Topic 1: Time Complexity - final

Chapter 1 - Introduction to PS.pdf (typed by me, not a book)
Chapter 2 - Algorithmic Analysis.pdf (typed by me)
https://www.youtube.com/watch?v=c_wUBeeJV9U

1. Motivation behind Time Complexity
2. Orders of Growth
3. Calculating Time Complexities
    a. Loops
    b. Phases
    c. Multiple Parameters
    d. Recursion
4. Thumb Rules
    a. Guessing the approximate time complexity based on the constraints
    b. Using integration as an approximation of addition
    c. Sum of 1/primes
5. Case Study: Kadane's Algorithm (Maximum Sum Subarray Problem)

Topic 2: Arrays - final

Data Structures.pdf
Algorithms.pdf

1. Sorting
    a. O(n^2) algorithms
        i. Bubble Sort
        ii. Selection Sort
        iii. Insertion Sort
    b. O(n*logn) algorithms
        i. Merge Sort
        ii. Quick Sort
    c. O(n) algorithms
        i. Counting Sort
        ii. Bucket Sort
    d. Variations of Merge Sort
        i. Inversion Count

e. Variations of Partitioning in Quicksort
    i. Rearranging the numbers of array to get alternate positive-negative pattern
    ii. Rearranging the numbers of array to get alternate odd-even pattern
    iii. Pushing all zeros to the end
    iv. Dutch National Flag Problem
    v. Finding Median in O(n) time
f. Using sort() function of STL
    i. Writing a custom compare function to sort strings based on the length
    ii. Writing a custom compare function to sort using multiple keys. Example, strings of the same length should be sorted lexicographically
    iii. Achieving stable sorting using STL's sort() function

2. Searching
    a. Binary Search
        i. To find the element in the sorted array
        ii. To find the lower and upper bounds of an element in the sorted array
    b. Variations of Binary Search
        i. Finding a peak element in 1D and 2D arrays
        ii. Given two sorted arrays, find the median of the merged array
        iii. Aggressive Cows Problem
        iv. Painter's Partition Problem

3. Rotations
    a. Array rotation
        i. Rotate the array using reversal algorithm
        ii. Print the rotated array without actually rotating the array
    b. Search in sorted and rotated array
        i. Find the pivot element in the sorted & rotated array
        ii. Binary search over sorted & rotated array

4. Sliding Window Technique
   https://www.youtube.com/playlist?list=PL_z_8CaSLPWeM8BDJmIYDaoQ5zuwyxnfj
    a. Fixed-sized window problems
        i. Maximum sum subarray of size 'k'
        ii. Maximum element in every subarray of size 'k'
        iii. First negative element in every subarray of size 'k'

iv. Count occurrences of an anagram of a pattern in a string

b. Variable-sized window problems
    i. Largest subarray with sum ‘k’
    ii. Largest substring with ‘k’ distinct characters
    iii. Largest substring with no repeating characters
    iv. Minimum window substring

5. Difference & Prefix Sum Arrays (CP Topic)
   https://blogarithms.github.io/articles/2018-11/difference-arrays
   https://wcipeg.com/wiki/Prefix_sum_array_and_difference_array
    a. Basics
        i. What is a difference array & prefix sum array
        ii. Analogy with calculus
        iii. How to use difference array & prefix sum array
        iv. Extending this concept to multiple dimensions
    b. Difference array problems
        i. CCC 2009 - Wireless
        ii. CCC 2011 - The cake is a dessert
        iii. 295A - Greg and Array
        iv. 816B - Karen and Coffee
        v. 276C - Little Girl & Maximum Sum
        vi. 1343D - Constant Palindrome Sum
    c. Prefix sum problems
        i. Leetcode - Trapping Rain Water
        ii. SPOJ SUBSEQ - Counting Subsequences
        iii. 877B - Nikita and String
        iv. 835C - Star sky
        v. 846D - Monitor

Topic 3: Linked Lists - almost finalized

Data Structures.pdf

1. Basic Operations
    a. Insertion of a node in SLL
        i. In the beginning
        ii. Somewhere in between

        iii. At the end
    b. Deletion of a node in SLL
        i. The head node
        ii. Node somewhere in between
        iii. The last node
    c. Traversing the SLL
        i. Get the length of SLL
        ii. Search an element in SLL
        iii. Left rotate the SLL by ‘k’ nodes
        iv. Reverse the SLL
        v. Reverse the SLL in blocks of size ‘k’
        vi. Check if the given SLL is a palindrome
        vii. Bring all even nodes before odd nodes
        viii. Push all zeros to the end

2. Loops in SLL
        i. Detect Loop
        ii. Count number of nodes in a loop
        iii. Remove loop

3. Sorting SLL
    a. Insertion Sort
    b. Merge Sort

4. Interesting Problems
        i. Find the intersection node when two SLLs intersect
        ii. Add two numbers represented in the form of SLLs
        iii. Given a DLL with one extra random pointer, clone this list

5. Circularly linked lists
        i. Insertion in CLL
        ii. Deletion in CLL
        iii. Traversing CLL
        iv. Insertion Sort in CLL

6. Doubly linked lists
        i. Insertion in DLL

ii. Deletion in DLL
iii. Traversing DLL
iv. Merge Sort DLL
v. Memory Efficient DLL

Topic 4: Stacks - final

Data Structures.pdf

1. Implementing Stacks
    a. Fixed size stack
    b. Dynamic stack using Table-Doubling
    c. In-built stack in STL

2. Design Problems in Stacks
    a. Implement two stacks in a single array
    b. Implement a stack that also gives minimum element in O(1) time
    c. Implement a stack that returns & deletes the middle element in O(1) time

3. Applications of Stacks
    a. Check if parentheses are balanced
    b. Interconversion & Evaluation of Infix, Prefix and Postfix notations

4. Stack problems (CP Topic)
    a. Stock Span Problem
    b. Maximum area under histogram
    c. SPOJ ANARC09A - Seinfeld
    d. SPOJ STPAR - Street Parade

Topic 5: Queues - final

Data Structures.pdf

1. Implementing Queues
    a. Fixed size queue
    b. Dynamic stack using Table-Doubling
    c. In-built queue in STL

2. Design Problems in Queues
    a. Implement a stack using queues
    b. Implement a queue using stacks

3. Queue Problems
    a. Given an array of non-negative integers, find the largest multiple of 3 that can be formed from array elements
    b. Given an integer N, find the least possible integer made up of only digits 9 and 0 such that it is divisible by N

4. Doubly ended queue
    a. C++ Implementation https://stackoverflow.com/questions/6292332/what-really-is-a-deque-in-stl
    b. Minimum Stack/ Minimum Queue https://cp-algorithms.com/data_structures/stack_queue_modification.html

Topic 6: Trees - final

Data Structures.pdf

1. Basics
    a. Structure of a binary tree
    b. Types of BT
        i. Full/Strict Binary Tree
        ii. Complete Binary Tree
        iii. Perfect Binary Tree
        iv. Skewed Binary Tree
    c. Recursive Codes
        i. Sum of all nodes in BT
        ii. Height of BT
        iii. Find the maximum element in BT
        iv. Check if a node exists in BT or not
        v. Check if tree is full/strict
        vi. Check if tree is complete
        vii. Check if tree is perfect
        viii. Check if tree is skewed
        ix. Compare if two trees are identical (and similar)

x. Check if a tree is foldable (and symmetric)
xi. Check if two trees are isomorphic
xii. Print boundary nodes of BT
xiii. Print left, right, top & bottom views of BT
xiv. Given a number, create the factor tree

2. Traversals
    a. Breadth First Traversal
        i. Level Order Traversal
    b. Depth First Traversal
        i. Preorder Traversal
        ii. Inorder Traversal
        iii. Postorder Traversal
    c. Level Order Traversal Variations
        i. Insertion of Node
        ii. Deletion of Node
        iii. Find the maximum width of BT
        iv. Print corner nodes at each level
        v. Find the sum of leaf nodes at the minimum level
        vi. Print BT vertically.
        vii. Reverse level order traversal
        viii. Spiral order traversal

3. BT Construction
    a. How many different BTs (and BSTs) are possible with n nodes?
    b. If given two traversals of BT, can we construct a BT uniquely?

4. Least Common Ancestor (LCA)
    a. Find LCA of two nodes in BT
    b. Find distance between two nodes in BT
    c. Find the k'th ancestor of a node
    d. Check if two nodes are cousins

5. Interesting Problems
    a. Traversal based
        i. Given Inorder & Preorder traversals of a binary tree, print postorder traversal

ii. Given a preorder traversal of a full BT as a string of characters ‘l’ (leaf) & ‘n’ (node), find the depth of the BT
iii. Given a BT, find all duplicate subtrees

b. Root to leaf path
i. Print all root to leaf paths
ii. Check if there is a root to leaf path which adds up to a given sum
iii. Find the length of the diameter (longest path between two nodes) of BT
iv. Find maximum possible sum from one leaf to another
v. Find the length of the path having maximum bends

c. LCA based
i. Given two nodes, count the number of turns between them

d. N-ary tree based
i. Given a very large n-ary tree, the root node has some info to pass to all its children. Each node can only pass the information to one child at a time. Find the minimum iterations required to pass info to all nodes of BT

6. Binary Search Trees
a. Basic operations on BST
i. Searching a key in BST
ii. Inserting a key in BST
iii. Deleting a key from BST

b. LCA of BST
i. Find the LCA of two nodes in BST
ii. Find the distance between two nodes in BST

c. Interval Trees
i. Implement a data structure which efficiently performs following operations:
- Add an interval
- Remove an interval
- Given an interval x, find if x overlaps with any of the existing intervals

d. BST Problems
i. Given a BT, check if it is BST or not
ii. Given a BT, return the size of the largest subtree which is a BST
iii. Given a BST, find the k’th smallest element in BST
iv. Given a BST and a value k, find the node with minimum absolute difference with the value k

v. Given two values k1 and k2, print all keys in range k1 and k2

7. Heaps
    a. Implementation
        i. Array based
        ii. Priority Queues from STL
    b. Applications
        i. K'th smallest element
        ii. Heap Sort
        iii. Sorting an almost sorted array
    c. Heap Problems
        i. Given an array, find k numbers with most occurrences i.e. top k numbers with maximum frequency
        ii. Given k sorted arrays of different sizes, merge them into one sorted array
        iii. Find median of the running streams of integers

Topic 7: Graphs - final

Data Structures.pdf
CP Algorithms.pdf
CP Handbook.pdf

1. Basics
    a. Graphs = Nodes + Edges
    b. Types of Graphs
        i. Directed vs Undirected
        ii. Unweighted vs Weighted
        iii. Cyclic vs Acyclic
    c. Representation of Graphs
        i. Adjacency Matrix
        ii. Adjacency List

2. Breadth First Search (BFS)
    a. Generic BFS algorithm
        i. Order in which nodes are being traversed in BFS
        ii. Concept of level and parent array

b. Code
    i. Adjacency List
    ii. Adjacency Matrix
    iii. Grid
c. Variations
    i. Shortest path from source to other vertices in an unweighted graph using parent array
    ii. Number of shortest paths from source to other vertices
    iii. Finding the least number of moves where states can be represented as nodes and transitions can be represented as edges
d. BFS Problems
    i. SPOJ PPATH - Prime Path
    ii. SPOJ NAKANJ - Minimum Knight Moves
    iii. SPOJ DIGOKEYS - Find the treasure
    iv. SPOJ ADACYCLE - Ada and Cycle
    v. SPOJ WATER - Water among cubes

3. Depth First Search (DFS)
    a. Generic DFS algorithm
        i. Order in which nodes are being traversed in DFS
    b. Code
        i. Adjacency List
        ii. Adjacency Matrix
        iii. Grid (Flood Fill Algorithm)
    c. Variations
        i. Find the number of connected components
        ii. Given a DAG, find a topological sorting order for it
        Follow up: Find all topological sorting orders for the given DAG
    d. DFS Problems
        i. Check if the given graph is Bipartite
        ii. SPOJ BUGLIFE - A Bug's Life (BFS way to check Bipartiteness. Recursion gives Stackoverflow.)
        iii. SPOJ UCV2013H - Slick
        iv. SPOJ ABCPATH - ABC Path
        v. SPOJ SERGRID - Grid

4. Connectivity
    a. Path between two nodes - BFS/DFS: anything will work
        i. Check if there is a path between the given two nodes/ Check if the given two nodes are connected
        ii. Return a path between the given two nodes
        iii. Return all possible paths between the given two nodes
    b. Cycles in an undirected graph - DFS
        i. Check if there exists a cycle in the graph
        ii. Return the number of cycles in the graph
        iii. Print vector having lengths of all cycles in the graph. If the graph is acyclic, print -1.
    c. Classification of edges in a directed graph - DFS
        i. Concept of Tree Edge, Back Edge, Forward Edge and Cross Edge
        ii. Entry time and exit time of each node during DFS traversal
    d. Cycles in a directed graph - DFS
        i. Check if there exists a cycle in the graph
        ii. Return the number of cycles in the graph
        iii. Print vector having lengths of all cycles in the graph. If the graph is acyclic, print -1.
    e. Critical Points and Edges - DFS (CP Topic)
        i. Find bridges in an undirected graph
        ii. Find articulation points in an undirected graph
    f. Strongly Connected Components (SCC)
        i. Concept of SCCs
        ii. Kosaraju's algorithm
        iii. Using Kosaraju's algorithm to build condensed graph
    g. Connectivity Problems
        i. SPOJ MAKEMAZE - Validate the maze
        ii. SPOJ ADASEA - Ada and Island
        iii. SPOJ EAGLE1 - Eagle and Dogs
        iv. SPOJ SUBMERGE - Submerging Islands
        v. SPOJ TOUR - Fake Tournament
        vi. SPOJ CAPCITY - Capital City
        vii. SPOJ GOODA - Good Travels

5. Disjoint Set Union (DSU)
    a. DSU - data structure and its applications
       https://cp-algorithms.com/data_structures/disjoint_set_union.html
    b. DSU Problems
        i. SPOJ CHAIN - Strange Food Chain

6. Minimum Spanning Tree (MST)
    a. Building the MST of an undirected graph
        i. Kruskal's Algorithm
        ii. Prim's Algorithm
    b. MST Problems
        i. SPOJ BLINNET - Bytelandian Blingors Network
        ii. SPOJ IITKWPCG - Help the old king
        iii. SPOJ MARYBMW - BMW

7. Shortest Path Algorithms
    a. Single-source shortest paths to all other vertices
        i. Dijkstra's algorithm
        ii. Bellman Ford algorithm
        iii. 0-1 BFS
        iv. Dial's algorithm
    b. All pairs shortest paths
        i. Floyd Warshall algorithm
        ii. Shortest paths of a fixed length
    c. Shortest Path Problems
        i. Minimum number of edges to be reversed to make a path between source and destination nodes
        ii. SPOJ SHPATH - The shortest path
        iii. SPOJ ADATRIP - Ada and Trip
        iv. SPOJ CCHESS - Costly Chess
        v. SPOJ CHICAGO - 106 Miles to Chicago
        vi. SPOJ ARBITRAG - Arbitrage
        vii. SPOJ SOCIALNE - Possible Friends
        viii. SPOJ INGRED - Ingredients

8. Flows
    a. Maximum flow - Ford Fulkerson algorithm
        i. Flow network
        ii. Ford Fulkerson algorithm
        iii. Edmonds Karp algorithm
        iv. Max-flow min-cut theorem
    b. Maximum flow - Push relabel algorithm (optional)
        i. Some definitions
        ii. Algorithm
        iii. Improved version
    c. Maximum flow - Dinic's algorithm (optional)
        i. Some definitions
        ii. Algorithm
    d. Maximum flow - MPM algorithm (optional)
        i. Algorithm
    e. Flow with demands
        i. Finding an arbitrary flow
        ii. Minimal flow
    f. Minimum cost flow
        i. Successive shortest path algorithm
        ii. Solving assignment problem using Min cost flow

Topic 8: Graphs Advanced (Completely Optional) - pending

CP Algorithms.pdf
https://www.youtube.com/playlist?list=PLJ5C_6qdAvBF0v3uOhAeDbuCv-Qq9xKj5

Topic 9: Divide and Conquer - almost finalized

CP Algorithms.pdf
Algorithms.pdf

1. Binary Exponentiation (CP Topic)
    a. Implementation
        i. On Numbers
        ii. On matrices

b. Applications
    i. Computing (x^n) mod m which will be later used in computing modular multiplicative inverse
    ii. Computing fibonacci numbers effectively
    Follow up: Linear Recurrent Sequences
    iii. Number of paths of length k in a graph
    iv. Applying permutation on a sequence k-times
c. Binary Exponentiation Problems
    i. SPOJ FIBOSUM - Fibonacci sum
    ii. SPOJ PERMSG - Permutation Exponentiation
    iii. SPOJ LASTDIG - Last Digit
    iv. SPOJ LOCKER - Magic of the locker
    v. SPOJ ZSUM - Just add it

2. Fast Fourier transform (CP Topic)
    a. Representation of Polynomials
        i. Coefficients
        ii. Roots
        iii. Samples
    b. Operations on Polynomials
        i. Evaluation
        ii. Addition
        iii. Multiplication
    c. Interconversion of Coefficients and Samples forms
        i. Deriving Discrete Fourier Transform using Divide and Conquer Approach
        ii. FFT algorithm
        iii. Inverse DFT
        iv. C++ Implementation
    d. Applications
        i. All possible sums
        ii. All possible scalar products
        iii. String matching
    e. FFT Problems
        i. SPOJ POLYMUL - Polynomial Multiplication
        ii. SPOJ ADAMATCH - Ada and Nucleobase
        iii. SPOJ MAXMATCH - Maximum Self Matching

Topic 10: Greedy Technique - almost finalized

Algorithms.pdf

1. Introduction
    a. Case Study: Change with minimum coins problem
    b. Doesn't produce optimal solution
2. Scheduling Problem
    a. Given 'n' jobs with their starting and ending times, find a schedule that includes as many jobs as possible.
    b. If we associate profit with each job and our task is to maximize profit, it becomes DP problem.
3. Sequencing Problem
    a. Given 'n' jobs with their deadlines and profit associated with it, find the sequence of jobs for maximizing the profit.
    b. Improved implementation using DSU
4. Fractional Knapsack Problem
    a. Given 'n' items with their weight and price, fill the knapsack with a capacity to maximize the value of items collected, if items could be taken in fraction.

Topic 11: Recursion and Backtracking- almost finalized

Algorithms.pdf

1. Recursion
    a. Problem Decomposition and recomposition
        i. Factorial
        ii. Fibonacci
        iii. GCD
        iv. Tower of Hanoi
    b. Recursion Problems
        i. Decimal to Binary conversion
        ii. Binary to Decimal conversion
        iii. SPOJ POUR1 - Pouring Water
        iv. SPOJ SEQ - Recursive sequence

2. Introduction to Backtracking
    a. Exhaustive searching of all possibilities
        i. Idea of efficient brute-force
        ii. General code for backtracking problems - to check if there exists a solution & to print all possible solutions
    b. Some famous backtracking problems
        i. N Queens problem
        ii. Knight's Tour problem
        iii. Sudoku Solving problem

Topic 12: Dynamic Programming - almost finalized

https://www.youtube.com/playlist?list=PL_z_8CaSLPWekqhdCPmFohncHwz8TY2Go

1. Introduction to DP
    a. Case study: Fibonacci numbers (derive T(n) = phi^n)
    b. Top down/Memoization & Bottom up/Tabulation approaches
    c. 2 types of DP problems: Optimization based & Combinatorics based
    d. Variant of Fibonacci problem: The staircase problem

2. Knapsack Problem
    a. 3 types of Knapsack problems
        i. Fractional Knapsack (Greedy)
        ii. 0-1 Knapsack
        iii. Unbounded Knapsack
    b. Variants of Knapsack
        i. Subset sum problem
        ii. Equal sum partition problem
        iii. Count of subsets for given sum
        iv. Minimum subset sum difference
        v. Number of subsets with given difference
    c. Variants of Unbounded Knapsack
        i. Rod cutting problem
        ii. Coin change problem - Number of ways to give change (combinatorial)
        iii. Coin change problem - Minimum coins to give change (optimization)

3. Longest Common Subsequence (LCS)
    a. LCS Problem
        i. Returning the length of the LCS
        ii. Printing the LCS
    b. Variants of LCS
        i. Longest common substring - length + printing the substring
        ii. Shortest common supersequence - length + printing the SCS
        iii. Minimum insertions & deletions to convert a string A to string B
        iv. Longest palindromic subsequence - length + printing the LPS
        v. Minimum insertions or deletions to convert a string A to palindrome
        vi. Longest repeating subsequence - length + printing LRS
        vii. Sequence pattern matching

4. Longest Increasing Subsequence (LIS)
    a. LIS Problem
        i. Returning the length of the LIS
        ii. Printing the LIS
    b. Variants of LIS
        i. Longest Common Increasing subsequence - length + printing LCIS
        ii. Longest Bitonic sequence - length + printing LBS
        iii. Convert array to strictly increasing with minimum replacements

5. Matrix Chain Multiplication
    a. MCM Problem
        i. Minimum number of multiplications required to multiply the matrix chain
        ii. Print the brackets around matrices for minimum multiplications
    b. Variants of MCM
        i. Palindrome Partitioning problem
        ii. Evaluate expression to true/Boolean parenthesization problem
        iii. Scrambled String problem
        iv. Egg drop problem

6. DP problems on grid
    a. Minimum cost path in a grid with 4 directional motion allowed
    b. Number of ways to reach from starting position to ending position by moving in specified directions
    c. Maximum sum submatrix in a 2D matrix

    d. Minimum sum submatrix in a 2D matrix
    e. Largest submatrix with sum divisible by k
    f. Largest submatrix with all 1s in a binary 2D matrix (2 cases: submatrix is square or rectangle)

7. DP problems on strings
    a. Edit distance
    b. Word break problem

8. Other classic DP problems
    a. Weighted Job scheduling
    b. Catalan number & its applications

Topic 13: Working with Bits - pending

https://www.youtube.com/playlist?list=PLX0iyO9CrCF1-4je7G0JMSr_50I0J2K3Z
https://www.youtube.com/playlist?list=PL2q4fbVm1Ik7ip1VkWwe5U_CEb93vw6Iu
https://www.youtube.com/playlist?list=PLb3g_Z8nEv1icFNrtZqByO1CrWVHLlO5g
https://www.youtube.com/watch?v=bjucBkxrMBs
https://www.youtube.com/watch?v=OEthLiejmHk
https://www.youtube.com/watch?v=mkiK_GCWX50

Topic 14: String Processing (CP Topic) - pending

CP Algorithms.pdf

Topic 15: Range Queries (CP Topic) - almost finalized

CP Algorithms.pdf

1. Square Root Decomposition
    a. Sqrt decomposition
    b. Sqrt decomposition Problems
        i. SPOJ DQUERY - D Query
        ii. SPOJ GIVEAWAY - Give away

2. Sparse Table
    a. Sparse Table
    b. Sparse Table Problems
        i. SPOJ RMQSQ - Range Minimum Query
        ii. SPOJ THRBL - Catapult that ball
        iii. SPOJ RPLN - Negative Score

3. Segment Tree
    a. Segment trees - simple and advanced versions
    b. Seg trees Problems
        i. SPOJ KQUERY - K query
        ii. SPOJ GSS1 - Can you answer these queries 1
        iii. SPOJ GSS3 - Can you answer these queries 3
        iv. SPOJ GSS4 - Can you answer these queries 4
        v. SPOJ GSS5 - Can you answer these queries 5

4. Fenwick Tree
    a. Fenwick tree
    b. FT Problems
        i. SPOJ CTRICK - Card Trick
        ii. SPOJ MATSUM - Matrix Sum
        iii. SPOJ YODANESS - Yodaness level
        iv. SPOJ DCEPC705 - Weird Points
        v. SPOJ DCEPC206 - It's a Murder.
        vi. SPOJ SUMSUM - Enjoy sum with operations

Topic 16: Number Theory (CP Topic) - to be modified a little

CP Algorithms.pdf

1. Fibonacci Numbers
    a. Fibonacci Number

2. Greatest Common Divisor (GCD)/ Highest Common Factor (HCF)
    a. Euclid's Algorithm
    b. Extended Euclid's Algorithm
    c. Linear Diophantine Equation

d. Lame's Theorem
   https://www.cut-the-knot.org/blue/LamesTheorem.shtml
e. Euclid Algorithm Problems
   i. SPOJ MAIN74 - Euclid Algorithm revisited

3. Prime Numbers
   a. Sieve of Eratosthenes
   b. Sieve of Eratosthenes having linear time complexity
   c. Euler Totient Function
   d. Number of Divisors/Sum of Divisors
   e. Prime Number Problems
      i. SPOJ TDPRIMES - Printing some primes
      ii. SPOJ HS08PAUL - A conjecture of Paul Erdos
      iii. SPOJ VECTAR8 - Primal Fear
      iv. SPOJ NGIRL - Namit in Trouble
      v. SPOJ DCEPC505 - Bazinga!
      vi. SPOJ BSPRIME - Binary Sequence of Primes
      vii. SPOJ LCMSUM - LCM Sum
      viii. SPOJ GCDEX - GCD Extreme
      ix. SPOJ TIP1 - Totient in Permutation
      x. SPOJ DCEPCA03 - Totient Extreme
      xi. SPOJ INVPHI - Smallest Inverse Totient Function
      xii. SPOJ DIVSUM - Divisor Summation

4. Modular Arithmetic
   a. Introduction, similarities and differences with normal arithmetic
      https://www.cut-the-knot.org/blue/Modulo.shtml
      https://www.cut-the-knot.org/blue/sim_diff.shtml
      https://www.cut-the-knot.org/blue/solutions.shtml
   b. Modular Multiplicative Inverse
   c. Linear Congruence Equation
   d. Chinese Remainder Theorem
      https://www.cut-the-knot.org/blue/chinese.shtml
   e. Factorial modulo p
   f. Discrete Root
   g. Primitive Root
   h. Discrete Logarithm

Topic 17: Algebra and Combinatorics (CP Topic) - pending

CP Algorithms.pdf

Topic 18: Computational Geometry (CP Topic) - pending

CP Algorithms.pdf
cp-geo.pdf