Home Answer Spaces Notifications Search Quora Add Question

C++ in Competitive Programming Algorithms in C++ +3

What are the algorithms required to solve all problems (using C++) in any competitive coding contest?

Answer Follow · 1.2k Request 2

16 Answers

Amit Rai, Algorithm Enthusiast, Hindi Poetry Lover
Updated Tue

Mathematics:

(a)Number Theory

1. Prime Number Generation (Sieve, Segmented Sieve)
2. Euler Totient Theorem
3. Fermat's Theorem
4. HCF & LCM (Euclid)
5. Linear Diophantine Equations (Extended Euclid)
6. Modulus Arithmetic (addition,multiplication,subtraction,modular Inverse)
7. Cycle Finding (Floyd Algo and Brent Algo)
8. Integer Factorization (Trial Division , Pollard Rho method)
9. Lucas Theorem (Simple & Advance)
10. Chinese Remainder Theorem
11. Wilson Theorem
12. Miller - Rabin Primality Testing
13. Perfect Numbers
14. Goldbach Conjecture

(b)Probability

1. Basic Probability and Conditional Probability
2. Random Variables
3. Probability Generating Functions
4. Expectation
5. Probability Distribution [Binomial, Poisson, Normal,Bernoulli]

(c)Counting

1. Pigeonhole principle
2. Inclusion Exclusion
3. Special Numbers [Stirling,Fibonacci,Catalan, Eulerian, Harmonic, Bernoulli]
4. Polya Counting
5. Burnside lemma

(d)Permutation Cycles

(e)Linear Algebra

1. Addition And Su

Upvote · 5.6k Share · 118

Transformations Using Matrix]

4. Determinant , Rank and Inverse Of Matrix [Gaussian Elimination , Gauss Jordan Elimination]
5. Solving System Of Linear Equations
6. Matrix Exponentiation To Solve Recurrences
7. Eigenvalues And Eigen vector
8. Roots of a polynomial [Prime factorization of a polynomial, Integer roots of a polynomial]
9. Lagrange Interpolation

(e)Game Theory

1. Basic Concepts & Nim Game [Grundy Theorem , Grundy Number]
2. Hackenbush

(f)Group Theory

1. Burnside Lemma
2. Polya's Theorem

Graphs:

(a)Graph Representation

1. Adjacency Matrix
2. Adjacency List
3. Incidence Matrix
4. Edge List

(b)Graph Types

1. Directed
2. Undirected
3. Weighted
4. Unweighted
5. Planar
6. Hamilton
7. Euler
8. Special Graphs

(c)DFS & It’s Application

1. Cycle Detection
2. Articulation Points
3. Bridges
4. Strongly Connected Component
5. Connected Component
6. Path Finding
7. Solving Maze
8. Biconnectivity ir

Upvote · 5.6k Share · 118

11. Planarity Testing
12. Flood-fill algorithm

(d)BFS & It's Application

1. Shortest Path (No. Of Edges)
2. Bipartite Checking
3. Connected Components

(d)Minimum Spanning Tree

1. Prim's Algorithm
2. Kruskal Algorithm

(d)Single Source Shortest-Path

1. Dijkstra
2. Bellman Ford

(e)All pair Shortest Path

1. Floyd Warshall's Algorithm

(f)Euler Tour

(g)Flow

1. Ford-Fulkerson [PFS,DFS,BFS]
2. Dinic's Algorithm
3. Min Cost - Max Flow [Successive Shortest Path Algo,Cycle Cancelling Algorithm]
4. Max Weighted BPM [Kuhn Munkres algorithm/Hungarian Method]
5. Stoer Wagner Min-Cut Algo
6. Hop-Kraft BPM
7. Edmond Blossom Shrinking Algorithm

(h)Other Important Topics On Graphs

1. 2-SAT,
2. LCA
3. Maximum Cardinality Matching
4. Application Flow
5. Min Path Cover Over Dag
6. Independent Edge Disjoint Path
7. Minimum Vertex Cover
8. Maximum Independent Set

Data Structures:

1. Arrays
2. Linked List
3. Trees (Binary Tree And Binary Search Tree)
4. Stacks

Related Questions

How much time it takes to be good in competitive programming (to rank less than 100 in Codechef short contest for example)?

What are the 10 algorithms one must know in order to solve most algorithm problems?

What are the top 30 most essential algorithms you must know for competitive programming?

What is the best strategy to improve my skills in competitive programming in C++ in 2-3 months?

Why do competitive programmers prefer to use C++ instead of Java in the programming contests?

What are the most important algorithms used in competitive programming? Please provide link if you can?

Ask Question · More Related Questions

Upvote · 5.6k Share · 118

7. Hash Tables
8. Disjoint-Set Data Structures
9. Trie
10. Segment Tree
11. Binary Index Tree
12. Treap

Searching And Sorting:

1. Linear Search
2. BInary Search
3. Ternary Search
4. Selection Sort
5. Bubble Sort
6. Insertion Sort
7. Merge Sort
8. Quick Sort
9. Quick Select
10. Heap Sort
11. Radix Sort
12. Counting Sort

Greedy:
Classical Problems of Greedy & Concept
example : Fractional Knapsack

Dynamic Programming Classical Problems

1. Edit Distance
2. Egg Dropping Puzzle
3. Integer Knapsack
4. Largest Independent Set
5. Longest Biotonic Subsequence
6. Longest Common Subsequence
7. Longest Common Substring
8. Longest Increasing Subsequence
9. Longest Palindromic Subsequence
10. Longest Palindromic Substring
11. Longest Substring Without Repeating Character
12. Matrix Chain Multiplication
13. Max Size Square Submatrix With One
14. Maximum Length Chain Pairs
15. Maximum Sum Increasing Subsequence
16. Optimal Binary Search Tree
17. Palindrome Part

20. Word Wrap Problem

Dynamic Programming Advanced Techniques

1. DP + Tree
2. DP + Bit Masking
3. DP + Binary Search
4. DP + Graph
5. DP + Matrix Exponentiation
6. DP + Probability Space
7. DP + Crack Recurrence

Divide & Conquer
Classical Problems & Concepts

1. Merge Sort
2. Closest Pair Points

Other Algorithm Design Techniques :

1. BackTracking
2. Man In Middle
3. Newton-Raphson to reach the fixed point
4. Brute Force
5. Constructive Algo
6. Sliding Window
7. Pancake Sorting

CP Resources:

- Algorithm Tutorials
- Data Structures and Algorithms
- Modular Multiplicative Inverse
- Heavy Light Decomposition
- Amey Dharwadker's post in Programming Contests
- Visualizing data structures and algorithms through animation
- Coding contest trick: Meet in the middle
- Introduction to Dynamic Programming
- Dynamic Programming Practice Problems
- Solving Linear Recurrence for Programming Contest
- Lalit Kundu's post in Threads @ IIIT Hyderabad
- Binary Indexed Trees with some Solved Example
- Dynamic programming problems with Bitmasking
- Let us code: Segment Trees
- Sai Kiran's post in How do I become a Googler
- https://cses.fi/book/book.pdf
- An awesome lis

Upvote · 5.6k Share · 118

- C++ Programming Tutorial
- Containers - C++ Reference
- Standard Template Library Programmer's Guide

177.8k views · View Upvoters · View Sharers

Gabriel B Nongsiej
I don't know why this answer was not voted more. A great resource. The ultimate list.

1 more comment from Balajiganapathi Senthilnathan

Thanh Trung Nguyen, Know some algorithms
Answered Jul 27, 2015 · Upvoted by Bhavik Dhandhalya, Hacker, Problem Setter on SPOJ & HackerEarth, ACM ICPC 2016 Participant

Before asking this question, you must understand this: In competitive programming, less than 10% of algorithms can be used to solve more than 90% of problems.

If you are really good in basic algorithms (DP, graph, data structures (stack, queue, heap, segment tree, fenwick tree), basic math), you can make it to red on Codeforces/Topcoder. You should learn more only if you are really interested in algorithm, in which case I don't think you need to ask this question in the first place.

Mastering each algorithm takes time. I'm in CP for around 8 years, red on Codeforces and still don't unders...(more)

Upvote · 401 Share · 1

Add a comment... Recommended All

Related Spaces (More Answers Below)

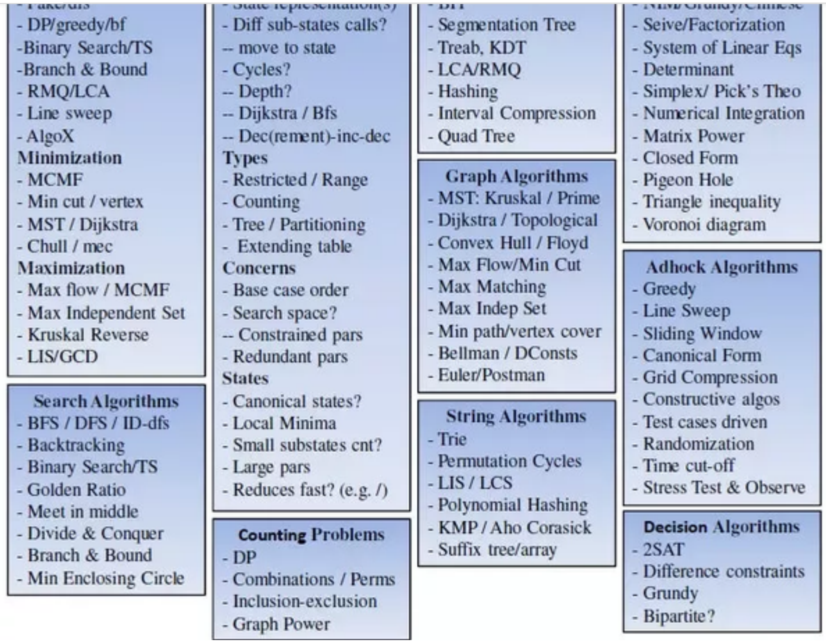
competitive programming
coding and cp blogs

Follow 105

Discover More Spaces >

Mostafa Saad Ibrahim, ACM ICPC World Finalist, Problem Setter, Coach
Answered Mar 22, 2014 · Upvoted by Nikhil Garg, ACM-ICPC World Finalist (2011, 2012) and Abhishek Pratap, ECE undergrad ,IIT-R

Upvote · 5.6k Share · 118



- Fake/dfs
- DP/greedy/bf
-Binary Search/TS
-Branch & Bound
- RMQ/LCA
- Line sweep
- AlgoX

Minimization
- MCMF
- Min cut / vertex
- MST / Dijkstra
- Chull / mec

Maximization
- Max flow / MCMF
- Max Independent Set
- Kruskal Reverse
- LIS/GCD

Search Algorithms
- BFS / DFS / ID-dfs
- Backtracking
- Binary Search/TS
- Golden Ratio
- Meet in middle
- Divide & Conquer
- Branch & Bound
- Min Enclosing Circle

- State representation(s)
- Diff sub-states calls?
-- move to state
- Cycles?
-- Depth?
-- Dijkstra / Bfs
-- Dec(rement)-inc-dec

Types
- Restricted / Range
- Counting
- Tree / Partitioning
- Extending table

Concerns
- Base case order
- Search space?
-- Constrained pars
- Redundant pars

States
- Canonical states?
- Local Minima
- Small substates cnt?
- Large pars
- Reduces fast? (e.g. /)

Counting Problems
- DP
- Combinations / Perms
- Inclusion-exclusion
- Graph Power

- BIT
- Segmentation Tree
- Treab, KDT
- LCA/RMQ
- Hashing
- Interval Compression
- Quad Tree

Graph Algorithms
- MST: Kruskal / Prime
- Dijkstra / Topological
- Convex Hull / Floyd
- Max Flow/Min Cut
- Max Matching
- Max Indep Set
- Min path/vertex cover
- Bellman / DConsts
- Euler/Postman

String Algorithms
- Trie
- Permutation Cycles
- LIS / LCS
- Polynomial Hashing
- KMP / Aho Corasick
- Suffix tree/array

- NIM/Grundy/Chinese
- Seive/Factorization
- System of Linear Eqs
- Determinant
- Simplex/ Pick's Theo
- Numerical Integration
- Matrix Power
- Closed Form
- Pigeon Hole
- Triangle inequality
- Voronoi diagram

Adhock Algorithms
- Greedy
- Line Sweep
- Sliding Window
- Canonical Form
- Grid Compression
- Constructive algos
- Test cases driven
- Randomization
- Time cut-off
- Stress Test & Observe

Decision Algorithms
- 2SAT
- Difference constraints
- Grundy
- Bipartite?

Here is my helper list. It lists most of needed algorithms/concepts.Some elements are not algorithms (e.g. Fake, States/Concerns) and little repetitions.

But 1 final advice:

Initially, Given great attention to thinking skills rather than the knowledge. This is helpful for both competitions and your future. To do so, make sure you are so good in adhocks, where no algorithms are required, just pure thinking.

One way to do so, is to focus with TC Div2 and CodeForces Div2. In each one, move in problem level by level. E.g. Master Div2-250, then move to Div2-500

320.1k views · View Upvoters · View Sharers · Answer requested by Ayan Some

Upvote · 3.1k Share · 17

Gabriele Renzi
"adhock" is probably "ad hoc" ?

Related Questions

How much time it takes to be good in competitive programming (to rank less than 100 in Codechef short contest for example)?

What are the 10 algorithms one must know in order to solve most algorithm problems?

What are the top 30 most essential algorithms you must know for competitive programming?

Related Questions

How much time it takes to be good in competitive programming (to rank less than 100 in Codechef short contest for example)?

What are the 10 algorithms one must know in order to solve most algorithm problems?

Upvote · 5.6k Share · 118

What is the best strategy to improve my skills in competitive programming in C++ in 2-3 months?

Why do competitive programmers prefer to use C++ instead of Java in the programming contests?

What are the most important algorithms used in competitive programming? Please provide link if you can?

How do I improve in competitive programming? Solved about 50 to 60 practice problems on codechef but managed to solve only 1 problem in last c...

I am not able to solve problems even after I have practiced quite a bit. I know the basic algorithms and data structures but cannot think when...

How can I learn to think faster while solving algorithmic/coding problems? How can I better visualize and solve the coding problems?

What's the toughest competitive programming problem you ever solved?

What are the best coding contests/competitions for teens?

What are some algorithms and data structures every competitive programmer should know?

Which is the best Android app for learning algorithms/tricks required for competitive coding?

How can I learn competitive coding?

What algorithms are most used to deal with permutation and combination problems in competitive programming?

Upvote · 5.6k Share · 118 ooo