

## List of Topics for programming Competitions -

1. **Basic Geometry/Euclidean Geometry/Coordfnate Geometry/ [3-D variants of everything].**
2. **Computational Geometry.**
  - a. Graham Scan algorithm for Convex Hull  $O(n * \log(n))$ .
  - b. Online construction of 3-D convex hull in  $O(n^2)$ .
  - c. Bentley Ottmann algorithm to list all intersection points of  $n$  line segments in  $O((n + I) * \log n)$ .
    - Suggested Reading -
      1. [http://softsurfer.com/Archive/algorithm\\_0108/algorithm\\_0108.htm](http://softsurfer.com/Archive/algorithm_0108/algorithm_0108.htm)
  - d. Rotating Calipers Technique.
    - Suggested Reading - <http://cgm.cs.mcgill.ca/~orm/rotcal.html>
    - Problems - Refer the article for a list of problems which can be solved using Rotating Calipers technique.
  - e. Line Sweep/Plane Sweep algorithms -
    - Area/Perimeter of Union of Rectangles.
    - Closest pair of points.
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lineSweep>
    - Problems - Follow the tutorial for list of problems.
  - f. Area of Union of Circles.
  - g. Delayunay Triangulation of  $n$  points in  $O(n * \log n)$ .
  - h. Voronoi Diagrams of  $n$  points in  $O(n * \log n)$  using Fortunes algorithm.
  - i. Point in a polygon problem -
    - $O(n)$  solution without preprocessing.
    - $O(\log n)$  algorithm with  $O(n * \log n)$  preprocessing for convex polygons.
  - j. Problems on computational geometry -
    - [BSHEEP](#), [BULK](#), [SEGVIS](#), [CONDUIT](#), [RUNAWAY](#), [DIRVS](#), [RAIN1](#), [SHAMAN](#), [TCUTTER](#), [LITEPIPE](#), [RHOMBS](#), [FSHEEP](#), [FLBRKLIN](#), [CERC07P](#), [BAC](#), [ALTARS](#), [CERC07C](#), [NECKLACE](#), [CH3D](#), [RECTANGL](#), [POLYSSQ](#), [FOREST2](#), [KPPOLY](#), [RAIN2](#), [SEGMENTS](#), [ARCHPLG](#), [BALLOON](#), [CIRCLES](#), [COMPASS](#), [BOWAMRT](#), [ICERINK](#) on SPOJ.
    - [CultureGrowth](#), [PolygonCover](#) on Topcoder.
  - k. Suggested Reading -
    - Computational Geometry: Algorithms and applications. Mark De Burg.
3. **String Algorithm.**
  - a. KnuthMorrisPratt algorithm.
    - Problems - NHAY, PERIOD on SPOJ.
    - Suggested Reading -
      1. Cormen chapter on Strings.
      2. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=stringSearching>
  - b. Aho Corasick algorithm.
    - Problems - WPUZZLES on SPOJ.
  - c. Suffix Arrays
    - $O(n^2 * \log n)$  Naive method of suffix array construction
    - $O(n * \log n^2)$  method of suffix array construction
    - $O(n * \log n)$  method of suffix array construction.
    - $O(n)$  method of suffix array construction
    - $O(n)$  LCA preprocess on Suffix Arrays to solve a variety of string problems.
  - d. Suffix Trees
    - $O(n)$  construction of Suffix trees using Ukkenon's algorithm.
    - $O(n)$  construction of Suffix Trees if provided with Suffix Arrays using Farach's algorithm.
  - e. Suffix Automata
    - $O(n)$  Suffix Automaton construction.
  - f. Dictionary Of Basic Factors
    - $O(n * \log n)$  method of DBF construction using Radix Sort.
  - g. Manachar's algorithm to find Lengh of palindromic substring of a string centered at a position for each position in the string.  
Runtime ->  $O(n)$ .
  - h. Searching and preprocessing Regular Expressions consisting of '?', '\*'.
  - i. Multi-dimentional pattern matching.
  - j. Problems on Strings [can be solved with a variety of techniques] -
    - [DISUBSTR](#), [PLD](#), [MSTRING](#), [REPEATS](#), [JEWELS](#), [ARCHIVER](#), [PROPKEY](#), [LITELANG](#), [EMOTICON](#), [WORDS](#), [AMCODES](#), [UCODES](#), [PT07H](#), [MINSEQ](#), [TOPALIN](#), [BWHEELER](#), [BEADS](#), [SARRAY](#), [LCS](#), [LCS2](#), [SUBST1](#), [PHRASES](#), [PRETILE](#) on SPOJ
    - [http://www.algorithmist.com/index.php/Category:String\\_algorithms](http://www.algorithmist.com/index.php/Category:String_algorithms)
4. **Basic Graphs [beginner].**
  - a. Representation of graphs as adjacency list, adjacency matrix, incidence matrix and edge list and uses of different representations in different scenarios.
  - b. Breadth First Search.
    - problems -
      1. [PPATH](#), [ONEZERO](#), [WATER](#) on SPOJ
  - c. Depth First Search.
  - d. Strongly Connected Components.
    - problems -
      1. [TOUR](#) and [BOTTOM](#) on SPOJ.
  - e. Biconnected Components, Finding articulation points and bridges].
    - problems -
      1. [RELINETS](#), [PT07A](#) on SPOJ.
  - f. Dijkstra algorithm -
    - problems -
      1. [SHPATH](#) on SPOJ.
  - g. Floyd Warshall algorithm -
    - problems -
      1. [COURIER](#) on SPOJ.
  - h. Minimum Spanning Tree
    - problems -
      1. [BLINNET](#) on SPOJ.
  - i. Flood-fill algorithm
  - j. Topological sort
  - k. Bellman-Ford algorithm.
  - l. Euler Tour/Path.
    - problems - [WORDS1](#) on SPOJ.
  - m. Suggested reading for most of the topics in Graph algorithms -
    - <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=graphsDataStrucs1>.
    - Also refer to the tutorial for problems concerning these techniques.
    - Cormen chapter 22 to 24.
5. **Flow networks/ matching etc etc. [Interdiate/Advanced].**
  - a. Maximum flow using Ford Fulkerson Method.
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=maxFlow>
    - problems - [TAXI](#), [POTHOLE](#), [IM](#), [QUEST4](#), [MUDDY](#), [EN](#), [CABLETV](#), [STEAD](#), [NETADMIN](#), [COCONUTS](#), [OPTM](#) on SPOJ.
  - b. Maximum flow using Dinics Algorithm.
    - Problems - [PROFIT](#) on spoj.
  - c. Minimum Cost Maximum Flow.
    - Successive Shortest path algorithm.
    - Cycle Cancelling algorithm.
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=minimumCostFlow1>
  - d. Maximum weighted Bipartite Matching (Kuhn Munkras algorithm/Hungarian Method)
    - problems - [GREED](#), [SCITIES](#), [TOURS](#) on SPOJ | [http://www.topcoder.com/stat?c=problem\\_statement&pn=8143](http://www.topcoder.com/stat?c=problem_statement&pn=8143)
  - e. Stoer Wagner min-cut algorithm.
  - f. Hopcroft Karp bipartite matching algorithm.
    - problems - [ANGELS](#) on SPOJ.
  - g. Maximum matching in general graph (blossom shrinking)
  - h. Gomory-Hu Trees.
    - i) Problems - [MCQUERY](#) on Spoj.
  - i. Chinese Postman Problem.
    - problems - <http://acm.uva.es/archive/nuevoportal/data/problem.php?p=4039>
    - Suggested Reading - <http://eie507.eie.polyu.edu.hk/ss-submission/B7a/>
  - j. Suggested Reading for the full category ->
    - Network flow - Algorithms and Applications by Ahuja
    - Cormen book chapter 25.
6. **Dynamic Programming.**

- a. Suggested Reading - Dynamic Programming(DP) as a tabulation method
    - Cormen chapter on DP
  - b. Standard problems (you should really feel comfortable with these types)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=8570&rd=12012&rm=269199&cr=7581406](http://www.topcoder.com/stat?c=problem_statement&pm=8570&rd=12012&rm=269199&cr=7581406)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10765&rd=14183](http://www.topcoder.com/stat?c=problem_statement&pm=10765&rd=14183)
  - c. State space reduction
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10902](http://www.topcoder.com/stat?c=problem_statement&pm=10902)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=3001](http://www.topcoder.com/stat?c=problem_statement&pm=3001)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=8605&rd=12012&rm=269199&cr=7581406](http://www.topcoder.com/stat?c=problem_statement&pm=8605&rd=12012&rm=269199&cr=7581406)
  - d. Solving in the reverse - easier characterizations looking from the end
    - <http://www.spoj.pl/problems/MUSKET/>
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=5908](http://www.topcoder.com/stat?c=problem_statement&pm=5908)
  - e. Counting/optimizing arrangements satisfying some specified properties
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=8306](http://www.topcoder.com/stat?c=problem_statement&pm=8306)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=7849](http://www.topcoder.com/stat?c=problem_statement&pm=7849)
  - f. Strategies and expected values
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10765&rd=14183](http://www.topcoder.com/stat?c=problem_statement&pm=10765&rd=14183)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10806](http://www.topcoder.com/stat?c=problem_statement&pm=10806)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=7828](http://www.topcoder.com/stat?c=problem_statement&pm=7828)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=7316](http://www.topcoder.com/stat?c=problem_statement&pm=7316)
  - g. DP on probability spaces
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=7422](http://www.topcoder.com/stat?c=problem_statement&pm=7422)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=2959](http://www.topcoder.com/stat?c=problem_statement&pm=2959)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10335](http://www.topcoder.com/stat?c=problem_statement&pm=10335)
  - h. DP on trees
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10800](http://www.topcoder.com/stat?c=problem_statement&pm=10800)
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=10737](http://www.topcoder.com/stat?c=problem_statement&pm=10737)
    - [http://www.topcoder.com/stat?c=problem\\_solution&rm=266678&rd=10958&pm=8266&cr=7581406](http://www.topcoder.com/stat?c=problem_solution&rm=266678&rd=10958&pm=8266&cr=7581406)
  - i. DP with datastructures
    - <http://www.spoj.pl/problems/INCSEQ/>
    - <http://www.spoj.pl/problems/INCDSEQ/>
    - <http://www.spoj.pl/problems/LIS2/>
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=1986](http://www.topcoder.com/stat?c=problem_statement&pm=1986)
  - j. Symmetric characterization of DP state
    - [http://www.topcoder.com/stat?c=problem\\_statement&pm=8610](http://www.topcoder.com/stat?c=problem_statement&pm=8610)
  - k. A good collection of problems
    - <http://codeforces.com/blog/entry/325>
    - <http://problemclassifier.appspot.com/index.jsp?search=dp&usr=>
7. **Greedy.**
- a. Suggested Reading -
    - Chapter on Greedy algorithms in Cormen.
    - <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=greedyAlg>
  - b. problems - refer to the topcoder tutorial.
8. **Number Theory.**
- a. Modulus arithmetic - basic postulates [Including modular linear equations , Continued fraction and Pell's equation]
    - Suggested Reading -
      1. Chapter 1 from Number Theory for Computing by SY Yan [ Recommended ]
      2. 31.1, 31.3 and 31.4 from Cormen
      3. [www.topcoder.com/tc?module=Static&d1=tutorials&d2=primeNumbers](http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primeNumbers)
    - Problems
      1. <http://projecteuler.net/index.php?section=problems&id=64>
      2. <http://projecteuler.net/index.php?section=problems&id=65>
      3. <http://projecteuler.net/index.php?section=problems&id=66>
      4. [http://www.topcoder.com/stat?c=problem\\_statement&pm=6408&rd=9826](http://www.topcoder.com/stat?c=problem_statement&pm=6408&rd=9826)
      5. [http://www.topcoder.com/stat?c=problem\\_statement&pm=2342](http://www.topcoder.com/stat?c=problem_statement&pm=2342)
  - b. Fermat's theorem, Euler Totient theorem ( totient function, order , primitive roots )
    - Suggested Reading
      1. 1.6, 2.2 from Number Theory by SY Yan
      2. 31.6 , 31.7 from Cormen
    - Problems
      1. <http://projecteuler.net/index.php?section=problems&id=70>
      2. <http://www.spoj.pl/problems/NDIVPHI/>
  - c. Chinese remainder theorem
    - Suggested Reading
      1. 31.5 from Cormen
      2. 1.6 from Number Theory by SY Yan
    - Problems
      1. Project Euler 271
      2. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10551&rd=13903](http://www.topcoder.com/stat?c=problem_statement&pm=10551&rd=13903)
  - d. Primality tests -
    - Deterministic  $O(\sqrt{n})$  approach
    - Probabilistic primality tests - Fermat primality test, Miller-Rabin Primality test
      1. Suggested Reading -
        - a. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting>
        - b. Cormen 31.8
        - c. 2.2 from Number Theory by SY Yan
      2. Problems -
        - a. PON, PRIC, SOLSTRAS on SPOJ
        - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4515](http://www.topcoder.com/stat?c=problem_statement&pm=4515)
  - e. Prime generation techniques - Sieve of Erasthones
    - Suggested Problems - PRIME1 on SPOJ
  - f. GCD using euclidean method
    - Suggested Reading
      1. 31.2 Cormen
    - Problems -
      1. GCD on SPOJ
      2. <http://uva.onlinejudge.org/external/114/11424.html>
  - g. Logarithmic Exponentiation
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting>
  - h. Integer Factorization
    - Naive  $O(\sqrt{n})$  method
    - Pollard Rho factorization
    - Suggested Reading
      1. 2.3 from Number Theory SY Yan
      2. 31.9 Cormen
    - Problems -
      1. [http://www.topcoder.com/stat?c=problem\\_statement&pm=2986&rd=5862](http://www.topcoder.com/stat?c=problem_statement&pm=2986&rd=5862)
      2. <http://www.spoj.pl/problems/DIVSUM2/>
      3. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4481&rd=6538](http://www.topcoder.com/stat?c=problem_statement&pm=4481&rd=6538)
  - i. Stirling numbers
  - j. Wilson theorem
    - $nCr \% p$  in  $O(p)$  preprocess and  $O(\log n)$  query
  - k. Lucas Theorem
  - l. Suggested Reading for Number Theory -
    - Number theory for computing by Song Y Yan [ Simple book describing concepts in details ]
    - Concepts are also superficially covered in Chapter 31 of Introduction to Algorithms by Cormen
    - <http://www.codechef.com/wiki/tutorial-number-theory>

- [http://www.algorithmist.com/index.php/Category:Number\\_Theory](http://www.algorithmist.com/index.php/Category:Number_Theory)
- m. Problems on Number Theory -
  - [http://www.algorithmist.com/index.php/Category:Number\\_Theory](http://www.algorithmist.com/index.php/Category:Number_Theory)
  - <http://problemclassifier.appspot.com/index.jsp?search=number&usr=>
- 9. **Math (Probability, Counting, Game Theory, Group Theory, Generating functions, Permutation Cycles, Linear Algebra)**
  - a. **Probability.**
    - Syllabus
      - Basic probability and Conditional probability
        1. Suggested problems
          - a. <http://www.spoj.pl/problems/CT16E/>
          - b. <http://www.spoj.pl/problems/CHICAGO/>
        - Random variables, probability generating functions
        - Mathematical expectation + Linearity of expectation
          1. Suggested problems
            - a. <http://www.spoj.pl/problems/FAVDICE/>
            - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10744](http://www.topcoder.com/stat?c=problem_statement&pm=10744)
          - Special discrete and continuous probability distributions
            1. Bernoulli, Binomial, Poisson, normal distribution
            2. Suggested Problem
              - a. <http://acm.sgu.ru/problem.php?contest=0&problem=498>
            - Suggested Readings
              1. Cormen appendix C (very basic)
              2. Topcoder probabiltiy tutorial <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=probabilities>
              3. [http://en.wikipedia.org/wiki/Random\\_variable](http://en.wikipedia.org/wiki/Random_variable)
              4. [http://en.wikipedia.org/wiki/Expected\\_value](http://en.wikipedia.org/wiki/Expected_value)
              5. William Feller, An introduction to probability theory and its applications
  - b. **Counting**
    - Syllabus
      - Basic principles - Pigeon hole principle, addition, multiplication rules
        1. Suggested problems
          - a. <http://acm.timus.ru/problem.aspx?space=1&num=1690>
          - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10805](http://www.topcoder.com/stat?c=problem_statement&pm=10805)
        3. Suggested readings
          - a. [http://en.wikipedia.org/wiki/Combinatorial\\_principles](http://en.wikipedia.org/wiki/Combinatorial_principles)
          - b. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=combinatorics>
          - c. <http://www.maa.org/editorial/knot/pigeonhole.html>
        - Inclusion-exclusion
          1. Suggested readings
            - a. [http://en.wikipedia.org/wiki/Inclusion-exclusion\\_principle](http://en.wikipedia.org/wiki/Inclusion-exclusion_principle)
          2. Suggested problems
            - a. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4463&rd=6536](http://www.topcoder.com/stat?c=problem_statement&pm=4463&rd=6536)
            - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10238](http://www.topcoder.com/stat?c=problem_statement&pm=10238)
          - Special numbers
            1. Suggested reading - Stirling, eurlerian, harmonic, bernoulli, fibonnacci numbers
              - a. [http://en.wikipedia.org/wiki/Stirling\\_number](http://en.wikipedia.org/wiki/Stirling_number)
              - b. [http://en.wikipedia.org/wiki/Eulerian\\_numbers](http://en.wikipedia.org/wiki/Eulerian_numbers)
              - c. [http://en.wikipedia.org/wiki/Harmonic\\_series\\_\(mathematics\)](http://en.wikipedia.org/wiki/Harmonic_series_(mathematics))
              - d. [http://en.wikipedia.org/wiki/Bernoulli\\_number](http://en.wikipedia.org/wiki/Bernoulli_number)
              - e. [http://en.wikipedia.org/wiki/Fibonnaci\\_numbers](http://en.wikipedia.org/wiki/Fibonnaci_numbers)
              - f. Concrete mathematics by Knuth
            2. Suggested problems
              - a. [http://www.topcoder.com/stat?c=problem\\_statement&pm=1643](http://www.topcoder.com/stat?c=problem_statement&pm=1643)
              - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8202&rd=11125](http://www.topcoder.com/stat?c=problem_statement&pm=8202&rd=11125)
              - c. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8725](http://www.topcoder.com/stat?c=problem_statement&pm=8725)
              - d. [http://www.topcoder.com/stat?c=problem\\_statement&pm=2292&rd=10709](http://www.topcoder.com/stat?c=problem_statement&pm=2292&rd=10709)
            - Advanced counting techniques - Polya counting, burnsides lemma
              1. Suggested reading
                - a. [http://en.wikipedia.org/wiki/Burnside's\\_lemma](http://en.wikipedia.org/wiki/Burnside's_lemma)
                - b. <http://petr-mitrichev.blogspot.com/2008/11/burnsides-lemma.html>
              2. Suggested Problems
                - a. [http://www.topcoder.com/stat?c=problem\\_statement&pm=9975](http://www.topcoder.com/stat?c=problem_statement&pm=9975)
                - b. <http://www.spoj.pl/problems/TRANSP/>
    - c. **Game theory**
      - Syllabus
        - Basic principles and Nim game
          1. Sprague grundy theorem, Grundy numbers
          2. Suggested readings
            - a. [http://en.wikipedia.org/wiki/Sprague%E2%80%93Grundy\\_theorem](http://en.wikipedia.org/wiki/Sprague%E2%80%93Grundy_theorem)
            - b. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=algorithmGames>
            - c. <http://www.ams.org/samplings/feature-column/fcarc-games1>
            - d. <http://www.codechef.com/wiki/tutorial-game-theory>
          3. Suggested problems
            - a. [http://www.topcoder.com/stat?c=problem\\_statement&pm=3491&rd=6517](http://www.topcoder.com/stat?c=problem_statement&pm=3491&rd=6517)
            - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=3491&rd=6517](http://www.topcoder.com/stat?c=problem_statement&pm=3491&rd=6517)
          - Hackenbush
            1. Suggested readings
              - a. <http://en.wikipedia.org/wiki/Hackenbush>
              - b. <http://www.ams.org/samplings/feature-column/fcarc-partizan1>
            2. Suggested problems
              - a. <http://www.cs.caltech.edu/ipsc/problems/g.html>
              - b. <http://www.spoj.pl/problems/PT07A/>
      - d. **Linear Algebra**
        - Syllabus
          - Matrix Operations
            1. Addition and subtraction of matrices
              - a. Suggested Reading
                - i. Cormen 28.1
            2. Multiplication ( Strassen's algorithm ), logarithmic exponentiation
              - a. Suggested reading
                - i. Cormen 28.2
                - ii. Linear Algebra by Kenneth Hoffman Section 1.6
              - b. Problems
                - i. <http://uva.onlinejudge.org/external/111/11149.html>
            3. Matrix transformations [ Transpose, Rotation of Matrix, Representing Linear transformations using matrix ]
              - a. Suggested Reading
                - i. Linear Algebra By Kenneth Hoffman Section 3.1,3.2,3.4,3.7
              - b. Problems
                - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=6877](http://www.topcoder.com/stat?c=problem_statement&pm=6877)
                - ii. JPIX on Spoj
            4. Determinant , Rank and Inverse of Matrix [ Gaussean Elimination , Gauss Jordan Elimination]
              - a. Suggested Reading
                - i. 28.4 Cormen
                - ii. Linear Algebra by Kenneth Chapter 1
              - b. Problems
                - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8174](http://www.topcoder.com/stat?c=problem_statement&pm=8174)
                - ii. [http://www.topcoder.com/stat?c=problem\\_statement&pm=6407&rd=9986](http://www.topcoder.com/stat?c=problem_statement&pm=6407&rd=9986)

- iii. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8587](http://www.topcoder.com/stat?c=problem_statement&pm=8587)
    - iv. HIGH on Spoj
  - 5. Solving system of linear equations
    - a. Suggested Reading
      - i. 28.3 Cormen
      - ii. Linear Algebra by Kenneth Chapter 1
    - b. Problems -
      - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=3942&rd=6520](http://www.topcoder.com/stat?c=problem_statement&pm=3942&rd=6520)
  - 6. Using matrix exponentiation to solve recurrences
    - a. Suggested Reading
      - i. <http://www.topcoder.com/tc?module=Static&d1=features&d2=010408>
    - b. Problems
      - i. REC, RABBIT1 , PLHOP on spoj
      - ii. [http://www.topcoder.com/stat?c=problem\\_statement&pm=6386](http://www.topcoder.com/stat?c=problem_statement&pm=6386) , [http://www.topcoder.com/stat?c=problem\\_statement&pm=7262](http://www.topcoder.com/stat?c=problem_statement&pm=7262) , [http://www.topcoder.com/stat?c=problem\\_statement&pm=6877](http://www.topcoder.com/stat?c=problem_statement&pm=6877)
  - 7. Eigen values and Eigen vectors
    - a. Problems
      - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=2423&rd=4780](http://www.topcoder.com/stat?c=problem_statement&pm=2423&rd=4780)
- Polynomials
  - 1. Roots of a polynomial [ Prime factorization of a polynomial, Integer roots of a polynomial, All real roots of a polynomial ]
    - a. Problems
      - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8273&rd=10798](http://www.topcoder.com/stat?c=problem_statement&pm=8273&rd=10798)
      - ii. POLYEQ , ROOTCIPH on Spoj
  - 2. Lagrange Interpolation
    - a. Problems
      - i. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10239](http://www.topcoder.com/stat?c=problem_statement&pm=10239)
      - ii. [http://www.topcoder.com/stat?c=problem\\_statement&pm=8725](http://www.topcoder.com/stat?c=problem_statement&pm=8725)
- e. Permutation cycles
  - Suggested Reading
    - 1. Art of Computer Programming by Knuth Vol. 3
  - Problems
    - 1. ShuffleMethod, Permutation and WordGame on topcoder.
- f. Group Theory
  - Burnside Lemma, Polias theorem
    - 1. Suggested Reading
      - a. Hernstein's topics in algebra
      - b. <http://petr-mitrichev.blogspot.com/2008/11/burnsides-lemma.html>
    - 2. Problems
      - a. TRANSP on spoj
      - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=9975](http://www.topcoder.com/stat?c=problem_statement&pm=9975)
- b. Generating functions
  - Suggested Reading
    - 1. Herbert Wilf's generating functionology
    - 2. Robert Sedgewick and Flajolet's Combinatorial analysis

## 10. Data Structures.

### i. Basic

- a. Arrays/Stacks/Queues :
  - Problems
    - 1. <https://www.spoj.pl/problems/STPAR/>
    - 2. <https://www.spoj.pl/problems/SHOP/>
    - 3. <https://www.spoj.pl/problems/WATER/>
  - Reading:
    - 1. CLRS: section 10.1
    - 2. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=dataStructures>
- b. Singly/Doubly Linked List :
  - Problems
    - 1. <https://www.spoj.pl/problems/POSTERS/>
  - Reading: CLRS: section 10.2, Mark Allen Weies Chapter 3
- c. Hash Tables :
  - Problems
    - 1. <https://www.spoj.pl/problems/HASHIT/>
    - 2. <https://www.spoj.pl/problems/CUCKOO/>
  - Reading: CLRS: Chapter 11, Mark Allen Weies Chapter 5
- d. Circular linked list / queue
  - Problems
    - 1. <https://www.spoj.pl/problems/CTRICK/>
- e. Binary/nary Trees
  - Reading
    - 1. CLRS: section 10.4
    - 2. CLRS: Chapter 12
    - 3. Mark Allen Weies Chapter 4
    - 4. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binarySearchRedBlack>
- f. Heaps
  - Problems
    - 1. <https://www.spoj.pl/problems/PRO/>
    - 2. <https://www.spoj.pl/problems/EXPEDI/>
  - Reading : Mark Allen Weies Chapter 6

### ii. Advanced

- a. Trie (Keyword tree)
  - Problems
    - 1. <https://www.spoj.pl/problems/MORSE/>
    - 2. <https://www.spoj.pl/problems/EMOTICON/>
  - Reading
- b. Interval trees / Segment Trees
  - Problems
    - 1. <https://www.spoj.pl/problems/ORDERS/>
    - 2. <https://www.spoj.pl/problems/FREQUENT/>
  - Reading
- c. Fenwick(Binary Indexed) trees
  - Problems
    - 1. <https://www.spoj.pl/problems/MATSUM/>
  - Reading: <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees>
- d. Disjoint data structures
  - Problems
    - 1. <https://www.spoj.pl/problems/BLINNET/>
    - 2. <https://www.spoj.pl/problems/CHAIN/>
  - Reading:
    - 1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=disjointDataStructure>
    - 2. Mark Allen Weies Chapter 8
- e. Range minimum Query (RMQ)
  - Problems
    - 1. <https://www.spoj.pl/problems/GSS1/>
  - Reading <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lowestCommonAncestor>
- f. Customized interval/segment trees (Augmented DS)
  - Problems
    - 1. <https://www.spoj.pl/problems/GSS3/>
    - 2. <https://www.spoj.pl/problems/RRSCHED/>
  - Reading: CLRS: Chapter 14 (augmented DS)

- g. AVL Trees
  - Problems
    1. <https://www.spoj.pl/problems/ORDERS/>
  - Reading
- iii. Miscellaneous (Not to be covered)
  - a. Splay Trees
  - b. B/B+ Trees
  - c. k-d Trees
  - d. Red-black Trees
  - e. Skip List
  - f. Binomial/ Fibonacci heaps
- iv. Exercises
  1. <https://www.spoj.pl/problems/LAZYPROG/> (Hint: Heaps)t
  2. <https://www.spoj.pl/problems/HELPR2D2/> (Hint: Interval Trees)
  3. <https://www.spoj.pl/problems/SAM/> (Hint: Heaps)
  4. <https://www.spoj.pl/problems/PRHYME/> (Hint: Trie)
  5. <https://www.spoj.pl/problems/HEAPULM/> (Hint: Interval Trees)
  6. <https://www.spoj.pl/problems/CORNET/> (Hint: Disjoint )
  7. <https://www.spoj.pl/problems/EXPAND/>
  8. <https://www.spoj.pl/problems/WPUZZLES/>
  9. <https://www.spoj.pl/problems/LIS2/>
- 11. Search Techniques/Bruteforce writing techniques/Randomized algorithms.
  - a. Backtracking - [Beginner].
    - problems ->
      1. N queens problems
      2. Knights Tour
      3. Sudoku Problem
      4. Tiling Problem.
      5. 15 puzzle.
  - b. Dancing Links and Algorithm X given by Knuth - [Advanced]
    - problems - PRLGAME, SUDOKU, NQUEEN on SPOJ
    - Suggested reading -
      1. <http://www-cs-faculty.stanford.edu/~uno/papers/dancing-color.ps.gz>
  - c. Binary Search - [Beginner].
    - problems - AGGRCOW on SPOJ. Refer the tutorial for more problems.
    - finding all real roots of a polynomial using binary search. [intermediate].
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binarySearch>
  - d. Ternary Search - [Intermediate].
    - problems -
      1. <http://www.spoj.pl/problems/KPPOLY/>
      2. <http://www.codechef.com/DEC09/problems/K1/>
      3. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4705&rd=7993](http://www.topcoder.com/stat?c=problem_statement&pm=4705&rd=7993)
      4. [http://www.topcoder.com/stat?c=problem\\_statement&pm=7741&rd=10671](http://www.topcoder.com/stat?c=problem_statement&pm=7741&rd=10671)
      5. [http://www.topcoder.com/stat?c=problem\\_statement&pm=6464&rd=9994](http://www.topcoder.com/stat?c=problem_statement&pm=6464&rd=9994)
      6. [http://www.topcoder.com/stat?c=problem\\_statement&pm=3501&rd=6529](http://www.topcoder.com/stat?c=problem_statement&pm=3501&rd=6529)
      7. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4567&rd=6539](http://www.topcoder.com/stat?c=problem_statement&pm=4567&rd=6539)
  - e. Meet in the middle [Intermediate].
    - problems -
      1. <http://www.spoj.pl/problems/MAXISET/>
      2. <http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=2868>
  - f. Hill Climbing [Advanced].
  - g. Regular Iteration to reach a fixed point [Advanced].
    - Newton-Raphson method to find root of a mathematical function.
    - Iterations to solve linear non-homogeneous system of equations.
  - h. Randomized Algorithms [Intermediate]-
    - Quick-Sort.
- 12. General programming issues in contests ->
  - a. Arithmetic Precision - [Beginner].
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=integersReals>
  - b. Representing sets with bitmasks and manipulating bitmasks - [Beginner].
    - Suggested Reading -
      1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=bitManipulation>
    - problems - refer to the tutorial link in Suggested reading section.