Data Structures and Algorithms for Online Programming Contest

Dynamic Programming

+ [Coin Change and variants](algorithms/coin_change.cpp)

+ [Knapsack Problem and variants](algorithms/knapsack.cpp)

+ [Matrix Chain Multiplication](algorithms/mcm.cpp)

+ [Longest Increasing Subsequence(O(n^2))](algorithms/lis.cpp)

+ [Longest Increasing Subsequence(O(nlogn))](algorithms/lis.cpp)

+ [Travelling Salesman Problem](algorithms/tsp.cpp)

+ [Maximum Sum Subarray(O(n^4) and O(n^3))](algorithms/maximum_sum_subarray.cpp)

+ [Kadane Algorithm](algorithms/kadane.cpp)

+ [Maximum Sum Subarray using Kadane(O(n^3))](algorithms/kadane.cpp)

+ [Optimal Binary Search Tree](algorithms/optimal_search_tree.cpp)

+ [Subset Sum](algorithms/subset_sum.cpp)

+ [Catalan Number](algorithms/CatalanNumber.cpp)

+ [DAG Minimum Path](algorithms/DAG_min_path.cpp)

+ [Minimum Cost Path](algorithms/min_cost_path.cpp)

+ [Digit Dp I](algorithms/Digit_dp_I.cpp)

+ [Digit Dp II](algorithms/Digit_dp_II.cpp)

+ [Digit Dp III](algorithms/Digit_dp_III.cpp)

+ [Digit Dp IV](algorithms/Digit_dp_IV.cpp)

Backtracking

+ [Permutation Generator](algorithms/permutation_generator.cpp)

+ [N-Queen](algorithms/nqueen.cpp)

+ [Prime Ring](algorithms/prime_ring.cpp)

Greedy Algorithm

+ [Huffman Coding](algorithms/Huffman_coding.cpp)

String Algorithm

+ [Aho-Corasick Algorithm](algorithms/Aho_Corasick.cpp)

+ [Knuth-Morris-Pratt's Algorithm](algorithms/kmp.cpp)

+ [Rabin Karp Pattern Searching](algorithms/rabin_karp.cpp)

+ [Z Algorithm](algorithms/z.cpp)

+ [Finite Automata Pattern Searching](algorithms/Finite_Automata_Pattern_Searching.cpp)

+ [Trie (Prefix/Radix Tree)](algorithms/Trie.cpp)

+ [Longest Common Subsequence](algorithms/lcs.cpp)

+ [Edit Distance](algorithms/edit_distance.cpp)

+ [Longest Palindromic Subsequence](algorithms/lps.cpp)

+ [Suffix Array](algorithms/suffix_lcp.cpp)

+ [Longest Common Prefix](algorithms/suffix_lcp.cpp)

+ [Minimum Expression](algorithms/Minimum_expression.cpp)

+ [Suffix Automata](algorithms/suffix_automata.cpp)

Graph Theory

+ [Floyd Warshall's](algorithms/floyd_warshall.cpp)

+ [Loop Detection](algorithms/topsort.cpp)

+ [Topological Sort](algorithms/topsort.cpp)

+ [Strongly Connected Component (Kosaraju)](algorithms/scc.cpp)

+ [Lowest Common Ancestor(sparse table)](algorithms/lca.cpp)

+ [Articulation Point](algorithms/articulation_point.cpp)

+ [Bridge](algorithms/bridge.cpp)

+ [Breadth First Search](algorithms/bfs.cpp)

+ [Dijkstra](algorithms/dijkstra.cpp)

+ [Bellman Ford's](algorithms/bellman_fords.cpp)

+ [Kruskal Minimum Spanning Tree](algorithms/kruskal.cpp)

+ [Minimum Vertex Cover](algorithms/min_vertex_cover.cpp)

+ [Maximum Flow (Edmonds Karp's) I](algorithms/max_flow.cpp)

+ [Maximum Flow (Edmonds Karp's) II](algorithms/Maximum_Flow_Problem_I_Edmond_Karp.cpp)

+ [Maximum Bipartite Matching](algorithms/maximum_bipartite_matching.cpp)

+ [Stable Marriage Problem](algorithms/stable_marriage_problem.cpp)

+ [Heavy Light Decomposition](algorithms/HLD.cpp)

Mathematics

+ [Power Function(Big mod)](algorithms/Power.cpp)

+ [Modular Mutiplicative Inverse(using Big mod)](algorithms/Power.cpp)

+ [Prime(Sieve of Erathonesis)](algorithms/prime.cpp)

+ [Segmented Sieve of Erathonesis](algorithms/prime.cpp)

+ [Prime factorization(using Sieve)](algorithms/prime.cpp)

+ [Prime factorization](algorithms/prime.cpp)

+ [Primality Test(School method)](algorithms/primality_test.cpp)

+ [Miller–Rabin Primality Test](algorithms/primality_test.cpp)

+ [Euler Totient (Phi Function)](algorithms/phi.cpp)

+ [Extended Euclid](algorithms/extended_euclid.cpp)

+ [Linear Diophatine Equation](algorithms/extended_euclid.cpp)

+ [Modular Mutiplicative Inverse(using Extended Euclid)](algorithms/extended_euclid.cpp)

+ [Matrix Exponentiation](algorithms/matrix_exp.xpp)

+ [Floyd Cycle Finding Algorithm](algorithms/floyd_cycle_finding.cpp)

+ [Big Integer](algorithms/big_integer.cpp)

+ [Josephus Recurrence](algorithms/Josephus_Recurrence.cpp)

+ [Fast Fourier Transform](algorithms/FFT.cpp)

Combinotorics

+ [Factorial](algorithms/factorial.cpp)

+ [nCr](algorithms/ncr.cpp)

+ [De-arrangement](algorithms/dearrangement.cpp)

Game Theory

+ [Game Tree(Memorization)](algorithms/game_tree.cpp)

+ [Nim](algorithms/nim.cpp)

+ [Misère Nim](algorithms/nim.cpp)

+ [Nimble Nim](algorithms/nim.cpp)

+ [Poker Nim](algorithms/nim.cpp)

+ [Prime Power Nim](algorithms/nim.cpp)

+ [Spagrue Grundy Problem](algorithms/grundy.cpp)

+ [Grundy Variant: Zero Nim Game](algorithms/grundy.cpp)

+ [Grundy Variant: Coins on Chessboard](algorithms/grundy.cpp)

+ [Green HackenBush(Colon Principle)](algorithms/hackenbush.cpp)

Binary Search

+ [Binary Search](algorithms/binary_search.cpp)

+ [Lower Bound](algorithms/binary_search.cpp)
+ [Upper Bound](algorithms/binary_search.cpp)
+ [Equal Range](algorithms/binary_search.cpp)

Data Structure

+ [Singly Linked List](algorithms/singly_linked_list.cpp)
+ [Doubly Linked List](algorithms/doubly_linked_list.cpp)
+ [Vector](algorithms/vector.cpp)
+ [Stack](algorithms/stack.cpp)
+ [Queue](algorithms/queue.cpp)
+ [List](algorithms/list.cpp)
+ [Hashtable](algorithms/hashtable.cpp)
+ [HashMap](algorithms/hashmap.cpp)
+ [HashSet](algorithms/hashset.cpp)
+ [Union Find(Disjoint Set)](algorithms/union_find.cpp)
+ [Binary Search Tree](algorithms/BST.cpp)
+ [Segment Tree](algorithms/segment_tree.cpp)
+ [Segment Tree (Lazy Propagation)](algorithms/segment_tree_lazy.cpp)
+ [2D Segment Tree (Quad tree)](algorithms/2D_segment_tree.cpp)
+ [Binary Indexed Tree](algorithms/BIT.cpp)
+ [2D Binary Indexed Tree](algorithms/2D_BIT.cpp)
+ [(AVL Tree) Self Balanced BST](algorithms/AVL.cpp)
+ [(Splay Tree) Self Balanced BST](algorithms/splay.cpp)
+ [Ternary Search Tree](algorithms/ternary_search_tree.cpp)
+ [Heap (Min)](algorithms/heap.cpp)

Computational Geometry

+ [Computational Geometry Template](algorithms/computational_geometry_template.cpp)

+ [Convex Hull (Jarvis's Algorithm or Wrapping)](algorithms/convex_hull_jarvis.cpp)

+ [Convex Hull (Graham Scan)](algorithms/convex_hull_graham_scan.cpp)

Hashing

+ [Double Hashing](algorithms/double_hashing.cpp)

+ [String Hashing by Map](algorithms/map_hashing.cpp)

+ [Berenstain String Hashing](algorithms/berenstain_hashing.cpp)

+ [Rolling Hash](algorithms/rabin_karp.cpp)

Sorting

+ [Merge Sort](algorithms/Merge_Sort.cpp)

+ [Quick Sort](algorithms/Quick_Sort.cpp)

+ [Heap Sort](algorithms/heap_sort.cpp)

+ [Bubble Sort](algorithms/Bubble_Sort.cpp)

+ [Insertion Sort](algorithms/Insertion_Sort.cpp)

+ [Selection Sort](algorithms/Selection_Sort.cpp)

+ [Bucket Sort](algorithms/Bucket_Sort.cpp)

+ [Count Sort](algorithms/Count_Sort.cpp)

+ [Radix Sort](algorithms/Radix_Sort.cpp)

+ [Pancake Sort](algorithms/Pancake_Sort.cpp)

Miscellaneous

+ [Template](template.cpp)