List of Topics for programming Competitions -

1. **Basic Geometry/Euclidean Geometry/Coordinate Geometry/ [3-D variants of everything].**
2. **Computational Geometry.**
   1. Graham Scan algorithm for Convex Hull O(n \* log(n)).
   2. Online construction of 3-D convex hull in O(n^2).
   3. Bentley Ottmann algorithm to list all intersection points of n line segments in O((n + I) \* logn).
      * Suggested Reading -
        1. [http://softsurfer.com/Archive/algorithm\_0108/algorithm\_0108.htm](http://www.google.com/url?q=http%3A%2F%2Fsoftsurfer.com%2FArchive%2Falgorithm_0108%2Falgorithm_0108.htm&sa=D&sntz=1&usg=AFQjCNGHQhcxX4Fmwl4SNxT_aka8mbXabQ)
   4. Rotating Calipers Technique.
      * Suggested Reading - [http://cgm.cs.mcgill.ca/~orm/rotcal.html](http://www.google.com/url?q=http%3A%2F%2Fcgm.cs.mcgill.ca%2F~orm%2Frotcal.html&sa=D&sntz=1&usg=AFQjCNEpr1ZpsZJ56mVXjfaYMG2f2sCbCA)
      * Problems - Refer the article for a list of problems which can be solved using Rotating Calipers technique.
   5. 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](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DlineSweep&sa=D&sntz=1&usg=AFQjCNHfGJkCXE6CJ6cW8GrhVB1sRa18nA)
      * Problems - Follow the tutorial for list of problems.
   6. Area of Union of Circles.
   7. Delayunay Triangulation of n points in O(n \* logn).
   8. Voronoi Diagrams of n points in O(n \* logn) using Fortunes algorithm.
   9. Point in a polygon problem -
      * O(n) solution without preprocessing.
      * O(logn) algorithm with O(n \* logn) preprocessing for convex polygons.
   10. Problems on computational geometry -
       * [BSHEEP](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBSHEEP&sa=D&sntz=1&usg=AFQjCNHrQnQIXAHii4tur4t2muJJ7pzzIQ), [BULK](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBULK&sa=D&sntz=1&usg=AFQjCNGkGqPc5jyvbR7umTXGAzOkyrV04Q), [SEGVIS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSEGVIS&sa=D&sntz=1&usg=AFQjCNHDL6-G9gyo1nQq4r_XBmOx3BdDWw), [CONDUIT](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCONDUIT&sa=D&sntz=1&usg=AFQjCNE_T_3ZvKY12Cj_wWBMJHES59QvxQ), [RUNAWAY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRUNAWAY&sa=D&sntz=1&usg=AFQjCNGUJdoxwo00a81YJmgEQv2c9tKAmA), [DIRVS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FDIRVS&sa=D&sntz=1&usg=AFQjCNGLi0ndUIgEia_o767dVma82r3FCg), [RAIN1](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRAIN1&sa=D&sntz=1&usg=AFQjCNHlj9N3r6qaVHAQn7yd9C_MSYsDzQ), [SHAMAN](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSHAMAN&sa=D&sntz=1&usg=AFQjCNFfHJa2psJr09sW-TsMvHnV8NoJyg), [TCUTTER](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FTCUTTER&sa=D&sntz=1&usg=AFQjCNGkc7R1QswuEHQDfSbgm1CN_EuouA), [LITEPIPE](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLITEPIPE&sa=D&sntz=1&usg=AFQjCNG5Gr3w6r78MNgTq7C3Z8-Q7A1Vag), [RHOMBS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRHOMBS&sa=D&sntz=1&usg=AFQjCNHFV81La4xqxYB8xtqDZro8GPc7yg), [FSHEEP](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FFSHEEP&sa=D&sntz=1&usg=AFQjCNGGKcTJ2DoyhrRJoHP9H2fgwgJ2ow), [FLBRKLIN](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FFLBRKLIN&sa=D&sntz=1&usg=AFQjCNGt2YjKpjnfXjs2zuocFIi9GsayBg), [CERC07P](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCERC07P&sa=D&sntz=1&usg=AFQjCNFPzmHxOyMDyEA8usNeWT8NDgGWpQ), [BAC](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBAC&sa=D&sntz=1&usg=AFQjCNF3Q6F6POk-t7n4MqmfIGAZqOwkNA), [ALTARS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FALTARS&sa=D&sntz=1&usg=AFQjCNH_qtGg91a3u6FSVb4uY1zD7UXFKA), [CERC07C](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCERC07C&sa=D&sntz=1&usg=AFQjCNFKDRB_j5Tn4EbPpy4z9rhTROKe8w), [NECKLACE](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FNECKLACE&sa=D&sntz=1&usg=AFQjCNFtVet7fKaSrNYEB40ZwPcZ2jtZ3w), [CH3D](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCH3D&sa=D&sntz=1&usg=AFQjCNFxd1wPbbMpMCY1ufqoBATQ3mcwDA), [RECTANGL](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRECTANGL&sa=D&sntz=1&usg=AFQjCNHOvAwG0bLHV39ptONdr9jjsG4xlQ), [POLYSSQ](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPOLYSSQ&sa=D&sntz=1&usg=AFQjCNHCsZRqf52QIsalJ6JnyBiEnvnFWA), [FOREST2](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FFOREST2&sa=D&sntz=1&usg=AFQjCNEyyyTMF69-Fc_HSEMoc3xTku42gA), [KPPOLY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FKPPOLY&sa=D&sntz=1&usg=AFQjCNHvsQMxoQ1UKJEUIwZ453CWCe6u-A), [RAIN2](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRAIN2&sa=D&sntz=1&usg=AFQjCNGmC6HqrcPL9yywlwZjWCQ-a2A-KA), [SEGMENTS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSEGMENTS&sa=D&sntz=1&usg=AFQjCNHilM8lQ6O66NjaVUHiKzzX8d4Lqw), [ARCHPLG](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FARCHPLG&sa=D&sntz=1&usg=AFQjCNHxZxWl6kTRlLwSm5f-hRVhBaoXUw), [BALLOON](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBALLOON&sa=D&sntz=1&usg=AFQjCNFJ2JC9vsd6-cgcKyvkSAIyTnAHyg), [CIRCLES](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCIRCLES&sa=D&sntz=1&usg=AFQjCNFBjOye5vwvHjFQZcU6pgm1CB4_4Q), [COMPASS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCOMPASS&sa=D&sntz=1&usg=AFQjCNHxm9AZPj8aQYHlsQmoumsQmQeBZA), [EOWAMRT](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEOWAMRT&sa=D&sntz=1&usg=AFQjCNH4RUwwKEfP3vpekPDoq20y2476SA), [ICERINK](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FICERINK&sa=D&sntz=1&usg=AFQjCNFF6PcrYWaK17Tt82GYwmPbqm1wCQ) on SPOJ.
       * [CultureGrowth](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D3996&sa=D&sntz=1&usg=AFQjCNH8N6avfA-fum614aB6dS5nlKuSMA), [PolygonCover](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8540&sa=D&sntz=1&usg=AFQjCNF7chzZhWrZMOyIjnfQTRX6vYE_yw) on Topcoder.
   11. Suggested Reading -
       * Computational Geometry: Algorithms and applications. Mark De Burg.
3. **String Algorithm**.
   1. 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](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DstringSearching&sa=D&sntz=1&usg=AFQjCNEUDbDJrsbEFUC1npCLKZY8hQSmiw)
   2. Aho Corasick algorithm.
      * Problems - WPUZZLES on SPOJ.
   3. Suffix Arrays
      * O(n^2 \* logn) Naive method of suffix array construction
      * O(n \* logn^2) method of suffix array construction
      * O(n \* logn) 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.
   4. 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.
   5. Suffix Automata
      * O(n) Suffix Automaton construction.
   6. Dictionary Of Basic Factors
      * O(n \* logn) method of DBF construction using Radix Sort.
   7. 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).
   8. Searching and preprocessing Regular Expressions consisting of ‘?’, ‘\*’.
   9. Multi-dimensional pattern matching.
   10. Problems on Strings [can be solved with a variety of techniques] -
       * [DISUBSTR](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FDISUBSTR&sa=D&sntz=1&usg=AFQjCNF2zSUg43c6hOCimrvccGWAVmd-Cw), [PLD](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPLD%2F&sa=D&sntz=1&usg=AFQjCNFfz9ttNKXV8w84VzWz6eij-XqRRA), [MSTRING](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMSTRING%2F&sa=D&sntz=1&usg=AFQjCNEzAlLDs8jFleiNQc8q-HeAN2iApw), [REPEATS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FREPEATS&sa=D&sntz=1&usg=AFQjCNFoItgwm_QmdLBzis2YzdVY06EDRA), [JEWELS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FJEWELS&sa=D&sntz=1&usg=AFQjCNHTJMeS-G9NT83okcxQxwXmOx32Bg), [ARCHIVER](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FARCHIVER&sa=D&sntz=1&usg=AFQjCNGf6rM5QbNL1A4esNBft6Xst-Cfvg), [PROPKEY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPROPKEY&sa=D&sntz=1&usg=AFQjCNFpn4jUrSeh901pvsbNwsVA8qtbPw), [LITELANG](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLITELANG&sa=D&sntz=1&usg=AFQjCNFgGk_4kDcCozOvqWXpUQZkmQG7rg), [EMOTICON](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEMOTICON&sa=D&sntz=1&usg=AFQjCNHxocwcdFLVEW_QvSJLTPqfxyLJmA), [WORDS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FWORDS&sa=D&sntz=1&usg=AFQjCNGPHsNtnjtfj-UOClSoNXkYXLr94g), [AMCODES](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FAMCODES&sa=D&sntz=1&usg=AFQjCNGQMruBH6mCdxcms6N3ubiqAa2bow), [UCODES](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FUCODES&sa=D&sntz=1&usg=AFQjCNFg1vsMeCg9bTP4p4nN-wLXTZ56lQ), [PT07H](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPT07H&sa=D&sntz=1&usg=AFQjCNFuNhCDGzp_ZGala89I94B62UM6jw), [MINSEQ](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMINSEQ&sa=D&sntz=1&usg=AFQjCNF8sDbNAXKP1eOpYubgI1elKrqRcw), [TOPALIN](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FTOPALIN&sa=D&sntz=1&usg=AFQjCNGuxL4DfD0j0Y2FmRTyTXJHDQsFMA), [BWHEELER](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBWHEELER&sa=D&sntz=1&usg=AFQjCNG4fgKPlo2H_TliBDlpaeTpQhE1mA), [BEADS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBEADS&sa=D&sntz=1&usg=AFQjCNFbfscdITKZDWoJMg6_SBZcD-mJnA), [SARRAY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSARRAY&sa=D&sntz=1&usg=AFQjCNEguNFEpH5__hdkijmgxSfpm965AA), [LCS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLCS&sa=D&sntz=1&usg=AFQjCNHBYacxaQzh3iVXzE5GyzrRShvRFQ), [LCS2](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLCS2&sa=D&sntz=1&usg=AFQjCNHAxNQd_9fxNWWSH_LhNfRyUMvrgA), [SUBST1](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSUBST1&sa=D&sntz=1&usg=AFQjCNF8fqDFATiXqd8kL0niOle_sG2okg), [PHRASES](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPHRASES&sa=D&sntz=1&usg=AFQjCNEpbdieZqSnBUyAAR48fvW3014ySQ), [PRETILE](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPRETILE&sa=D&sntz=1&usg=AFQjCNFn_-8ijgscKShQV0BX7qklAAnPBw) on SPOJ
       * [http://www.algorithmist.com/index.php/Category:String\_algorithms](http://www.google.com/url?q=http%3A%2F%2Fwww.algorithmist.com%2Findex.php%2FCategory%3AString_algorithms&sa=D&sntz=1&usg=AFQjCNGFY_b7--kIqVTfs6-m7jq5lRnqNg)
4. **Basic Graphs [beginner]**.
   1. Representation of graphs as adjacency list, adjacency matrix, incidence matrix and edge list and uses of different representations in different scenarios.
   2. Breadth First Search.
      * problems -
        1. [PPATH](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPPATH&sa=D&sntz=1&usg=AFQjCNHP0qJmzvPZxTIGNF3vPf2ZBdjJZQ), [ONEZERO](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FONEZERO&sa=D&sntz=1&usg=AFQjCNGUicduZ0f8yKCYxdHzCeT9zjBHEA), [WATER](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FWATER&sa=D&sntz=1&usg=AFQjCNG50loi0Rzurv8c2S83J2NSDS7JYw) on SPOJ
   3. Depth First Search.
   4. Strongly Connected Components.
      * problems -
        1. [TOUR](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FTOUR&sa=D&sntz=1&usg=AFQjCNHRtiTBx0VovaOXm2vNknzzqkhEqw) and [BOTTOM](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBOTTOM&sa=D&sntz=1&usg=AFQjCNF5ZxUZlDlT7AY_eU_nJfbiJYZDzQ) on SPOJ.
   5. Biconnected Components, Finding articulation points and bridges].
      * problems -
        1. [RELINETS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRELINETS&sa=D&sntz=1&usg=AFQjCNEkbPg45j_z2Adi3bJf72erl78IEg), [PT07A](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPT07A&sa=D&sntz=1&usg=AFQjCNEkmY9z8JVJa2cAllxqB2FtghjBgQ) on SPOJ.
   6. Dijkstra algorithm -
      * problems -
        1. [SHPATH](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSHPATH&sa=D&sntz=1&usg=AFQjCNHb4sPmATfcukKWAPyqGJBssomOVw) on SPOJ.
   7. Floyd Warshall algorithm -
      * problems -
        1. [COURIER](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCOURIER&sa=D&sntz=1&usg=AFQjCNEo7NyYAr9cfd9mDfqKmhSPQYadhQ) on SPOJ.
   8. Minimum Spanning Tree
      * problems -
        1. [BLINNET](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBLINNET&sa=D&sntz=1&usg=AFQjCNHgUy75fEIsFJMINnI5zf_dHjQU-g) on SPOJ.
   9. Flood-fill algorithm
   10. Topological sort
   11. Bellman-Ford algorithm.
   12. Euler Tour/Path.
       * problems - [WORDS1](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FWORDS1&sa=D&sntz=1&usg=AFQjCNHjoji068MUzVw96m6ehHLNEj1eNw) on SPOJ.
   13. Suggested reading for most of the topics in Graph algorithms -
       * [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=graphsDataStrucs1](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DgraphsDataStrucs1&sa=D&sntz=1&usg=AFQjCNHhxFY2EvUsl9A1Gqx19hmJ5TkpuA).
       * Also refer to the tutorial for problems concerning these techniques.
       * Cormen chapter 22 to 24.
5. **Flow networks/ matching etc etc. [Interdiate/Advanced].**
   1. Maximum flow using Ford Fulkerson Method.
      * Suggested Reading -
        1. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=maxFlow](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DmaxFlow&sa=D&sntz=1&usg=AFQjCNGUMYitrdXxKZckgZE4sJXCJ-jxtA)
      * problems - [TAXI](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FTAXI&sa=D&sntz=1&usg=AFQjCNFE_58rty8tgbu3R6iLBtjVoIxz1w), [POTHOLE](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPOTHOLE&sa=D&sntz=1&usg=AFQjCNFC2q3Akp93D69doENSHYEmgSpEvw), [IM](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FIM&sa=D&sntz=1&usg=AFQjCNEkFB-VzlWD-i6_PO3EGJDJzHNkpg), [QUEST4](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FQUEST4&sa=D&sntz=1&usg=AFQjCNGzZ5zMMYZSMrmDbQ9PuGtYonR8Ww), [MUDDY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMUDDY&sa=D&sntz=1&usg=AFQjCNEC2nE8e8-mWwkMg8gAU-T0Ih7u2Q), [EN](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEN&sa=D&sntz=1&usg=AFQjCNFJfLrA3-r2p5qxaHl3dBuCcBCCbg), [CABLETV](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCABLETV&sa=D&sntz=1&usg=AFQjCNE7V8p-tbwjPrljrZ38fS8XiBtpaQ), [STEAD](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSTEAD&sa=D&sntz=1&usg=AFQjCNHXDgaBjLgZHtOuaw-InCfSdFJ1NQ), [NETADMIN](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FNETADMIN&sa=D&sntz=1&usg=AFQjCNGe5UlGM1ztLbrbZmpADYqw2oSCsw), [COCONUTS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCOCONUTS&sa=D&sntz=1&usg=AFQjCNHXoR3VdVqlIye54f38rRpwP7PLfw), [OPTM](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FOPTM&sa=D&sntz=1&usg=AFQjCNEgWLeZ0gY6PeGn_MEBGeGiUac9Rg) on SPOJ.
   2. Maximum flow using Dinics Algorithm.
      * Problems - [PROFIT](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPROFIT&sa=D&sntz=1&usg=AFQjCNHyOw5mNI4xzvu9pZOKzGfC7qAoMQ) on spoj.
   3. 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](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DminimumCostFlow1&sa=D&sntz=1&usg=AFQjCNFuwERaIcVE0TtQ8bbNA4pqoZv2dQ)
   4. Maximum weighted Bipartite Matching (Kuhn Munkras algorithm/Hungarian Method)
      * problems - [GREED](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FGREED&sa=D&sntz=1&usg=AFQjCNHXybhQL7F8NfuP_kzpoYtNG0DCBQ), [SCITIES](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSCITIES&sa=D&sntz=1&usg=AFQjCNG_uP-_WiPYVnyGNSkeMl3hNfSA9w), [TOURS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FTOURS&sa=D&sntz=1&usg=AFQjCNF6CHN7u6K1LqM0ktfo8pryexC15w) on SPOJ | [http://www.topcoder.com/stat?c=problem\_statement&pm=8143](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8143&sa=D&sntz=1&usg=AFQjCNE_393R5xBW2oYOLP1hLcMks4jdjg)
   5. Stoer Wagner min-cut algorithm.
   6. Hopcroft Karp bipartite matching algorithm.
      * problems - [ANGELS](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FANGELS&sa=D&sntz=1&usg=AFQjCNHMf1-oALn89HPVkxMG5O0CKUIQ7A) on SPOJ.
   7. Maximum matching in general graph (blossom shrinking)
   8. Gomory-Hu Trees.
      * i) Problems - [MCQUERY](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMCQUERY&sa=D&sntz=1&usg=AFQjCNF_tjThifTqIDILdIHy2U3C3m13ZQ) on Spoj.
   9. Chinese Postman Problem.
      * problems - [http://acm.uva.es/archive/nuevoportal/data/problem.php?p=4039](http://www.google.com/url?q=http%3A%2F%2Facm.uva.es%2Farchive%2Fnuevoportal%2Fdata%2Fproblem.php%3Fp%3D4039&sa=D&sntz=1&usg=AFQjCNHJ4Xj3pKU6hUYrhTo4BZDOx2vXbw)
      * Suggested Reading - [http://eie507.eie.polyu.edu.hk/ss-submission/B7a/](http://www.google.com/url?q=http%3A%2F%2Feie507.eie.polyu.edu.hk%2Fss-submission%2FB7a%2F&sa=D&sntz=1&usg=AFQjCNHL6JrQQFsmSMMmYCzJBBoZmn6p8g)
   10. Suggested Reading for the full category ->
       * Network flow - Algorithms and Applications by Ahuja
       * Cormen book chapter 25.
6. **Dynamic Programming.**
   1. Suggested Reading - Dynamic Programming(DP) as a tabulation method
      * Cormen chapter on DP
   2. 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.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8570%26rd%3D12012%26rm%3D269199%26cr%3D7581406&sa=D&sntz=1&usg=AFQjCNH6LU_oZhstLxSG1dFDPMKKvF62ow)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10765&rd=14183](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10765%26rd%3D14183&sa=D&sntz=1&usg=AFQjCNGoZPc3el8fajwfVHElC4k0BQKq9Q)
   3. State space reduction
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10902](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10902&sa=D&sntz=1&usg=AFQjCNHudTwe4XFfEyK-96U4a_O7nKLMhA)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=3001](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D3001&sa=D&sntz=1&usg=AFQjCNEpsb5Vwx0MNfQc3WjEXKS6WFjc2A)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=8605&rd=12012&rm=269199&cr=7581406](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8605%26rd%3D12012%26rm%3D269199%26cr%3D7581406&sa=D&sntz=1&usg=AFQjCNGAXkdJpl1cHEYCVlIowk9g7emXqA)
   4. Solving in the reverse - easier characterizations looking from the end
      * [http://www.spoj.pl/problems/MUSKET/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMUSKET%2F&sa=D&sntz=1&usg=AFQjCNF4iRignJt3u3dXwCtBBm3n-MZm1A)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=5908](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D5908&sa=D&sntz=1&usg=AFQjCNE9tKju_hgYLduE7ks8SWlI13gjug)
   5. Counting/optimizing arrangements satisfying some specified properties
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=8306](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8306&sa=D&sntz=1&usg=AFQjCNHY2UTBUjAla1OQCAPWRsM38OHScQ)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=7849](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D7849&sa=D&sntz=1&usg=AFQjCNFDlsbzqjJiVnW_fN4A9gNxD3aZrw)
   6. Strategies and expected values
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10765&rd=14183](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10765%26rd%3D14183&sa=D&sntz=1&usg=AFQjCNGoZPc3el8fajwfVHElC4k0BQKq9Q)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10806](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10806&sa=D&sntz=1&usg=AFQjCNEAxqdU5alwrHJ1gMZFYokUsgwQbQ)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=7828](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D7828&sa=D&sntz=1&usg=AFQjCNHDp1p6xNyke9OijIdE8txgAxZ1oA)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=7316](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D7316&sa=D&sntz=1&usg=AFQjCNFsOD7bKUFWBOjc3H6B6Yyr7uQ1ig)
   7. DP on probability spaces
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=7422](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D7422&sa=D&sntz=1&usg=AFQjCNEWf0Vo0Y6mS4d5UyVv2ySuXIynmw)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=2959](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D2959&sa=D&sntz=1&usg=AFQjCNGj8cxClYTfeIvrlsmr-tUiLOTmog)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10335](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10335&sa=D&sntz=1&usg=AFQjCNGx6tIp4OgLEmrXVytzzQHyKs3hOQ)
   8. DP on trees
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10800](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10800&sa=D&sntz=1&usg=AFQjCNFTYtXeR05Zl1picOF0KI6_h4UPaw)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=10737](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10737&sa=D&sntz=1&usg=AFQjCNH3IbGp6MgQiFcoduICusCJPbJIpQ)
      * [http://www.topcoder.com/stat?c=problem\_solution&rm=266678&rd=10958&pm=8266&cr=7581406](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_solution%26rm%3D266678%26rd%3D10958%26pm%3D8266%26cr%3D7581406&sa=D&sntz=1&usg=AFQjCNGYDBIWpUwdgy9n4UQqxWvs0_8bkw)
   9. DP with datastructures
      * [http://www.spoj.pl/problems/INCSEQ/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FINCSEQ%2F&sa=D&sntz=1&usg=AFQjCNFAV8lsFKsNmhTW0GUHTmT9j1933Q)
      * [http://www.spoj.pl/problems/INCDSEQ/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FINCDSEQ%2F&sa=D&sntz=1&usg=AFQjCNFzGbiixef7-nLRnkT_eWkCdga5Ag)
      * [http://www.spoj.pl/problems/LIS2/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Franks%2FLIS2%2F&sa=D&sntz=1&usg=AFQjCNFKJwgfbiMOExW894YlstP87YJWcw)
      * [http://www.topcoder.com/stat?c=problem\_statement&pm=1986](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D1986&sa=D&sntz=1&usg=AFQjCNG2oWgJVKecQskRT4wkk2j2Oiq_Kw)
   10. Symmetric characterization of DP state
       * [http://www.topcoder.com/stat?c=problem\_statement&pm=8610](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D8610&sa=D&sntz=1&usg=AFQjCNEqAgBLHdd49j9VsVaf4XEAqoGBGA)
   11. A good collection of problems
       * [http://codeforces.com/blog/entry/325](http://www.google.com/url?q=http%3A%2F%2Fcodeforces.com%2Fblog%2Fentry%2F325&sa=D&sntz=1&usg=AFQjCNHj-ZKaV2RG8Z047rcp4JXbkiM_Hg)
       * <http://problemclassifier.appspot.com/index.jsp?search=dp&usr=>
7. **Greedy.**
   1. Suggested Reading -
      * Chapter on Greedy algorithms in Cormen.
      * http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=greedyAlg
   2. problems - refer to the topcoder tutorial.
8. **Number Theory.**
   1. 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.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DprimeNumbers&sa=D&sntz=1&usg=AFQjCNEKO-9OH1cY7-GbcfcNiR8oliKDAw)
      * Problems
        1. [http://projecteuler.net/index.php?section=problems&id=64](http://www.google.com/url?q=http%3A%2F%2Fprojecteuler.net%2Findex.php%3Fsection%3Dproblems%26id%3D64&sa=D&sntz=1&usg=AFQjCNGll3P-CKppQF_DQ6Yr9T4x7dZC8w)
        2. [http://projecteuler.net/index.php?section=problems&id=65](http://www.google.com/url?q=http%3A%2F%2Fprojecteuler.net%2Findex.php%3Fsection%3Dproblems%26id%3D65&sa=D&sntz=1&usg=AFQjCNGRZ70ap9kDbPXryA6cMFViXowVCA)
        3. [http://projecteuler.net/index.php?section=problems&id=66](http://www.google.com/url?q=http%3A%2F%2Fprojecteuler.net%2Findex.php%3Fsection%3Dproblems%26id%3D66&sa=D&sntz=1&usg=AFQjCNHvCY0cBg2UWT-wxVxnGkO1kG5o2w)
        4. [http://www.topcoder.com/stat?c=problem\_statement&pm=6408&rd=9826](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D6408%26rd%3D9826&sa=D&sntz=1&usg=AFQjCNEaPQEfpWWOP6othlFGpzb9vf9D6g)
        5. [http://www.topcoder.com/stat?c=problem\_statement&pm=2342](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D2342&sa=D&sntz=1&usg=AFQjCNHa_Cy1GHASfzpAKSx8X_TAVSnHNQ)
   2. 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](http://www.google.com/url?q=http%3A%2F%2Fprojecteuler.net%2Findex.php%3Fsection%3Dproblems%26id%3D70&sa=D&sntz=1&usg=AFQjCNHP2ypWWT0KRj6wJ-43wcZV3SfDnQ)
        2. [http://www.spoj.pl/problems/NDIVPHI/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FNDIVPHI%2F&sa=D&sntz=1&usg=AFQjCNHhJ8fV9Sqpt2WM30pD1nlrjbv7aA)
   3. 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.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10551%26rd%3D13903&sa=D&sntz=1&usg=AFQjCNHkwpzK0Y7bpA1ZxggqA_wJ2bFGJQ)
   4. Primality tests -
      * Deterministic O(sqrt(n) ) approach
      * Probabilistic primality tests - Fermat primality test, Miller-Rabin Primality test
        1. Suggested Reading -
           1. [*http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting*](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DprimalityTesting&sa=D&sntz=1&usg=AFQjCNGESIkX7rmPFjJ_yx8IKxuZRsNA4A)
           2. Cormen 31.8
           3. 2.2 from Number Theory by SY Yan
        2. Problems -
           1. PON, PRIC, SOLSTRAS on SPOJ
           2. [http://www.topcoder.com/stat?c=problem\_statement&pm=4515](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D4515&sa=D&sntz=1&usg=AFQjCNEv8xM30xjlA71lK3bgsX6u5vrt6A)
   5. Prime generation techniques - Sieve of Erastothenes
      * Suggested Problems - PRIME1 on SPOJ
   6. GCD using euclidean method
      * Suggested Reading
        1. 31.2 Cormen
      * Problems -
        1. GCD on SPOJ
        2. [http://uva.onlinejudge.org/external/114/11424.html](http://www.google.com/url?q=http%3A%2F%2Fuva.onlinejudge.org%2Fexternal%2F114%2F11424.html&sa=D&sntz=1&usg=AFQjCNFocMHM1lWNT9MtA3A1yRHROmaMKg)
   7. Logarithmic Exponentiation
      * Suggested Reading -
        1. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DprimalityTesting&sa=D&sntz=1&usg=AFQjCNGESIkX7rmPFjJ_yx8IKxuZRsNA4A)
   8. 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.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D2986%26rd%3D5862&sa=D&sntz=1&usg=AFQjCNGQkQftHtaGV4CVVUt4H0UXslmofg)
        2. [http://www.spoj.pl/problems/DIVSUM2/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FDIVSUM2%2F&sa=D&sntz=1&usg=AFQjCNGsmpSq4SsVqrMMRVVt8XYE7TKANA)
        3. [http://www.topcoder.com/stat?c=problem\_statement&pm=4481&rd=6538](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D4481%26rd%3D6538&sa=D&sntz=1&usg=AFQjCNH1lO1gpDJ99YMALj5A4EtciLlThw)
   9. Stirling numbers
   10. Wilson theorem
       * nCr % p in O(p) preprocess and O(log n ) query
   11. Lucas Theorem
   12. 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.google.com/url?q=http%3A%2F%2Fwww.codechef.com%2Fwiki%2Ftutorial-number-theory&sa=D&sntz=1&usg=AFQjCNHPFa-C7y2hFvP0lqY1SkoC4C9zxQ)
       * [http://www.algorithmist.com/index.php/Category:Number\_Theory](http://www.google.com/url?q=http%3A%2F%2Fwww.algorithmist.com%2Findex.php%2FCategory%3ANumber_Theory&sa=D&sntz=1&usg=AFQjCNH0jypXH-endzREh_SEoxeQ9QAXhQ)
   13. Problems on Number Theory -
       * [http://www.algorithmist.com/index.php/Category:Number\_Theory](http://www.google.com/url?q=http%3A%2F%2Fwww.algorithmist.com%2Findex.php%2FCategory%3ANumber_Theory&sa=D&sntz=1&usg=AFQjCNH0jypXH-endzREh_SEoxeQ9QAXhQ)
       * <http://problemclassifier.appspot.com/index.jsp?search=number&usr=>
9. **Math (Probability, Counting, Game Theory, Group Theory, Generating functions, Permutation Cycles, Linear Algebra)**
   1. **Probability.**

*Syllabus*

* + - Basic probability and Conditional probability
      1. Suggested problems
         1. [http://www.spoj.pl/problems/CT16E/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCT16E%2F&sa=D&sntz=1&usg=AFQjCNG6_fDMZSsBGjcjtCWCtENiLOOkRg)
         2. [http://www.spoj.pl/problems/CHICAGO/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCHICAGO%2F&sa=D&sntz=1&usg=AFQjCNFYu0---9CwCb7xC3uaBGyM2-Tl4A)
    - Random variables, probability generating functions
    - Mathematical expectation + Linearity of expectation
      1. Suggested problems
         1. [http://www.spoj.pl/problems/FAVDICE/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FFAVDICE%2F&sa=D&sntz=1&usg=AFQjCNGV35ZgvRCvmAlQKSNX-N6RGBDRig)
         2. [http://www.topcoder.com/stat?c=problem\_statement&pm=10744](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D10744&sa=D&sntz=1&usg=AFQjCNFOwPTb_3uuVf15P_SQ3lE0ICTq4w)
    - Special discrete and continuous probability distributions
      1. Bernoulli, Binomial, Poisson, normal distribution
      2. Suggested Problem
         1. [http://acm.sgu.ru/problem.php?contest=0&problem=498](http://www.google.com/url?q=http%3A%2F%2Facm.sgu.ru%2Fproblem.php%3Fcontest%3D0%26problem%3D498&sa=D&sntz=1&usg=AFQjCNE-qseN_HAzaurfwTjN5zt07D5jTg)
    - Suggested Readings
      1. Cormen appendix C (very basic)
      2. Topcoder probabilty tutorial [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=probabilities](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3Dprobabilities&sa=D&sntz=1&usg=AFQjCNHov438KhnvQhbabtoM8UCrZFZV0g)
      3. [http://en.wikipedia.org/wiki/Random\_variable](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FRandom_variable&sa=D&sntz=1&usg=AFQjCNEkvOf1b56_vUlvlokh_0DAPF4erg)
      4. [http://en.wikipedia.org/wiki/Expected\_value](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FExpected_value&sa=D&sntz=1&usg=AFQjCNG9JtXu_jF4NOyIEA_JqbZxSVtR7w)
      5. William Feller, An introduction to probability theory and its applications
  1. **Counting**

*Syllabus*

* + - Basic principles - Pigeon hole principle, addition, multiplication rules
      1. Suggested problems
         1. http://acm.timus.ru/problem.aspx?space=1&num=1690
         2. http://www.topcoder.com/stat?c=problem\_statement&pm=10805
      2. Suggested readings
         1. http://en.wikipedia.org/wiki/Combinatorial\_principles
         2. http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=combinatorics
         3. http://www.maa.org/editorial/knot/pigeonhole.html
    - Inclusion-exclusion
      1. Suggested readings
         1. http://en.wikipedia.org/wiki/Inclusion–exclusion\_principle
      2. Suggested problems
         1. http://www.topcoder.com/stat?c=problem\_statement&pm=4463&rd=6536
         2. http://www.topcoder.com/stat?c=problem\_statement&pm=10238
    - Special numbers
      1. Suggested reading - Stirling, eurlerian, harmonic, bernoulli, fibonnacci numbers
         1. http://en.wikipedia.org/wiki/Stirling\_number
         2. http://en.wikipedia.org/wiki/Eulerian\_numbers
         3. http://en.wikipedia.org/wiki/Harmonic\_series\_(mathematics)
         4. http://en.wikipedia.org/wiki/Bernoulli\_number
         5. http://en.wikipedia.org/wiki/Fibonnaci\_numbers
         6. Concrete mathematics by Knuth
      2. Suggested problems
         1. http://www.topcoder.com/stat?c=problem\_statement&pm=1643
         2. http://www.topcoder.com/stat?c=problem\_statement&pm=8202&rd=11125
         3. http://www.topcoder.com/stat?c=problem\_statement&pm=8725
         4. http://www.topcoder.com/stat?c=problem\_statement&pm=2292&rd=10709
    - Advanced counting techniques - Polya counting, burnsides lemma
      1. Suggested reading
         1. http://en.wikipedia.org/wiki/Burnside's\_lemma
         2. http://petr-mitrichev.blogspot.com/2008/11/burnsides-lemma.html
      2. Suggested Problems
         1. http://www.topcoder.com/stat?c=problem\_statement&pm=9975
         2. http://www.spoj.pl/problems/TRANSP/

c. Game theory

*Syllabus*

* + - Basic principles and Nim game
      1. Sprague grundy theorem, grundy numbers
      2. Suggested readings
         1. http://en.wikipedia.org/wiki/Sprague%E2%80%93Grundy\_theorem
         2. http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=algorithmGames
         3. http://www.ams.org/samplings/feature-column/fcarc-games1
         4. http://www.codechef.com/wiki/tutorial-game-theory
      3. Suggested problems
         1. http://www.topcoder.com/stat?c=problem\_statement&pm=3491&rd=6517
         2. http://www.topcoder.com/stat?c=problem\_statement&pm=3491&rd=6517
    - Hackenbush
      1. Suggested readings
         1. http://en.wikipedia.org/wiki/Hackenbush
         2. http://www.ams.org/samplings/feature-column/fcarc-partizan1
      2. Suggested problems
         1. http://www.cs.caltech.edu/ipsc/problems/g.html
         2. http://www.spoj.pl/problems/PT07A/

**d. Linear Algebra**

*Syllabus*

* + - Matrix Operations
      1. Addition and subtraction of matrices
         1. Suggested Reading

Cormen 28.1

* + - 1. Multiplication ( Strassen's algorithm ), logarithmic exponentiation
         1. Suggested reading

Cormen 28.2

Linear Algebra by Kenneth Hoffman Section 1.6

* + - * 1. Problems

http://uva.onlinejudge.org/external/111/11149.html

* + - 1. Matrix transformations [ Transpose, Rotation of Matrix, Representing Linear transformations using matrix ]
         1. Suggested Reading

Linear Algebra By Kenneth Hoffman Section 3.1,3.2,3.4,3.7

* + - * 1. Problems

http://www.topcoder.com/stat?c=problem\_statement&pm=6877

JPIX on Spoj

* + - 1. Determinant , Rank and Inverse of Matrix [ Gaussean Elimination , Gauss Jordan Elimination]
         1. Suggested Reading

28.4 Cormen

Linear Algebra by Kenneth Chapter 1

* + - * 1. Problems

http://www.topcoder.com/stat?c=problem\_statement&pm=8174

http://www.topcoder.com/stat?c=problem\_statement&pm=6407&rd=9986

http://www.topcoder.com/stat?c=problem\_statement&pm=8587

HIGH on Spoj

* + - 1. Solving system of linear equations
         1. Suggested Reading

28.3 Cormen

Linear Algebra by Kenneth Chapter 1

* + - * 1. Problems -

http://www.topcoder.com/stat?c=problem\_statement&pm=3942&rd=6520

* + - 1. Using matrix exponentiation to solve recurrences
         1. Suggested Reading

[http://www.topcoder.com/tc?module=Static&d1=features&d2=010408](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dfeatures%26d2%3D010408&sa=D&sntz=1&usg=AFQjCNHpIKTDyWZYmQ5DT1nW2aAGMR9SuA)

* + - * 1. Problems

REC, RABBIT1 , PLHOP on spoj

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=6877

* + - 1. Eigen values and Eigen vectors
         1. Problems

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 ]
         1. Problems

http://www.topcoder.com/stat?c=problem\_statement&pm=8273&rd=10798

POLYEQ , ROOTCIPH on Spoj

* + - 1. Lagrange Interpolation
         1. Problems

http://www.topcoder.com/stat?c=problem\_statement&pm=10239

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

* + - Bernside Lemma, Polias theorem
      1. Suggested Reading
         1. Hernstein's topics in algebra
         2. <http://petr-mitrichev.blogspot.com/2008/11/burnsides-lemma.html>
      2. Problems
         1. TRANSP on spoj
         2. http://www.topcoder.com/stat?c=problem\_statement&pm=9975
  1. Generating functions
     + Suggested Reading
       1. Herbert Wilf's generating functionology
       2. Robert Sedgewick and Flajoulet's Combinatorial analysis

1. **Data Structures.**
2. **Basic**
3. Arrays/Stacks/Queues :

* Problems

1. [https://www.spoj.pl/problems/STPAR/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSTPAR%2F&sa=D&sntz=1&usg=AFQjCNEq6A04RzU3S1ePUBKXyW1xgM5dAg)
2. [https://www.spoj.pl/problems/SHOP/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSHOP%2F&sa=D&sntz=1&usg=AFQjCNEt7s3kwWaepXxjtxpj2_m_0AjvoA)
3. [https://www.spoj.pl/problems/WATER/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FWATER%2F&sa=D&sntz=1&usg=AFQjCNF8qWyfoJ2kKC55ugxVTkH9ZDEdUw)

* Reading:

1. CLRS: section 10.1
2. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=dataStructures](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DdataStructures&sa=D&sntz=1&usg=AFQjCNHDAKr3N1J_wQq8hfWMoliIueBv4Q)

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/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FHASHIT%2F&sa=D&sntz=1&usg=AFQjCNEI7U2uOrPolCyoGG4F0TAeWOGGxw)
2. [https://www.spoj.pl/problems/CUCKOO/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCUCKOO%2F&sa=D&sntz=1&usg=AFQjCNF3m4kuZ7P5JSF6_vye2zTJap_vsQ)

* Reading: CLRS: Chapter 11, Mark Allen Weies Chapter 5

d. Circular linked list / queue

* Problems
  + - 1. [https://www.spoj.pl/problems/CTRICK/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCTRICK%2F&sa=D&sntz=1&usg=AFQjCNGlZz0BJVJw9_w2WcP8wsGC0w1gWw)

e. Binary/nary Trees

* Reading

1. CLRS: section 10.4
2. CLRS: Chapter 12
3. Mark Allen Weies Chapter 4
4. h[ttp://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binarySearchRedBlack](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DbinarySearchRedBlack&sa=D&sntz=1&usg=AFQjCNFQYJmnmOCklYKNKLgbx7DIamv7CA)

f. Heaps

* Problems

1. [https://www.spoj.pl/problems/PRO/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPRO%2F&sa=D&sntz=1&usg=AFQjCNHJh9hj8HUSkeoNvqEoi_QvihAvqQ)
2. h[ttps://www.spoj.pl/problems/EXPEDI/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEXPEDI%2F&sa=D&sntz=1&usg=AFQjCNH-yiza1bADOLmFz5rWoRXMqeWqvQ)

* Reading : Mark Allen Weies Chapter 6

**ii. Advanced**

1. Trie (Keyword tree)
   * + Problems
2. [https://www.spoj.pl/problems/MORSE/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMORSE%2F&sa=D&sntz=1&usg=AFQjCNHImKo3SeuzclVNFWfbCLDs1-M_uQ)
3. [https://www.spoj.pl/problems/EMOTICON/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEMOTICON%2F&sa=D&sntz=1&usg=AFQjCNHzwNIDK98gg4Hv1I0AjIAlG-ZC1A)
   * + Reading
4. Interval trees / Segment Trees
   * + Problems
       1. [https://www.spoj.pl/problems/ORDERS/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FORDERS%2F&sa=D&sntz=1&usg=AFQjCNH595gMpucXwN3yyEY-JfhSQuNO8g)
       2. [https://www.spoj.pl/problems/FREQUENT/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FFREQUENT%2F&sa=D&sntz=1&usg=AFQjCNFUiZPMiokXs-mediQ60rfUeV8dGQ)
     + Reading
5. Fenwick(Binary Indexed) trees
   * + Problems
       1. [https://www.spoj.pl/problems/MATSUM/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMATSUM%2F&sa=D&sntz=1&usg=AFQjCNF510mlk85u0iVnlnLBdKeWoHDIsg)
     + Reading: [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DbinaryIndexedTrees&sa=D&sntz=1&usg=AFQjCNEvQfok3Ial7efxYumQFYDjrK7V0w)
6. Disjoint data structures
   * + Problems
       1. [https://www.spoj.pl/problems/BLINNET/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FBLINNET%2F&sa=D&sntz=1&usg=AFQjCNGc-tjbgaQm5pp9_VPQiYamiNZOxw)
       2. [https://www.spoj.pl/problems/CHAIN/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCHAIN%2F&sa=D&sntz=1&usg=AFQjCNFCNjrvQAEP_8BiULzaST3mqj38PQ)
     + Reading:
       1. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=disjointDataStructure](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DdisjointDataStructure&sa=D&sntz=1&usg=AFQjCNERkZqikzPKw61ZyaGVwT4P5aa9zQ)
       2. Mark Allen Weies Chapter 8
7. Range minimum Query(RMQ)
   * + Problems
       1. [https://www.spoj.pl/problems/GSS1/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FGSS1%2F&sa=D&sntz=1&usg=AFQjCNG4w_e9fbWPtNh2qOgvVDL-tnh7Ag)
     + Reading [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lowestCommonAncestor](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DlowestCommonAncestor&sa=D&sntz=1&usg=AFQjCNH520nkp8-EIP9gtemN5sJOeoIUMQ)
8. Customized interval/segment trees (Augmented DS)
   * + Problems
       1. [https://www.spoj.pl/problems/GSS3/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FGSS3%2F&sa=D&sntz=1&usg=AFQjCNFCyG1ZzMqG5ycXrmUEmwGBq7YeWA)
       2. [https://www.spoj.pl/problems/RRSCHED/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FRRSCHED%2F&sa=D&sntz=1&usg=AFQjCNGDFjjAKCNVpoq-2vo0nrYiMpjdug)
     + Reading: CLRS: Chapter 14 (augmented DS)

g. AVL Trees

* + - Problems

1. [https://www.spoj.pl/problems/ORDERS/](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FORDERS%2F&sa=D&sntz=1&usg=AFQjCNH595gMpucXwN3yyEY-JfhSQuNO8g)

* + - Reading

**iii. Miscellaneous (Not to be covered)**

* 1. Splay Trees
  2. B/B+ Trees
  3. k-d Trees
  4. Red-black Trees
  5. Skip List
  6. Binomial/ Fibonacci heaps

**iv. Exercices**

1. [**https://www.spoj.pl/problems/LAZYPROG**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLAZYPROG&sa=D&sntz=1&usg=AFQjCNFp-sbJ85NHaFqr01hXfVUrWCzU7w)**/ (Hint: Heaps)t**
2. [**https://www.spoj.pl/problems/HELPR2D2/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FHELPR2D2%2F&sa=D&sntz=1&usg=AFQjCNE409cp7zM9XvfhQn5Yc0K15uMvTQ) **(Hint: Interval Trees)**
3. [**https://www.spoj.pl/problems/SAM/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FSAM%2F&sa=D&sntz=1&usg=AFQjCNEdaTd8wCBjEpIjQrHgRSi_k9_xXg) **(Hint: Heaps)**
4. [**https://www.spoj.pl/problems/PRHYME/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FPRHYME%2F&sa=D&sntz=1&usg=AFQjCNFm8f7BkiJtnBL07U34lsamMQowqA) **(Hint: Trie)**
5. [**https://www.spoj.pl/problems/HEAPULM/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FHEAPULM%2F&sa=D&sntz=1&usg=AFQjCNEbEp5yyVDrNpXKbXw1m0D_cOJj8g) **(Hint: Interval Trees)**
6. [**https://www.spoj.pl/problems/CORNET/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FCORNET%2F&sa=D&sntz=1&usg=AFQjCNFAoLqE7p3PQjAFyqJASo6bnwUzKw) **(Hint: Disjoint )**
7. [**https://www.spoj.pl/problems/EXPAND/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FEXPAND%2F&sa=D&sntz=1&usg=AFQjCNHTKS92PsE2D8GTNJ8RnO3ILv4V8w)
8. [**https://www.spoj.pl/problems/WPUZZLES/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FWPUZZLES%2F&sa=D&sntz=1&usg=AFQjCNEpgmA2D6PkKLyEpq5xMVdDHlja1g)
9. [**https://www.spoj.pl/problems/LIS2/**](http://www.google.com/url?q=https%3A%2F%2Fwww.spoj.pl%2Fproblems%2FLIS2%2F&sa=D&sntz=1&usg=AFQjCNE-RvEL7_k-afvDeu-zXdclyKOWfw)
10. **Search Techniques/Bruteforce writing techniques/Randomized algorithms.**
    1. Backtracking - [Beginner].
       * problems ->
         1. N queens problems
         2. Knights Tour
         3. Sudoku Problem
         4. Tiling Problem.
         5. 15 puzzle.
    2. 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](http://www.google.com/url?q=http%3A%2F%2Fwww-cs-faculty.stanford.edu%2F~uno%2Fpapers%2Fdancing-color.ps.gz&sa=D&sntz=1&usg=AFQjCNHu0lxKhGN_idpOsYodoLldZxTvXw)
    3. Binary Search - [Beginner].
       * poblems - 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](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DbinarySearch&sa=D&sntz=1&usg=AFQjCNGI8fRSFi-dpE53Pk6OzDdOw3QNhw)
    4. Ternary Search - [Intermediate].
       * problems -
         1. [http://www.spoj.pl/problems/KPPOLY/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FKPPOLY%2F&sa=D&sntz=1&usg=AFQjCNFOq30QswwLH7lhuMwDrnuyjkhYgA)
         2. [http://www.codechef.com/DEC09/problems/K1/](http://www.google.com/url?q=http%3A%2F%2Fwww.codechef.com%2FDEC09%2Fproblems%2FK1%2F&sa=D&sntz=1&usg=AFQjCNHIRBoHnR5gyqsd1exQdsRQKn2PAg)
         3. [http://www.topcoder.com/stat?c=problem\_statement&pm=4705&rd=7993](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D4705%26rd%3D7993&sa=D&sntz=1&usg=AFQjCNFDc6UFaEVlMzQ9S6uhSdt3bWaJJA)
         4. [http://www.topcoder.com/stat?c=problem\_statement&pm=7741&rd=10671](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D7741%26rd%3D10671&sa=D&sntz=1&usg=AFQjCNENkNqruJhvydQpExwP_LQEUV-ZHg)
         5. [http://www.topcoder.com/stat?c=problem\_statement&pm=6464&rd=9994](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D6464%26rd%3D9994&sa=D&sntz=1&usg=AFQjCNFDlxrJ1_4THnPGQpZA75bb6vii2Q)
         6. [http://www.topcoder.com/stat?c=problem\_statement&pm=3501&rd=6529](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D3501%26rd%3D6529&sa=D&sntz=1&usg=AFQjCNHeiaHTNY3qUhuKrimsy3nTKJtMCQ)
         7. [http://www.topcoder.com/stat?c=problem\_statement&pm=4567&rd=6539](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Fstat%3Fc%3Dproblem_statement%26pm%3D4567%26rd%3D6539&sa=D&sntz=1&usg=AFQjCNGyqDMvoRfcl1graaaUc5d3b7qrhw)
    5. Meet in the middle [Intermediate].
       * problems -
         1. [http://www.spoj.pl/problems/MAXISET/](http://www.google.com/url?q=http%3A%2F%2Fwww.spoj.pl%2Fproblems%2FMAXISET%2F&sa=D&sntz=1&usg=AFQjCNGaXBYpDzr-X0I309ZcOZqh5aU81g)
         2. [http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=2868](http://www.google.com/url?q=http%3A%2F%2Facm.zju.edu.cn%2Fonlinejudge%2FshowProblem.do%3FproblemCode%3D2868&sa=D&sntz=1&usg=AFQjCNGHIjcKf7UscbqzOr_PIQ1zlppOTw)
    6. Hill Climbing [Advanced].
    7. 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.
    8. Randomized Algorithms [Intermediate]-
       * Quick-Sort.
11. **General programming issues in contests** ->
    1. Arithmetic Precision - [Beginner].
       * Suggested Reading -
         1. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=integersReals](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DintegersReals&sa=D&sntz=1&usg=AFQjCNEEQUAd3b6QJ5OtYdw2VK48GqyHrA)
    2. Representing sets with bitmasks and manipulating bitmasks - [Beginner].
       * Suggested Reading -
         1. [http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=bitManipulation](http://www.google.com/url?q=http%3A%2F%2Fwww.topcoder.com%2Ftc%3Fmodule%3DStatic%26d1%3Dtutorials%26d2%3DbitManipulation&sa=D&sntz=1&usg=AFQjCNHKFrJp82CK9QwV0MRXLjVfFgn3lw)
       * problems - refer to the tutorial link in Suggested reading section.