List of Topics for programming Competitions -

1. **Basic Geometry/Euclidean Geometry/Coordfinate 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>
   4. 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.
   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>
      * 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.spoj.pl/problems/BSHEEP), [BULK](http://www.spoj.pl/problems/BULK), [SEGVIS](http://www.spoj.pl/problems/SEGVIS), [CONDUIT](http://www.spoj.pl/problems/CONDUIT), [RUNAWAY](http://www.spoj.pl/problems/RUNAWAY), [DIRVS](http://www.spoj.pl/problems/DIRVS), [RAIN1](http://www.spoj.pl/problems/RAIN1), [SHAMAN](http://www.spoj.pl/problems/SHAMAN), [TCUTTER](http://www.spoj.pl/problems/TCUTTER), [LITEPIPE](http://www.spoj.pl/problems/LITEPIPE), [RHOMBS](http://www.spoj.pl/problems/RHOMBS), [FSHEEP](http://www.spoj.pl/problems/FSHEEP), [FLBRKLIN](http://www.spoj.pl/problems/FLBRKLIN), [CERC07P](http://www.spoj.pl/problems/CERC07P), [BAC](http://www.spoj.pl/problems/BAC), [ALTARS](http://www.spoj.pl/problems/ALTARS), [CERC07C](http://www.spoj.pl/problems/CERC07C), [NECKLACE](http://www.spoj.pl/problems/NECKLACE), [CH3D](http://www.spoj.pl/problems/CH3D), [RECTANGL](http://www.spoj.pl/problems/RECTANGL), [POLYSSQ](http://www.spoj.pl/problems/POLYSSQ), [FOREST2](http://www.spoj.pl/problems/FOREST2), [KPPOLY](http://www.spoj.pl/problems/KPPOLY), [RAIN2](http://www.spoj.pl/problems/RAIN2), [SEGMENTS](http://www.spoj.pl/problems/SEGMENTS), [ARCHPLG](http://www.spoj.pl/problems/ARCHPLG), [BALLOON](http://www.spoj.pl/problems/BALLOON), [CIRCLES](http://www.spoj.pl/problems/CIRCLES), [COMPASS](http://www.spoj.pl/problems/COMPASS), [EOWAMRT](http://www.spoj.pl/problems/EOWAMRT), [ICERINK](http://www.spoj.pl/problems/ICERINK) on SPOJ.
       * [CultureGrowth](http://www.topcoder.com/stat?c=problem_statement&pm=3996), [PolygonCover](http://www.topcoder.com/stat?c=problem_statement&pm=8540) 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>
   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-dimentional pattern matching.
   10. Problems on Strings [can be solved with a variety of techniques] -
       * [DISUBSTR](http://www.spoj.pl/problems/DISUBSTR), [PLD](http://www.spoj.pl/problems/PLD/), [MSTRING](http://www.spoj.pl/problems/MSTRING/), [REPEATS](http://www.spoj.pl/problems/REPEATS), [JEWELS](http://www.spoj.pl/problems/JEWELS), [ARCHIVER](http://www.spoj.pl/problems/ARCHIVER), [PROPKEY](http://www.spoj.pl/problems/PROPKEY), [LITELANG](http://www.spoj.pl/problems/LITELANG), [EMOTICON](http://www.spoj.pl/problems/EMOTICON), [WORDS](http://www.spoj.pl/problems/WORDS), [AMCODES](http://www.spoj.pl/problems/AMCODES), [UCODES](http://www.spoj.pl/problems/UCODES), [PT07H](http://www.spoj.pl/problems/PT07H), [MINSEQ](http://www.spoj.pl/problems/MINSEQ), [TOPALIN](http://www.spoj.pl/problems/TOPALIN), [BWHEELER](http://www.spoj.pl/problems/BWHEELER), [BEADS](http://www.spoj.pl/problems/BEADS), [SARRAY](http://www.spoj.pl/problems/SARRAY), [LCS](http://www.spoj.pl/problems/LCS), [LCS2](http://www.spoj.pl/problems/LCS2), [SUBST1](http://www.spoj.pl/problems/SUBST1), [PHRASES](http://www.spoj.pl/problems/PHRASES), [PRETILE](http://www.spoj.pl/problems/PRETILE) on SPOJ
       * <http://www.algorithmist.com/index.php/Category:String_algorithms>
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.spoj.pl/problems/PPATH), [ONEZERO](http://www.spoj.pl/problems/ONEZERO), [WATER](http://www.spoj.pl/problems/WATER) on SPOJ
   3. Depth First Search.
   4. Strongly Connected Components.
      * problems -
        1. [TOUR](http://www.spoj.pl/problems/TOUR) and [BOTTOM](http://www.spoj.pl/problems/BOTTOM) on SPOJ.
   5. Biconnected Components, Finding articulation points and bridges].
      * problems -
        1. [RELINETS](http://www.spoj.pl/problems/RELINETS), [PT07A](http://www.spoj.pl/problems/PT07A) on SPOJ.
   6. Dijkstra algorithm -
      * problems -
        1. [SHPATH](http://www.spoj.pl/problems/SHPATH) on SPOJ.
   7. Floyd Warshall algorithm -
      * problems -
        1. [COURIER](http://www.spoj.pl/problems/COURIER) on SPOJ.
   8. Minimum Spanning Tree
      * problems -
        1. [BLINNET](http://www.spoj.pl/problems/BLINNET) on SPOJ.
   9. Flood-fill algorithm
   10. Topological sort
   11. Bellman-Ford algorithm.
   12. Euler Tour/Path.
       * problems - [WORDS1](http://www.spoj.pl/problems/WORDS1) on SPOJ.
   13. 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. [Intermediate/Advanced].**
   1. Maximum flow using Ford Fulkerson Method.
      * Suggested Reading -
        1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=maxFlow>
      1. problems - [TAXI](http://www.spoj.pl/problems/TAXI), [POTHOLE](http://www.spoj.pl/problems/POTHOLE), [IM](http://www.spoj.pl/problems/IM), [QUEST4](http://www.spoj.pl/problems/QUEST4), [MUDDY](http://www.spoj.pl/problems/MUDDY), [EN](http://www.spoj.pl/problems/EN), [CABLETV](http://www.spoj.pl/problems/CABLETV), [STEAD](http://www.spoj.pl/problems/STEAD), [NETADMIN](http://www.spoj.pl/problems/NETADMIN), [COCONUTS](http://www.spoj.pl/problems/COCONUTS), [OPTM](http://www.spoj.pl/problems/OPTM) on SPOJ.
   2. Maximum flow using Dinics Algorithm.
      * Problems - [PROFIT](http://www.spoj.pl/problems/PROFIT) 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>
   4. Maximum weighted Bipartite Matching (Kuhn Munkras algorithm/Hungarian Method)
      * problems - [GREED](http://www.spoj.pl/problems/GREED), [SCITIES](http://www.spoj.pl/problems/SCITIES), [TOURS](http://www.spoj.pl/problems/TOURS) on SPOJ | <http://www.topcoder.com/stat?c=problem_statement&pm=8143>
   5. Stoer Wagner min-cut algorithm.
   6. Hopcroft Karp bipartite matching algorithm.
      * problems - [ANGELS](http://www.spoj.pl/problems/ANGELS) on SPOJ.
   7. Maximum matching in general graph (blossom shrinking)
   8. Gomory-Hu Trees.
      * i) Problems - [MCQUERY](http://www.spoj.pl/problems/MCQUERY) on Spoj.
   9. 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/>
   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.topcoder.com/stat?c=problem_statement&pm=10765&rd=14183>
   3. State space reduction
      * <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=8605&rd=12012&rm=269199&cr=7581406>
   4. 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>
   5. 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=7849>
   6. 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=10806>
      * <http://www.topcoder.com/stat?c=problem_statement&pm=7828>
      * <http://www.topcoder.com/stat?c=problem_statement&pm=7316>
   7. DP on probability spaces
      * <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=10335>
   8. DP on trees
      * <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_solution&rm=266678&rd=10958&pm=8266&cr=7581406>
   9. DP with datastructures
      * <http://www.spoj.pl/problems/INCSEQ/>
      * <http://www.spoj.pl/problems/INCDSEQ/>
      * [http://www.spoj.pl/problems/LIS2/](http://www.spoj.pl/ranks/LIS2/)
      * <http://www.topcoder.com/stat?c=problem_statement&pm=1986>
   10. Symmetric characterization of DP state
       * <http://www.topcoder.com/stat?c=problem_statement&pm=8610>
   11. A good collection of problems
       * <http://codeforces.com/blog/entry/325>
       * <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.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>
        5. <http://www.topcoder.com/stat?c=problem_statement&pm=2342>
   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>
        2. <http://www.spoj.pl/problems/NDIVPHI/>
   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>
   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.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting)
           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>
   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>
   7. Logarithmic Exponentiation
      * Suggested Reading -
        1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting>
   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>
        2. <http://www.spoj.pl/problems/DIVSUM2/>
        3. <http://www.topcoder.com/stat?c=problem_statement&pm=4481&rd=6538>
   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.algorithmist.com/index.php/Category:Number_Theory>
   13. Problems on 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)**
   1. **Probability.**

*Syllabus*

* + - Basic probability and Conditional probability
      1. Suggested problems
         1. <http://www.spoj.pl/problems/CT16E/>
         2. <http://www.spoj.pl/problems/CHICAGO/>
    - Random variables, probability generating functions
    - Mathematical expectation + Linearity of expectation
      1. Suggested problems
         1. <http://www.spoj.pl/problems/FAVDICE/>
         2. <http://www.topcoder.com/stat?c=problem_statement&pm=10744>
    1. 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>
    2. Suggested Readings
       1. Cormen appendix C (very basic)
       2. Topcoder probabilty tutorial<http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=probabilities>
       3. <http://en.wikipedia.org/wiki/Random_variable>
       4. <http://en.wikipedia.org/wiki/Expected_value>
       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
    1. 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>

* + - * 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
  + 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/>
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. h[ttps://www.spoj.pl/problems/POSTERS/](about:blank)

* 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. h[ttp://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binarySearchRedBlack](http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binarySearchRedBlack)

           f. Heaps

* Problems

1. <https://www.spoj.pl/problems/PRO/>
2. h[ttps://www.spoj.pl/problems/EXPEDI/](https://www.spoj.pl/problems/EXPEDI/)

* Reading : Mark Allen Weies Chapter 6

**ii. Advanced**

1. Trie (Keyword tree)
   * + Problems
2. <https://www.spoj.pl/problems/MORSE/>
3. <https://www.spoj.pl/problems/EMOTICON/>
   * + Reading
4. Interval trees / Segment Trees
   * + Problems
       1. <https://www.spoj.pl/problems/ORDERS/>
       2. <https://www.spoj.pl/problems/FREQUENT/>
     + Reading
5. Fenwick(Binary Indexed) trees
   * + Problems
       1. <https://www.spoj.pl/problems/MATSUM/>
     + Reading: <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees>
6. 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
7. Range minimum Query(RMQ)
   * + Problems
       1. <https://www.spoj.pl/problems/GSS1/>
     + Reading <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lowestCommonAncestor>
8. 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)**

* 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**](https://www.spoj.pl/problems/LAZYPROG)**/ (Hint: Heaps)t**
2. [**https://www.spoj.pl/problems/HELPR2D2/**](https://www.spoj.pl/problems/HELPR2D2/) **(Hint: Interval Trees)**
3. [**https://www.spoj.pl/problems/SAM/**](https://www.spoj.pl/problems/SAM/) **(Hint: Heaps)**
4. [**https://www.spoj.pl/problems/PRHYME/**](https://www.spoj.pl/problems/PRHYME/) **(Hint: Trie)**
5. [**https://www.spoj.pl/problems/HEAPULM/**](https://www.spoj.pl/problems/HEAPULM/) **(Hint: Interval Trees)**
6. [**https://www.spoj.pl/problems/CORNET/**](https://www.spoj.pl/problems/CORNET/) **(Hint: Disjoint )**
7. [**https://www.spoj.pl/problems/EXPAND/**](https://www.spoj.pl/problems/EXPAND/)
8. [**https://www.spoj.pl/problems/WPUZZLES/**](https://www.spoj.pl/problems/WPUZZLES/)
9. [**https://www.spoj.pl/problems/LIS2/**](https://www.spoj.pl/problems/LIS2/)
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>
    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>
    4. 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>
         4. <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>
         6. <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>
    5. Meet in the middle [Intermediate].
       * problems -
         1. <http://www.spoj.pl/problems/MAXISET/>
         2. <http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=2868>
    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>
    2. Representing sets with bitmasks and manipulating bitmasks - [Beginner].
       1. Suggested Reading -
          1. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=bitManipulation>
       2. problems - refer to the tutorial link in Suggested reading section.