

Programming Syllabus For advance Level In CCS(CUET Computer Society)

Geometry

1. Computational Geometry

- a. Graham scan algorithm for convex Hull $O(n \cdot \log(n))$
- b. Online construction of 3-D convex Hull in $O(n^2)$
- c. Bentley ottmann algorithm to list all intersecting points of n line segments in $O((n+1) \cdot \log(n))$
 - http://softsurfer.com/Archive/algorithm_0108/algorithm_0108.htm
- d. Rotating Calipers Technique
 - <http://cgm.cs.mcgill.ca/~orm/rotcal.html>
 - Problem refer to Rotating calipers technique
- e. Line sweep/ Plane sweep algorithms
 - Area / Perimeter of Union of Rectangles.
 - Closet Pair of Points.
 - <http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=lineSweep>
- f. Area of Union of Circles
- g. Delaunay Triangulation of n points in $O(n \cdot \log(n))$
- h. Voronoi Diagrams of n points in $O(n \cdot \log(n))$ using Fortunes algorithm
- i. points in a polygon problem –
 - $O(n)$ solution without preprocessing
 - $O(\log n)$ algorithm with $O(n \cdot \log(n))$ preprocessing for convex polygons
- j. Problems under this category-
 - BSHEEP, BLUX, CONDUIT, RUNAWAY, SHAMAN, TCUTTER, LITEPIPE, FSHEEP, FLBRKLIN, BAC, COMPASS, CIRCLES, SEGMENTS, RAIN2, KPPOLY, RECTANGL on SPOJ
 - CultureGrowth, PolygonCover on Topcoder

String Algorithm

- Dictionary of Basic Factors
 - $O(n \cdot \log(n))$ method of DBT construction using Radix sort
- Manacher's algorithm to find Length of palindromic substring of a string centered at a position for each position in the string
 - Runtime $\rightarrow O(n)$
- Multi-dimensional pattern matching
- Problems on Strings [can be solved with a variety of techniques]
 - DISUBSTER, PLD, MSTRING, REPPEATS, JEWELS, ARCHIVER, PROPXKEY, LITELANG, EMOTICON, WORDS, AMCODES, TOPALIN, BEADS, SARRY, LCS, LCS2, SUBSTR on SPOJ
 - http://algorithmist.com/index.php/Category:String_algorithms

Graph

- Euler Tour/Path
 - problems - WORDS1 on SPOJ
- Hamiltonian Cycle
- Kth Shortest Path
- Suggested reading for most of the topics in graph algorithm -
 - <http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=graphsDataStructs1>

Flow Networks/Matching

- a. Maximum flow using Ford Fulkerson Method
 - Suggested reading-
<http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=maxFlow>
 - Problems - TAXI, POTHOLES, TM QUEST4, MUDDY, EN STEAD, COCONUTS 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 -
<http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=minimumCostFlow1>
- d. Maximum Weighted Bipartite Matching (Kuhn Munkres algorithm/Hungarian algorithm)
 - problems - GREED, SCITIES, TOURS on SPOJ
 - http://community.topcoder.com/stat?c=problem_statement&pm=8143
- e. Stoer wagner min-cut algorithm
- f. Hopcroft karp bipartite matching algorithm
 - problems- ANGLES on SPOJ
- g. Maximum matching in general graph (blossom shrinking)
- h. Suggested reading for Full category
 - Network flow- Algorithms and Applications by Ahuja

Dynamic Programming:

- Bitmask DP (Traveling salesman problem)
 - Modular DP (DP with MOD value as a state)
-
- a. Suggested Reading - Dynamic Programming(DP) as a tabulation method
 - Corman 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=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=3001
 - 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
- 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=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=10806
 - http://www.topcoder.com/stat?c=problem_statement&pm=7828
 - 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=2959
 - 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=10737
 - http://www.topcoder.com/stat?c=problem_solution&rm=266678&rd=10958&pm=8266&cr=7581406
- i. Symmetric characterization of DP state
- http://www.topcoder.com/stat?c=problem_statement&pm=8610
- j. A good collection of problems
- <http://codeforces.com/blog/entry/325>
 - <http://problemclassifier.appspot.com/index.jsp?search=dp&usr=>

Greedy:

- Maximum Sum 2D in $O(n^3)$
- Maximum Rectangle $O(n^2)$

Number Theory:

- a. Chinese remainder theorem
- Suggested Reading

1.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

b. Logarithmic Exponentiation

■ Suggested Reading -

1.<http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primalityTesting>

c. 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

d. Stirling numbers

e. Wilson theorem

■ $nCr \% p$ in $O(p)$ preprocess and $O(\log n)$ query

f. Lucas Theorem 1.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_Theorem.

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)

a. Probability:

■ 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 probability 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

b. Counting

■ Special numbers

1. Suggested reading

- Stirling, Eulerian, harmonic, Bernoulli, Fibonacci numbers

- a. http://en.wikipedia.org/wiki/Stirling_number
- b. 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
- e. http://en.wikipedia.org/wiki/Fibonacci_numbers

Knuth

2. Suggested problems

- a. http://www.topcoder.com/stat?c=problem_statement&pm=1643
- b. http://www.topcoder.com/stat?c=problem_statement&pm=8202&rd=11125
- c. http://www.topcoder.com/stat?c=problem_statement&pm=8725
- d. http://www.topcoder.com/stat?c=problem_statement&pm=2292&rd=10709

■ Advanced counting techniques

- Polya counting, Burnside's lemma

1. Suggested reading

- a. 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
- b. <http://www.spoj.pl/problems/TRANSP/>

c. Game theory

■ 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

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

b. 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

■ Matrix Operations

1. 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

ii. JPIX on Spoj

2. Determinant, Rank and Inverse of Matrix [Gaussian Elimination, Gauss Jordan Elimination]

a. Suggested Reading

i. CORMEN

ii. Linear Algebra by Kenneth Chapter 1

b. Problems

i. http://www.topcoder.com/stat?c=problem_statement&pm=8174

ii. http://www.topcoder.com/stat?c=problem_statement&pm=6407&rd=9986

iii. http://www.topcoder.com/stat?c=problem_statement&pm=8587

iv. HIGH on Spoj

4. Solving system of linear equations

a. Suggested Reading

- i. Cormen
- ii. Linear Algebra by Kenneth Chapter 1

b. Problems -

- i. http://www.topcoder.com/stat?c=problem_statement&pm=3942&rd=6520

5. 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=7262,
http://www.topcoder.com/stat?c=problem_statement&pm=6877

6. Eigen values and Eigen vectors

a. Problems

- i. http://www.topcoder.com/stat?c=problem_statement&pm=2423&rd=4780

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. Herstein'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

g. Generating functions

■ Suggested Reading

1. Herbert Wilf's generating functionology
2. Robert Sedgewick and Flajolet's Combinatorial analysis

Data Structures.

i. Basic

a. 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

ii. Advanced

a. Interval trees / Segment Trees

■ Problems

1. <https://www.spoj.pl/problems/ORDERS/>
2. <https://www.spoj.pl/problems/FREQUENT/>

■ Reading

b. Fenwick (Binary Indexed) trees

■ Problems

1. <https://www.spoj.pl/problems/MATSUM/>

■ Reading: <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees>

c. 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

d. Range minimum Query (RMQ)

■ Problems

1. <https://www.spoj.pl/problems/GSS1/>

■ Reading: <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lowestCommonAncestor>

e. 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)

f. AVL Trees

■ Problems 1. <https://www.spoj.pl/problems/ORDERS/>

g. BST & Variation

h. Least Common Ancestor

iii. Miscellaneous [if possible]

- 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/>

Search Techniques/Bruteforce writing techniques/Randomized algorithms:

a. Backtracking - [Beginner].

■problems-

1. Sudoku Problem
2. Tiling Problem.
3. 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. Hill Climbing [Advanced].

d. 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.

e. Min-max algorithm