ICPC Training 200 Problems By Topic

Trí tuệ nhân tạo (Trường Đại học Cần Thơ)

ICPC TRAINING PROBLEM SET – 200 PROBLEMS BY TOPIC

This collection contains 200 practice problems grouped by topic. Each entry includes: Title, Source, Difficulty, Topics, and Short Notes.

Note: This first version contains well-known contest problems by title and source. If you need direct clickable links for specific problems, tell me which ones and I will add links on demand.

Sorting & Binary Search

1. Sort the Array
Source: Codeforces 451B
Difficulty: Medium
Topics: Sorting, Implementation
Notes: Check if the array can be sorted by reversing one subsegment.

2. Restaurant Customers
Source: CSES
Difficulty: Easy
Topics: Sorting, Sweep Line
Notes: Sort by time, sweep to find max concurrent customers.

3. Aggressive Cows
Source: SPOJ - AGGRCOW
Difficulty: Medium
Topics: Binary Search on Answer
Notes: Binary search maximum minimum distance between cows.

4. Wood Cutting (EKO)
Source: SPOJ - EKO
Difficulty: Medium
Topics: Binary Search on Answer
Notes: Binary search on height of saw.

5. K-th Number
Source: Codeforces
Difficulty: Medium
Topics: Order Statistics, Binary Search
Notes: Binary search plus prefix counting or persistent segment tree.

6. Collecting Coins (variant)
Source: AtCoder / CF variant
Difficulty: Medium
Topics: Sorting, Greedy
Notes: Sort and greedy selection.

7. Minimize Maximum Pair Sum
Source: CF / custom
Difficulty: Medium
Topics: Sorting, Two Pointers
Notes: Sort and pair smallest with largest.

8. Two Pointers - Subarray Sum
Source: Classic
Difficulty: Easy-Medium
Topics: Two Pointers, Sliding Window
Notes: Use sliding window for positive arrays.

9. Median Maintenance
Source: Classic
Difficulty: Medium
Topics: Heaps, Order Statistics
Notes: Maintain two heaps to produce medians online.

10. Closest Pair (1D)
Source: Classic
Difficulty: Easy
Topics: Sorting, Two Pointers
Notes: Sort and check neighbors.

11. Pair of Topics
Source: Codeforces 1324C
Difficulty: Medium
Topics: Two Pointers, Prefix Sums
Notes: Partition and count pairs by criteria.

12. Array and Operations (CF)
Source: Codeforces variant
Difficulty: Medium
Topics: Sorting, Greedy

Notes: Sort and pair with operations.

13. Schedule Optimization
Source: AtCoder
Difficulty: Medium
Topics: Sorting, Greedy
Notes: Sort by end-time for interval scheduling.

14. Minimize Sum of Differences
Source: Classic
Difficulty: Medium
Topics: Sorting, Greedy
Notes: Sort arrays and pair correspondingly.

15. Binary Search on Monotone Function
Source: Generic
Difficulty: Medium
Topics: Binary Search on Answer
Notes: Template problem to practice predicate-based binary search.

16. Search in Rotated Array
Source: LeetCode / CF
Difficulty: Easy-Medium
Topics: Binary Search Variants
Notes: Modified binary search on pivoted array.

17. Longest Subarray with Sum ≤ K
Source: Classic
Difficulty: Medium
Topics: Two Pointers, Sliding Window
Notes: Use two pointers for non-negative arrays.

18. Maximize Minimum Distance
Source: CF/STO
Difficulty: Medium
Topics: Binary Search on Answer
Notes: Place items maximizing min distance.

19. Minimize Maximum Workload
Source: Partition problems
Difficulty: Medium
Topics: Binary Search on Answer
Notes: Binary search the maximum allowed and check feasibility.

20. Serega and Fun
Source: CF-like
Difficulty: Medium
Topics: Two Pointers, Greedy
Notes: Pairing and greedy strategy.

Data Structures

1. Fenwick Tree - Range Sum Queries
Source: CSES / Classic
Difficulty: Easy-Medium
Topics: Fenwick Tree, BIT
Notes: Update point, query prefix sums.

2. Range Update Range Query (Fenwick)
Source: Classic
Difficulty: Medium
Topics: Fenwick Tree, Difference Array
Notes: Use two BITs to support range add and range sum.

3. Segment Tree - Point Update Range Query
Source: Classic
Difficulty: Easy-Medium
Topics: Segment Tree
Notes: Basic point update segment tree.

4. Segment Tree - Range Update Lazy
Source: CSES/CF
Difficulty: Medium
Topics: Segment Tree, Lazy Propagation
Notes: Range add and range sum with lazy.

5. Persistent Segment Tree - K-th Number
Source: Classic/CF
Difficulty: Hard

Topics: Persistent DS, Segment Tree
Notes: Use persistent segment tree for offline kth queries.

6. Order Statistics Tree (PBDS)
Source: G++ PBDS examples
Difficulty: Medium
Topics: Order Statistic Tree
Notes: Use policy-based data structure for order statistics.

7. Merge Sort Tree
Source: CF variants
Difficulty: Hard
Topics: Segment Tree, Merge Sort Tree
Notes: Store sorted vectors at nodes for offline queries.

8. Treap - Balanced BST
Source: Classic
Difficulty: Medium
Topics: Treap, BST
Notes: Implicit treap for sequence operations.

9. Sqrt Decomposition - Range Query
Source: Classic
Difficulty: Medium
Topics: Sqrt Decomposition
Notes: Block decomposition for queries and updates.

10. Union-Find (DSU) Basic
Source: Classic
Difficulty: Easy
Topics: DSU
Notes: Union by rank and path compression.

11. DSU with rollback
Source: CF/Offline
Difficulty: Hard
Topics: DSU, Offline Queries
Notes: Support persistent union-find for divide and conquer.

12. Sparse Table - RMQ
Source: CSES/Classic
Difficulty: Easy
Topics: Sparse Table, RMQ
Notes: Static array, idempotent operations.

13. Binary Indexed Tree 2D
Source: Advanced
Difficulty: Hard
Topics: BIT 2D
Notes: 2D Fenwick for point update rectangle sum.

14. Range Minimum Query - Segment Tree
Source: Classic
Difficulty: Easy-Medium
Topics: Segment Tree, RMQ
Notes: Static RMQ using segment tree or sparse table.

15. Dynamic Order Statistics
Source: CF PBDS
Difficulty: Medium
Topics: Ordered Set
Notes: Kth element, order_of_key operations.

16. Min Segment Tree with Index
Source: CF
Difficulty: Medium
Topics: Segment Tree
Notes: Query min value and its index for greedy picks.

17. Interval Tree / Set of Intervals
Source: Classic
Difficulty: Medium
Topics: Sets, Interval Management
Notes: Use set of intervals for merging and queries.

18. Suffix Array Construction (DS heavy)
Source: Strings/Advanced
Difficulty: Hard
Topics: Suffix Array

Notes: Use SA for many string queries with DS support.

19. Priority Queue - Dijkstra Variants
Source: Graph DS
Difficulty: Medium
Topics: Heaps, PQ
Notes: Custom comparator or pair handling.

20. Multiset operations practice
Source: CF exercises
Difficulty: Easy-Medium
Topics: Multiset
Notes: Erase single element, handle duplicates.

21. Segment Tree Beats (advanced)
Source: CF advanced
Difficulty: Hard
Topics: Segment Tree Beats
Notes: Advanced segtree for range chmin/chmax problems.

22. MinQueue / Sliding Window Min (Deque)
Source: Classic
Difficulty: Easy
Topics: Deque, Sliding Window
Notes: Maintain monotonic deque for minima.

23. Disjoint Set for Kruskal
Source: Graph/DS
Difficulty: Easy-Medium
Topics: DSU, MST
Notes: Use for Kruskal MST algorithms.

24. Rope / Implicit Treap for Sequence
Source: Classic
Difficulty: Hard
Topics: Implicit Treap, Rope
Notes: Maintain sequence with splits and merges.

25. Count distinct in range (Mo + DS)
Source: Mo's algorithm
Difficulty: Hard
Topics: Mo's algorithm, Counter DS
Notes: Offline queries using frequency array.

Graph Theory

1. BFS - Shortest Reach
Source: HackerRank/CSES
Difficulty: Easy
Topics: BFS, Graph
Notes: Unweighted shortest path.

2. Dijkstra - Shortest Routes I
Source: CSES
Difficulty: Medium
Topics: Dijkstra, Priority Queue
Notes: Single-source shortest paths with non-negative weights.

3. 0-1 BFS
Source: Classic
Difficulty: Medium
Topics: Deque, BFS
Notes: Edges weights 0 or 1 use deque.

4. Bellman-Ford - Negative Cycles
Source: Classic
Difficulty: Medium
Topics: Bellman-Ford
Notes: Detect negative cycles and shortest paths with negative edges.

5. SPFA (practice)
Source: Classic
Difficulty: Medium
Topics: Shortest Path
Notes: Understand pitfalls and worst-case.

6. Topological Sort
Source: Classic
Difficulty: Easy

Topics: Topo Sort, DAG
Notes: Order nodes with no incoming dependencies.

7. SCC - Kosaraju
Source: Classic
Difficulty: Medium
Topics: SCC, Kosaraju
Notes: Find strongly connected components.

8. SCC - Tarjan
Source: Classic
Difficulty: Medium
Topics: Tarjan, SCC
Notes: Single-pass SCC discovery.

9. Kruskal - MST
Source: Classic
Difficulty: Easy-Medium
Topics: MST, DSU
Notes: Sort edges and union.

10. Prim - MST
Source: Classic
Difficulty: Medium
Topics: MST, Priority Queue
Notes: Use priority queue for dense graphs.

11. Lowest Common Ancestor (binary lifting)
Source: Classic/CF
Difficulty: Medium
Topics: Tree, LCA
Notes: Preprocess with binary lifting for queries.

12. Tree Diameter
Source: Classic
Difficulty: Easy
Topics: Tree, BFS
Notes: Two BFS/DFS technique.

13. Articulation Points and Bridges
Source: Classic
Difficulty: Medium
Topics: DFS tree, Bridges
Notes: Find critical edges/nodes.

14. Bipartite Check and Coloring
Source: Classic
Difficulty: Easy
Topics: Bipartite Graph
Notes: BFS/DFS coloring technique.

15. Hopcroft–Karp Bipartite Matching
Source: Classic
Difficulty: Hard
Topics: Matching
Notes: Fast bipartite matching implementation.

16. Dinic Max Flow
Source: CF/Classic
Difficulty: Hard
Topics: Max Flow
Notes: Efficient flow algorithm for capacity networks.

17. Min-Cut (via Max Flow)
Source: Classic
Difficulty: Medium
Topics: Min Cut, Max Flow
Notes: Use max flow to derive min cut.

18. Eulerian Path & Circuit
Source: Classic
Difficulty: Medium
Topics: Euler Path
Notes: Check in-degree/out-degree properties.

19. Graph Coloring (k-colorability small)
Source: Classic
Difficulty: Hard
Topics: Graph Coloring, Backtracking

Notes: NP-hard general; small k brute force/backtracking.

20. Shortest Path with Potentials (Johnson)
Source: Advanced
Difficulty: Hard
Topics: Johnson, Reweighting
Notes: All pairs shortest path with Dijkstra + potentials.

21. Topological DP on DAG
Source: CF/Classic
Difficulty: Medium
Topics: DP on DAG
Notes: Compute longest paths in DAG via topo order.

22. Multi-source BFS
Source: Classic
Difficulty: Easy
Topics: BFS, Multi-source
Notes: Push multiple sources into queue initially.

23. Graph with Parity (odd/even paths)
Source: CF
Difficulty: Medium
Topics: Bipartite, BFS
Notes: Model parity with layered graph.

24. Shortest Path with Teleporters (0-1 BFS variant)
Source: CF/AtCoder
Difficulty: Medium
Topics: 0-1 BFS
Notes: Use deque for edges weight 0/1.

25. Constrained Shortest Path (k edges)
Source: Advanced
Difficulty: Hard
Topics: DP + Graph
Notes: State includes number of edges used.

26. Minimum Steiner Tree (approx)
Source: Advanced/ICPC
Difficulty: Hard
Topics: Steiner Tree
Notes: NP-hard; small terminals DP solution.

27. Graph Reduction & Contraction
Source: Advanced
Difficulty: Hard
Topics: Graph Transformations
Notes: Contract components to simplify problem.

28. Graph Embedding / Planarity checks
Source: Advanced
Difficulty: Hard
Topics: Planarity
Notes: Rare in ICPC but useful for special tasks.

29. Flow with lower bounds
Source: Advanced
Difficulty: Hard
Topics: Flow, Circulation
Notes: Transform to feasible flow problem.

30. Maximum Clique (small n via bitmask)
Source: Classic
Difficulty: Hard
Topics: Bitmask, Backtracking
Notes: Meet-in-the-middle or bitmask DP for small n.

31. Minimum Path Cover in DAG
Source: Classic
Difficulty: Hard
Topics: Matching, DAG
Notes: Transform to bipartite matching.

32. Graph isomorphism (practical)
Source: Advanced
Difficulty: Hard
Topics: Graph Hashing

Notes: Use canonical labeling or hashing for small graphs.

33. Chinese Postman / Route Inspection
Source: Classic
Difficulty: Hard
Topics: Eulerian, Matching
Notes: Make all vertices even-degree via pairing odds.

34. Maximum Bipartite Matching Applications
Source: CF/ICPC
Difficulty: Medium
Topics: Matching
Notes: Assignment problems and flow conversions.

Dynamic Programming

1. Fibonacci / Basic DP templates
Source: Classic
Difficulty: Easy
Topics: DP 1D
Notes: Simple iterative DP examples.

2. 0/1 Knapsack
Source: Classic
Difficulty: Easy-Medium
Topics: DP, Knapsack
Notes: DP over weight or value.

3. Unbounded Knapsack
Source: Classic
Difficulty: Medium
Topics: DP
Notes: Allow multiple copies of items.

4. Longest Increasing Subsequence (n log n)
Source: Classic
Difficulty: Medium
Topics: LIS
Notes: Patience sorting technique.

5. Longest Common Subsequence
Source: Classic
Difficulty: Easy-Medium
Topics: LCS
Notes: 2D DP standard.

6. Edit Distance (Levenshtein)
Source: Classic
Difficulty: Medium
Topics: String DP
Notes: Compute min operations to convert strings.

7. Digit DP - Count numbers with properties
Source: CF/Classic
Difficulty: Hard
Topics: Digit DP
Notes: DP on digits with tight/leading flags.

8. DP on Trees - Tree DP template
Source: CF/ICPC
Difficulty: Medium
Topics: Tree DP
Notes: Post-order DP on tree nodes.

9. Bitmask DP - TSP (small n)
Source: Classic
Difficulty: Hard
Topics: Bitmask DP
Notes: DP over subsets for traveling salesman variants.

10. Divide and Conquer DP Optimization
Source: CF/Advanced
Difficulty: Hard
Topics: DP Optimization
Notes: Use when quadrangle inequality holds.

11. Convex Hull Trick - Li Chao Tree
Source: Advanced
Difficulty: Hard
Topics: CHT, Li Chao

Notes: Maintain lines to query min quickly.

12. Knuth Optimization (DP)
Source: Advanced
Difficulty: Hard
Topics: DP Optimization
Notes: Use for certain monotone costs.

13. Longest Palindromic Subsequence
Source: Classic
Difficulty: Medium
Topics: DP, Strings
Notes: 2D DP symmetry.

14. Subset Sum (meet-in-the-middle variant)
Source: Classic
Difficulty: Medium
Topics: Meet-in-the-middle
Notes: Split set and combine sums.

15. Palindrome Partitioning
Source: Classic
Difficulty: Medium
Topics: DP, Strings
Notes: DP with precomputed palindrome table.

16. Counting Paths in DAG
Source: CF/Classic
Difficulty: Medium
Topics: DP on DAG
Notes: Count number of ways using topo order.

17. Matrix Chain Multiplication
Source: Classic
Difficulty: Medium
Topics: DP
Notes: Parenthesization DP.

18. Weighted Interval Scheduling
Source: Classic
Difficulty: Medium
Topics: DP, Binary Search
Notes: Sort intervals and binary search previous non-overlap.

19. Partition Array into K segments (DP + optimization)
Source: Advanced
Difficulty: Hard
Topics: DP + Optimization
Notes: Use divide-and-conquer optimization.

20. String DP - Wildcards Matching
Source: Classic
Difficulty: Medium
Topics: DP, Strings
Notes: Wildcard pattern matching DP.

21. DP with bitset optimization
Source: CF tricks
Difficulty: Medium
Topics: Bitset
Notes: Use C++ bitset for speedups.

22. Probability DP / Expected Value
Source: Advanced
Difficulty: Hard
Topics: DP, Probability
Notes: Compute expectations with DP equations.

23. Longest Path in Tree (DP)
Source: Classic
Difficulty: Medium
Topics: Tree DP
Notes: DP to compute diameter variants.

24. DP on Grid with Obstacles
Source: Classic
Difficulty: Medium
Topics: Grid DP

Notes: Simple 2D DP transitions.

25. Minimum Cost Path (DP or Dijkstra variant)
Source: Classic
Difficulty: Medium
Topics: DP + Graph
Notes: Use Dijkstra for weighted grids.

26. Optimal Binary Search Tree (DP)
Source: Advanced
Difficulty: Hard
Topics: DP, Knuth
Notes: Use DP for expected search cost minimization.

27. DP with monotone queue optimization
Source: Advanced
Difficulty: Hard
Topics: DP Optimization
Notes: Use monotonic queue for convex transitions.

28. Substring DP - Counting distinct substrings with DP
Source: Advanced
Difficulty: Hard
Topics: Strings, DP
Notes: Combine suffix structures and DP.

29. Game DP - Sprague-Grundy basics
Source: Classic/Advanced
Difficulty: Medium
Topics: Game Theory, DP
Notes: Compute Grundy numbers for impartial games.

30. DP with state compression (profile DP)
Source: Advanced
Difficulty: Hard
Topics: Profile DP
Notes: Use bitmask over columns for grid DP.

31. DP on Trees - centroid decomposition based DP
Source: Advanced
Difficulty: Hard
Topics: Centroid Decomposition
Notes: Solve distance/count queries via centroid decomposition.

32. Counting subsequences (DP)
Source: CF/Classic
Difficulty: Medium
Topics: DP, Combinatorics
Notes: DP to count number of subsequences satisfying property.

33. Partition into Palindromes
Source: Classic
Difficulty: Medium
Topics: DP, Strings
Notes: DP with palindrome precompute.

34. Staircase DP - combinatorial counts
Source: Classic
Difficulty: Easy-Medium
Topics: DP, Combinatorics
Notes: Classic tiling/counting dynamic programming.

35. DP on DAG with costs and constraints
Source: Advanced
Difficulty: Hard
Topics: DP, Graph
Notes: State includes accumulated cost.

36. Weighted Matching on Tree (DP)
Source: Classic
Difficulty: Hard
Topics: Tree DP
Notes: Matchings weighted via DP.

37. Optimal Strategy for Game with DP and minimax
Source: Classic
Difficulty: Hard
Topics: DP, Game Theory

Notes: Use minimax with DP caching.

38. Resource Allocation DP (knapsack variants)
Source: Classic
Difficulty: Medium
Topics: DP
Notes: Multiple constraints optimization.

39. DP with sparse transitions (graphical DP)
Source: Advanced
Difficulty: Hard
Topics: DP, Graph
Notes: Optimize heavy transitions with data structures.

40. Subsequence Automaton DP problems
Source: Advanced
Difficulty: Hard
Topics: Automaton, DP
Notes: Combine automaton transitions with DP.

41. DP practice - Codeforces gym problems compilation
Source: CF gym
Difficulty: Varied
Topics: All DP types
Notes: Mixed set of DP challenges.

Greedy

1. Interval Scheduling (Maximum non-overlapping intervals)
Source: Classic
Difficulty: Easy
Topics: Greedy, Sorting
Notes: Sort by end time and pick greedily.

2. Activity Selection variations
Source: Classic
Difficulty: Easy-Medium
Topics: Greedy
Notes: Greedy reasoning on intervals.

3. Huffman Coding (Greedy)
Source: Classic
Difficulty: Medium
Topics: Greedy, Heaps
Notes: Build optimal prefix code tree by combining smallest weights.

4. Assigning Workstations (Greedy)
Source: CF/ICPC variants
Difficulty: Medium
Topics: Greedy, Multiset
Notes: Use multiset to assign resources.

5. Coin Change - Greedy correctness cases
Source: Classic
Difficulty: Medium
Topics: Greedy, Math
Notes: Recognize canonical coin systems where greedy works.

6. Scheduling with Deadlines and Penalties (Greedy)
Source: Classic
Difficulty: Medium
Topics: Greedy, DSU/Heap
Notes: Use greedy with priority queue for max profit scheduling.

7. Greedy for Minimizing Maximum Difference
Source: CF variants
Difficulty: Medium
Topics: Greedy
Notes: Pairing techniques to minimize extremes.

8. Greedy with Proofs - Exchange argument practice
Source: Advanced
Difficulty: Hard
Topics: Greedy
Notes: Problems requiring formal greedy proof.

9. Greedy for Trees - select k nodes
Source: CF variants
Difficulty: Medium
Topics: Tree, Greedy

Notes: Choose nodes based on contribution heuristics.

10. Greedy with Sorting + Two Pointers
Source: Classic
Difficulty: Easy
Topics: Greedy, Two Pointers
Notes: Combine methods for efficient solutions.

11. Minimize number of operations (Greedy)
Source: CF
Difficulty: Medium
Topics: Greedy, Implementation
Notes: Greedy design for operation minimization.

12. Greedy for string rearrangement
Source: CF
Difficulty: Medium
Topics: Greedy, Strings
Notes: Reorder characters to meet constraints.

13. Greedy for resource allocation
Source: Classic
Difficulty: Medium
Topics: Greedy, Heaps
Notes: Allocate resources optimally via greedy heuristics.

14. Greedy for range covering problems
Source: Classic
Difficulty: Medium
Topics: Greedy, Interval Covering
Notes: Choose intervals to cover segments with minimal number.

15. Greedy - Maximizing sum under constraints
Source: CF
Difficulty: Medium
Topics: Greedy
Notes: Use greedy pick with proof.

Math & Combinatorics

1. Modular Exponentiation (binary exponentiation)
Source: Classic
Difficulty: Easy
Topics: Modular Arithmetic
Notes: Fast pow under modulus.

2. GCD and Extended GCD
Source: Classic
Difficulty: Easy
Topics: Number Theory
Notes: Compute x,y for ax+by=gcd(a,b).

3. Modular Inverse and Fermat's Little Theorem
Source: Classic
Difficulty: Medium
Topics: Number Theory
Notes: Inverse modulo prime using pow(a, p-2, p).

4. Sieve of Eratosthenes & primes
Source: Classic
Difficulty: Easy
Topics: Primes, Sieve
Notes: Generate primes up to n efficiently.

5. Chinese Remainder Theorem (CRT)
Source: Classic
Difficulty: Hard
Topics: CRT
Notes: Solve system of congruences.

6. Combinatorics - nCr with mod
Source: Classic
Difficulty: Medium
Topics: Combinatorics
Notes: Precompute factorials and inverses.

7. Pollard Rho (factorization)
Source: Advanced
Difficulty: Hard

Topics: Factorization
Notes: Randomized factorization for large numbers.

8. Primality testing (Miller-Rabin)
Source: Advanced
Difficulty: Hard
Topics: Primality
Notes: Deterministic bases for 64-bit integers.

9. Number of divisors / sum of divisors
Source: Classic
Difficulty: Medium
Topics: Number Theory
Notes: Use prime factorization.

10. Lucas Theorem (nCr mod p)
Source: Advanced
Difficulty: Hard
Topics: Combinatorics
Notes: Use for prime mod p with large n.

11. Inclusion-Exclusion Principle
Source: Classic
Difficulty: Medium
Topics: Combinatorics, Counting
Notes: Count with overlapping conditions.

12. Counting lattice paths (Catalan-related)
Source: Classic
Difficulty: Medium
Topics: Combinatorics
Notes: Binomial-counting with constraints.

13. Discrete logarithm (Baby-step Giant-step)
Source: Advanced
Difficulty: Hard
Topics: Number Theory
Notes: Solve a^x ≡ b (mod m) in O(sqrt(m)).

14. Farey sequence / rational approximations
Source: Advanced
Difficulty: Hard
Topics: Number Theory
Notes: Useful for geometry rational approximations.

15. Modular linear equations and systems
Source: Classic
Difficulty: Medium
Topics: Number Theory
Notes: Solve linear congruences.

16. Quadratic residues and Legendre symbol basics
Source: Advanced
Difficulty: Hard
Topics: Number Theory
Notes: Advanced number theory tools.

17. Combinatorial game counts (Catalan, Stirling basics)
Source: Classic
Difficulty: Medium
Topics: Combinatorics
Notes: Use combinatorial identities.

18. Fast convolution via FFT (math foundations)
Source: Advanced
Difficulty: Hard
Topics: FFT
Notes: Use FFT for polynomial multiplication.

19. Applications of modular arithmetic in counting problems
Source: Classic
Difficulty: Medium
Topics: Number Theory
Notes: Mod arithmetic to keep numbers manageable.

20. Chinese Postman math variant (pairing odds)
Source: Classic
Difficulty: Hard
Topics: Graph Math

Notes: Pair odd-degree vertices optimally.

Strings

1. KMP - Pattern Matching
Source: Classic
Difficulty: Medium
Topics: KMP, Strings
Notes: Find occurrences efficiently.

2. Z-function and applications
Source: Classic
Difficulty: Medium
Topics: Z-function
Notes: Alternative to KMP for some tasks.

3. Rabin-Karp / Rolling Hash
Source: Classic
Difficulty: Medium
Topics: Hashing, Strings
Notes: String hashing with mod and base.

4. Aho-Corasick - multiple patterns
Source: Classic
Difficulty: Hard
Topics: Automaton, Strings
Notes: Multiple pattern matching in linear time.

5. Suffix Array / LCP
Source: Classic
Difficulty: Hard
Topics: Suffix Array
Notes: Build SA for substring queries.

6. Suffix Automaton - distinct substrings
Source: Advanced
Difficulty: Hard
Topics: Suffix Automaton
Notes: Count distinct substrings and occurrences.

7. Longest Palindromic Substring (Manacher)
Source: Classic
Difficulty: Medium
Topics: Manacher
Notes: Linear algorithm for palindromes.

8. Edit distance variants (restricted moves)
Source: Classic
Difficulty: Hard
Topics: DP, Strings
Notes: DP with custom costs.

9. Minimal rotation / Booth's algorithm
Source: Classic
Difficulty: Medium
Topics: Strings
Notes: Find lexicographically minimal rotation.

10. Find smallest cyclic shift (applications)
Source: Classic
Difficulty: Medium
Topics: Strings
Notes: Use Booth or suffix techniques.

11. Substring queries with hashes
Source: CF/CSES
Difficulty: Medium
Topics: Hashing, Strings
Notes: Precompute rolling hashes for O(1) substring compare.

12. Edit distance under modulo constraints
Source: Advanced
Difficulty: Hard
Topics: Strings, DP
Notes: Combine DP and modular arithmetic.

13. String compression / run-length encoding tasks
Source: Classic
Difficulty: Easy
Topics: Strings

Notes: Greedy compression algorithms.

14. Wildcard pattern matching (DP)
Source: LeetCode/CF
Difficulty: Medium
Topics: DP, Strings
Notes: Support '?' and '*' wildcards.

15. Counting occurrences across multiple strings (Aho variant)
Source: Advanced
Difficulty: Hard
Topics: Aho-Corasick
Notes: Aggregate counts across texts.

16. Longest common prefix queries (LCP RMQ)
Source: Suffix Array theory
Difficulty: Hard
Topics: Suffix Array, LCP
Notes: Use LCP with RMQ for queries.

17. Periodic strings and borders (prefix-function)
Source: KMP theory
Difficulty: Medium
Topics: Prefix-function
Notes: Detect periodicity by prefix function.

18. Subsequence automaton / next-occurrence DP
Source: Advanced
Difficulty: Hard
Topics: Automaton, DP
Notes: Precompute next indices for subsequence checks.

19. Anagram grouping and frequency hashing
Source: Classic
Difficulty: Easy
Topics: Hashing, Strings
Notes: Use sorted strings or frequency vectors.

20. Minimum cyclic shift with suffix array
Source: Advanced
Difficulty: Hard
Topics: Suffix Array, Strings
Notes: Use SA over doubled string.

Geometry

1. Point in Polygon (ray casting)
Source: Classic
Difficulty: Medium
Topics: Geometry
Notes: Even-odd rule or winding number.

2. Convex Hull (Graham / Andrew)
Source: Classic
Difficulty: Medium
Topics: Geometry, Convex Hull
Notes: Compute hull and use for further problems.

3. Line intersection and orientation
Source: Classic
Difficulty: Easy-Medium
Topics: Geometry
Notes: Robust orientation and intersection checks.

4. Closest Pair of Points (2D)
Source: Classic
Difficulty: Hard
Topics: Geometry, Divide and Conquer
Notes: O(n log n) divide and conquer approach.

5. Rotating Calipers (diameter, width)
Source: Advanced
Difficulty: Hard
Topics: Geometry
Notes: Compute pairs on convex hull.

6. Polygon area and centroid
Source: Classic
Difficulty: Easy

Topics: Geometry
Notes: Shoelace formula for area.

7. Circle-Line and Circle-Circle intersections
Source: Classic
Difficulty: Medium
Topics: Geometry
Notes: Geometric formulas and edge cases.

8. Integer geometry - lattice points on segment
Source: Classic
Difficulty: Medium
Topics: Number Theory, Geometry
Notes: Use gcd to count lattice points.

9. Tangents between circles (externals/internals)
Source: Advanced
Difficulty: Hard
Topics: Geometry
Notes: Compute tangent points analytically.

10. Geometry with precision handling (eps issues)
Source: Classic
Difficulty: Medium
Topics: Geometry
Notes: Practice robust comparisons and eps usage.

Flow & Matching

1. Max Flow - Edmonds-Karp (intro)
Source: Classic
Difficulty: Medium
Topics: Max Flow
Notes: Simple BFS-based augmenting path implementation.

2. Dinic's Algorithm - Max Flow
Source: Classic
Difficulty: Hard
Topics: Max Flow, Dinic
Notes: Layered graph and blocking flow.

3. Min-Cost Max-Flow (successive shortest path)
Source: Advanced
Difficulty: Hard
Topics: MCMF
Notes: Find min cost augmenting paths.

4. Bipartite Matching - Hopcroft–Karp
Source: Classic
Difficulty: Hard
Topics: Matching
Notes: Efficient matching for bipartite graphs.

5. Assignment problem - Hungarian algorithm
Source: Advanced
Difficulty: Hard
Topics: Matching, MCMF
Notes: Optimize assignment with cost matrix.

6. Flow with lower and upper bounds
Source: Advanced
Difficulty: Hard
Topics: Flow, Circulation
Notes: Transform to standard flow via super-source/sink.

7. Maximum Flow applications - project selection
Source: CF/ICPC
Difficulty: Medium
Topics: Flow Applications
Notes: Min cut formulation for selection problems.

8. Edge-disjoint paths via flow
Source: Classic
Difficulty: Medium
Topics: Flow
Notes: Model disjoint path problems as flows.

9. Vertex capacities via node-splitting
Source: Classic
Difficulty: Medium
Topics: Flow modeling

Notes: Split node into in/out with capacity edge.

10. Flow with scaling / performance tricks
Source: Advanced
Difficulty: Hard
Topics: Flow, Optimization
Notes: Implementations tips for large graphs.

Mixed / ICPC-Style

1. ICPC Regional Past Problem - example A
Source: ICPC Regional Archive
Difficulty: Hard
Topics: Mixed
Notes: Representative problem from regional archive.

2. ICPC Regional Past Problem - example B
Source: ICPC Regional Archive
Difficulty: Hard
Topics: Mixed
Notes: Representative problem from regional archive.

3. ICPC World Finals Past Problem - example C
Source: ICPC WF Archive
Difficulty: Hard
Topics: Mixed
Notes: Representative world finals problem.

4. Multi-Topic Puzzle (combined graph + DP)
Source: CF/ICPC styled
Difficulty: Hard
Topics: Mixed
Notes: Combine multiple techniques under time pressure.

5. Implementation-heavy ICPC problem (corner cases)
Source: ICPC/Regional
Difficulty: Hard
Topics: Implementation, Ad-hoc
Notes: Practice robust implementation and edge cases.