testing algorithm)
    - spoj.com - [DIV](http://www.spoj.com/problems/DIV/) ;(divisor sieve)
    - codechef.com - [LEVY](https://www.codechef.com/problems/LEVY), [editorial](https://discuss.codechef.com/questions/8115/levy-editorial)
    - codechef.com - [PRETNUM](https://www.codechef.com/problems/PRETNUM), [editorial](https://discuss.codechef.com/questions/28909/pretnum-editorial)
    - codechef.com - [KPRIME](https://www.codechef.com/problems/KPRIME), [editorial](https://discuss.codechef.com/questions/17915/kprime-editorial)
    - codechef.com - [DIVMAC](https://www.codechef.com/problems/DIVMAC), [editorial](https://discuss.codechef.com/questions/84917/divmac-editorial-unofficial) (segment tree with sieve)
    - codechef.com - [PPERM](https://www.codechef.com/problems/PPERM), [editorial](https://discuss.codechef.com/questions/2555/pperm-editorial) ;(a bit advanced sieve application)
* **General**
  + [Stanford Algoriths 1](https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V)
  + [Stanford Algoriths 2](https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6)
* Test 1 - <https://www.codechef.com/ALPAST01>
* Test 1 - <https://www.codechef.com/ADMOCK01>
* Test 2 - <https://www.codechef.com/ADMOCK02>

[**Expert**](https://www.codechef.com/certification/data-structures-and-algorithms/prepare#)

This level is intended to test that the candidate is an expert in algorithms and data structures, and has a deep understanding of the topics.

## **Syllabus:**

The syllabus for Expert Level is open-ended. Everything in Advanced Level will be included, along with:

* Treaps
* Persistent Data Structures
* HLD
* Centroid Decomposition
* Computational Geometry
* Fast Fourier Transforms
* Game Theory
* Gaussian Elimination
* Dynamic Programming Optimizations (eg. Convex Hull Trick, Divide and Conquer Optimization, Knuth Optimization)
* Advanced String algorithms (Tries, KMP, Aho-Corasik, Suffix arrays, Suffix trees)
* Flows (Max-Flow, Min Cost Max Flow)
* Refer other [handout](https://docs.google.com/document/d/1eP0eIG3wux-E6lAugqneGxVRnFz8IO5oXsctaX02sYk/edit?usp=sharing)
* Past Test 1 - <https://www.codechef.com/ELPAST01>
* Mock Test 1 - <https://www.codechef.com/EXMOCK01>
* Test 2 - <https://www.codechef.com/EXMOCK02>

## Codeforces

* From **rating 1000 to 1400**
  + You can write straight-forward simulation fast. (within 5-10 minutes)
  + You can write brute force fast. (within 5-10 minutes)
  + You can divide the problem into some cases in your brain or in your paper. (e.g. N=2, N=3, or N>=4)
  + Practice: Atcoder Beginner Contest problems, Div 2 Codeforces.
  + Look solutions after 30mins only and editorials after contest
* From **rating 1400 to 1900**
  + Implement major algorithms like: Brute force DP DFS BFS Dijkstra Binary Indexed Tree nCr, nPr Mod inverse Bitmasks Binary Search
  + You can code faster (For example, 5 minutes for R1100 problems, 10 minutes for R1400 problems)
  + Practice: Atcoder problems C within 10 mins, D within 20 minutes
  + Solve algorithm problems and div 2 hard problems and also div 1 easy problems.
  + Next solve Topcoder SRM div1 easy problems for more than 1500 rating.
  + Participate in virtual contests almost every day
* From **rating 1900 to 2200**
  + Solve at least 1000 problems in different problems
  + Equip your code template library including
  + Learn some more algorithms and segment trees also
  + You need to solve very fast: 5 mins for R1100, 10 mins for R1500, 15 mins for R1800, 40 mins for R2000.
  + Decent mathematical thinking and problem solving approaches
* From rating **2200 to 2400**
  + You should be able to solve difficult problems in div 1
  + You should solve boss problems which are ad-hoc or many-step consideration problems
  + You should solve at least 2000 - 4000 problems of of AtCoder Regular Contest, especially ARC058-ARC090, 50+ virtual contests of div1
  + Solve R2800 problems for 2 hours a week

Review your mistakes for improving rating,don’t care about ratings and standing during contest, Try to solve at least two problems a week that is out of your comfort zone.