Classics of Competitive Programming

Mamnoon Siam

Misc.

Exchange Argument

1. USACO 2019 February Contest, Platinum Problem 1. Cow Dating [odds algo?]

Binary Search

1. Given grid $g,\ g_{i,j} \in \{0, 1\}$. $Q\ (\leq\ 10^6)$ queries $query(x_1, y_1, x_2, y_2)$ = maximum subsquare totally consisting of 1s and fully inside $[x_1, x_2]\ \times\ [y_1, y_2]$subgrid. Animals and Puzzles. 2d sparse table. Binary search. (a trivial dp-ish thing).
2. Problem - 1109C
3. The Kingdom of JOIOI (JOI17_joioi)
4. Fire in the city - 845E ✱✱
5. USACO 2020 February Contest, Platinum Problem 1. Delegation
6. Problem - 101234A
7. Kth largest subarray

Parallel

1. Task Meteors (met)
2. Task Solar lamps (lam)
3. COCI 2020./2021. - Round #6 Index [given array, query(l,r) = max x such that there are at least x numbers >= x in range [l, r]]
4. ROI 2021 final p8

Not Binary Search

1. Given a weighted graph, find an odd length cycle s.t. AND-sum of weights of edges of the cycle is maximum.
2. APIO Bali Sculptures.

Smart Greedy

1. Prefix Enlightenment.
2. APIO 2007 Backup / JOISC18 Candies

3. Buy Low Sell High
4. 1415E - New Game Plus!
    - Given numbers. Distribute them in k+1 sequences. For each sequence, reorder the numbers. Take the sum of prefix sums. Add those k+1 summations. Maximize this.

Randomized Algorithms

1. Blogewoosh #6 (shuffling a sequence randomly yields $\approx \log N$ prefix max changes).
    a. 1101F - Trucks and Cities
    b. 101149M - Ex Machina (find the second max of a hidden array)
2. Birthday paradox.
3. Well… randomized binary-search expected complexity $\approx 2\log(N)$. See 981F - Round Marriage's editorial discussion (*todo).
4. Randomized algorithms lecture, part 1 & 2
5. G. Nuts and Bolts [quick sort]
6. Shortest path query when shortest path len >= N/c

Smart Observations & Cool Tricks

1. A. Airplane Cliques: Given a tree of N vertices, X. A pair of vertices are good if their distance is at most X. $f(k)$ =number of subsets of size *k* s.t. each pair of vertices from the subset is cool. Find $f(1)$, $f(2)$, $f(3)$, … $f(n)$.
2. 1250I - Show Must Go On (gym) [smallest subsets]
3. 1142C - U2 [geo,transformation]
4. Rope (JOI17_rope)
5. How to prove the composition of two functions of type min(c, max(b,x+a)) is of the same type and many more!
6. F - Flip and Rectangles

Manhattan Distances, Chebyshev Distances

1. Rotating by 45 degrees and zooming √2x. IOI 2007 Pairs
2. Generalization of Chebyshev distance. Multidimensional Queries
3. USACO 2020 February Contest, Platinum Problem 2. Equilateral Triangles

Convex Objects

1. Minkowski sum! 300iq contest 2 (gp of moscow). Combine two convex functions in O(n).
2. timeismoney (balkan11_timeismoney) [minimize X*Y, hyperbolas!]
3. 1366F - Jog Around The Graph
4. 1019E - Raining season [similar to dynamic diameter?? h(x) = f(x) + g(x) where f, g, h are hulls]

5. Logistical Questions [*****] (then XX OC GP of Zhejiang Problem C)
6. USACO 2020 January Contest, Platinum Problem 3. Falling Portals
7. HDU 2993: max subarray avg of len >= k
8. Minimum Variance Spanning Tree: subtask T=2 [see 14th article of this, similar to balkan::timeismoney]

DP

1. Jewel Thief [(max, +) convolution, given that one of the arrays is concave]
2. KAIST 2018 Spring. Touch The Sky [see journal may 19 2020]
3. XX OC GP of Zhejiang Problem C

Backtracking

1. Fancy Antiques

~~Ad fucking hoc~~ Bruh~

1. Gym 1267I - Intriguing Selection [constructive, sorting, partial order] (**TODO:** how to solve these problems myself?)
2. COCI 2020-21 Round 4, Task Hop

Hashing

1. 1320D - Reachable Strings
2. The Untended Antiquity [+2d range update, point query]

Two-Step tasks

1. View problem - Airline Route Map (JOI18_airline) [bitmasks, recover indices, connect u and i-th additional node, if u's i-th bit is on]
2. JOISC 2021 Day 3 Ancient Machine: can you transform the data such that unuseful information is not sent? In general, **try to strip off unuseful information as much as possible**. For example, in this problem, if there are two adjacent ones in the binary string, one of them will not be useful, and we can make that 0, which results in no two ones being adjacent to each other. Now we can do shenanigans using Fibonacci numbers i.e. **Zeckendorf representation of a number** :)!

Is one thing reachable from another?

- Hashing
- Find lexicographically smallest from both directions [if transitions are reversible]
- 1320D - Reachable Strings

- 1500C - Matrix Sorting [backward: fix the last operation? second-last operation? …]

Deal with Extremal/lexicographically extremal Stuff

1. APIO Ninjas
2. View problem - Furniture (JOI20_furniture): maintain up-rightmost and down-leftmost path in grid.
3. Cells Blocking - 300iq Contest 3: work with two extreme paths in grid
4. Nerc - Japanese Game

Sorting

1. 1504E - Travelling Salesman Problem: given two arrays a, c. reorder (a[i], c[i]) such that sum[max(c[i], a[i+1] - a[i])] is minimized. [think of how a **final** arrangement may look like.]
2. 1513F - Swapping Problem

Processes

1. USACO 2017 December Contest, Platinum Problem 3. Greedy Gift Takers [approach to attack infinitely long processes: **pick some specific object, track that and only that**]

Combinatorics

Mind = Blown ❤️❤️❤️

1. 1458 - Latin Square [permutations, so, you can represent permutations by n pairs (i, p[i]), and if you need to take the inverse, just do swap(first, second) for all i, then the inverse is {q[first] = second}. Moreover, if you see the pairs as points on 2d plane, you can see that taking inverse is just mirroring the points wrt y=x line]
2. 1458D - Flip and Reverse [bruhx2] [euler tour, greedy]

Misc.

1. 1270E - Divide Points [induction, coloring board/grid] (similar, D - Choosing Points)
2. Almost Friends [monovariant? Smooth jumps]
3. 1019C - Sergey's problem [induction, graphs]

Process, Invariance, Monovariance

1. 1320D - Reachable Strings [bonus: hashing] [note that the operation is reversible. In such cases, a common idea is to find the lex. minimal/maximal string from A and from B

and check if they are the same. If you do so, you will see that the string looks like a bunch of 0s in the beginning, and then some 1's with at most one zero between two consecutive ones.]

PHP

1. 1500A - Going Home
2. Encryption (hard)
3. https://codeforces.com/problemset/problem/577/B

Data Structure

General Outline

Range queries

1. Divide and conquer. For each interval of the segment tree (which is the direct result of divide and conquer), let m be the mid-value. Explicitly find an answer (or some information) for [i, m] for every i (i.e. calc ans(/info) for every suffix). Do the same for prefixes of [m+1, i]. Try to merge suffix and prefix somehow. [Remember IOI 18 Meetings?]
2. Sweep through R, maintain the answer for every L <= R in some DS (read- segment tree :)).
3. Let's say you are given some function f(l, r) that takes a subarray A[l...r] of the given array A, and outputs some value that depends on the content of A[l...r]. If the query(l, r) = sum of f(i, j) such that l <= i <= j <= r, then this is a probable approach: sweep through r, maintain an array B, where B[i] = f(i,i) + f(i,i+1) + f(i,i+2) + … + f(i, r). Most of the times you will need to maintain B in a segment tree, and when you go from r to r+1, you will get some segment-tree-updateable change in B.
    - Sometimes you may need to work with segment tree's historic sums.
4. Precalculate some optimal stuff (e.g. for each index calculate rightmost[i] <= i and leftmost[i] >= i), and to find answer to queries, check condition (e.g. comparing l <= rightmost[i] or leftmost[i] <= r). For example, see ROI 2021 final stage problem 7

Updates and Queries

1. Let's say the updates are like applying some non-commutative, non-associative function over a range. And you have to find the value at indices in between updates. And… let's be super optimistic and further assume that finding the composition (well, our updates are something like this: do a[i] = f(a[i]) for i in [l, r], where the function f and the range [l, r] is given in every update. Try to see that after two updates on the i-th element, a[i] = g(f(a[i])), now if we are trying to say that we want to find the composition of f and g, that

is, h(x) = g(f(x))) of two of those functions are easy to calculate and can be done fast. Say no more, king! Sweep through 1 to |a|, maintain the q updates in the leaves of a segment tree of q leaves, where i-th leaf has the function given in the i-th update. Now, while sweeping, activate/deactivate a leaf, recalc segment tree. For query, you have to find the composition of a prefix of the segment tree. That's super neat! See this for example.

Todo

1. https://codeforces.com/gym/102586/problem/A
2. https://codeforces.com/gym/102586/problem/L
3. https://www.acmicpc.net/problem/18910
4. https://codeforces.com/contest/1290/problem/E
5. https://atcoder.jp/contests/agc036/tasks/agc036_b

Some Basic Sub-strategies

1. Foehn Phenomena (JOI17_foehn_phenomena) [basic theme of undo, modify, redo]
2. 1223F - Stack Exterminable Arrays
3. 1500D - Tiles for Bathroom [2 pointer]
4. HDU - 5213: Given $A[]$, k, answer Q queries (L_1, R_1, L_2, R_2): $\sum_{L_1 \le i \le R_1} \sum_{L_2 \le j \le R_2} [A_i + A_j = k]$.

   [Divide each query into 4 ranges, calc answer for each of them using mo's algo, use those values to calc actual answer (alternatively, divide A into blocks)]
5. Given array (nonnegative integers), count subarrays with sum >= 2*max. [cartesian dnc, iterate on the smaller side]

Smart Bijective Transformations

1. XXI OC, GP of Xiaomi, Problem D: f(S) for a string is a sequence T of length |S|, where T[i] = smallest unused positive int, if S[i] is the first occurrence of the i-th char, otherwise T[i] = the int that is already assigned for S[i]. Two strings P, Q are equivalent, iff f(P) = f(Q). Count how many equivalence classes are there among all substrings of a given string. [So, we need to smartly make some bijective transformations so that we can easily compare two strings. Let g(S) be a sequence T of length |S| where T[i] = distance from i to the previous occurrence of S[i], or 0 if it's the first occurrence of this char. Now what do we have? Two strings' g()'s 0s must coincide (there are at most σ(=26) of them) and the interleaving subarrays must match the corresponding subarray in the other string. Now, comparing two substring is easy, then just run the usual unique substring count algo]

Misc.

1. Video: Segment Tree Beats
2. IOI 2018 Seats [see this], (similar problem: 1270H - Number of Components, 1175F - The Number of Subpermutations [hashing!])
3. Army of Me (split tree?), I - Intrinsic Intervals, Tutorial on Permutation Tree (析合树)
4. 1361F - Johnny and New Toy [dumb]
5. 1401E - Divide Square [count number of horizontal and vertical segment intersections]
6. APIO 2020 — Problem B
7. Bridges (APIO19_bridges)
8. Street Lamps (APIO19_street_lamps)
9. New Home (APIO18_new_home)
10. Land of the Rainbow Gold (APIO17_rainbow)
11. Railway (BOI17_railway) [virtual tree]
12. Interval tree - Wikipedia
13. APIO 2009 B Convention / JOISC 2021 Events 2: N tasks are given by their start time and end time. Choose the max number of tasks such that no two intersect. Print the lexicographically smallest subset. [well :) do the magic with the intervals, after that its easy]
14. Is Wavelet Tree Actually Useless?
15. IZho 2021 Day 2 Rooms: [transform some process to an exact formula, which will lead us to standard ds optimization] Subproblem: given two arrays a, b of length n and n+1, for each i, find: $\max_{1 \le l \le i \le r \le n} \min(b_l, b_{r+1}) - \sum_{i=l}^{r} a_i$. [use stack + segment tree / divide and conquer.]
16. F1 (F2 actually, find max subarray? → try to find observations. What happens when you move the right pointer / make the sub segment larger by one? Calculate for each r)
17. [Tutorial] Li Chao Tree Extended
18. A Simple Introduction to Li-Chao Segment Tree
19. A Simple Introduction to Parallel Binary Search
20. A Simple Introduction to CDQ Divide and Conquer
21. USACO 2018 US Open Contest, Platinum Problem 1. Out of Sorts [**]
22. Find a number in range that occurs more than (r-l+1)/2 times.
    - One possible candidate is the median
    - https://codeforces.com/blog/entry/89810?#comment-781992, Boyer–Moore majority vote algorithm
    - https://codeforces.com/blog/entry/89810?#comment-782614
    - lg^2 probabilistic solution: randomly select indices
23. Problem - 840D: find numbers in a range that occur at least (r-l+1)/k times, (k small)
24. Problem - 643G: generalization of boyer-moore majority vote algo
25. https://codeforces.com/contest/1516/problem/D [sparse table jumping, query[l, r]]

26. https://codeforces.com/contest/1479/problem/D: find a number in [l, r] that occurs odd times in path(u, v) [probabilities]
27. 1285E - Delete a Segment
28. https://loj.ac/p/2461
29. https://official.contest.yandex.ru/opencupXIX/contest/11930/problems/G/
30. https://codeforces.com/gym/102059/problem/K
31. https://www.codechef.com/JULY18A/problems/SUBWAY similar to joisc wild boar + but sparse table instead of segtree

Divide and Conquer

1. IOI 2014 Holiday
2. Offline dynamic connectivity
3. solve(l, r) -> solve all queries fully inside [l, r] (add some)
4. Sweeping (JOI20_sweeping)
5. Rikka With Maximum Subsegment Sum: sum{ max subsegment sum }
6. Range Diameter sum
7. 1217F - Forced Online Queries Problem: DCP
8. Vasya and Good Sequences
9. Subsequence Sum Queries: subsequences in [l, r] which have sum = 0 mod m (~20)
    a. USACO 2020 January Contest, Platinum Problem 2. Non-Decreasing Subsequences
10. Dynamic Connectivity:
    a. 813F - Bipartite Checking10.

Amortization

[Tutorial] A powerful representation of integer sets

A simple introduction to "Segment tree beats"

1. Stick Game. [intervals]
2. Interval of intervals [intervals]
3. 1137F - Matches Are Not a Child's Play [hld]
4. 1172E - Nauuo and ODT [hld, wtf is this?]
5. Sweeping (JOI20_sweeping) [maintaining groups]

Offline Tricks [read: offline sweep]

1. Given an array of distinct ints and an integer K(fixed). Answer $query(l, r)$: how many pairs (i, j) are there such that $a_i + a_j = K$ and$l \leq i \leq j \leq r$.
2. Sweep Line and Segment Trees
3. Dfs on tree and maintain stuff on root-to-node path: 1076E - Vasya and a Tree, F - Colorful Tree.

    a. 986E - Prince's Problem: find sum of min(x, a[w]) where w \in path(u,v), for each query (u,v,x)

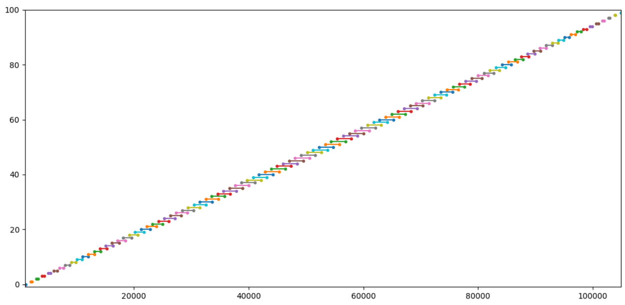
4. Distinct numbers in a range.
    a. 813E - Army Creation
    b. 1000F - One Occurrence
5. JOISC 2019 Antennas.
6. Employment. Thread.
7. Cartesian tree of an array. Few problems:
    a. Gardener Alex.
    b. Cartesian Tree.
    c. Tree Depth. (not a ds problem though. #counting #dp #usaco)
8. A bit modified version of cartesian tree: build(l, r) make the tree on a[l...r]. Min values in this range are grouped in the root, child subtrees are recursively built by calling build on the interleaved > min ranges.
    a. This is equal to the minimum operations needed to make [0,0,...,0] equal to some array with positive numbers using: choose l, r, x, do a[l] max= x, a[l+1] max= x, …, a[r] max= x. Call this f(a).
    b. Obviously f(a) is the number of nodes in the cartesian tree.
    c. There is another way to count the number of nodes of the cartesian tree. Add edges between l--r such that a[l] = a[r] and a[l] < a[i] > a[r] for all l < i < r. Number of nodes on the cartesian tree is equal to the number of connected components on this graph.
    d. We can take minimal edges of the graph such that the connectivity still holds in such a way no edges “cross”. That is, for a specific number, check only the two consecutive positions of that number’s occurrence, if you can add the edge. So now we have a forest of line graphs. In this case, #components = #nodes - #edges. That means, # of nodes of the cartesian tree = n - such minimal edges (i.e. ranges). Calculating the number of edges is waayyy easier sometimes.
    e. USACO Feb '21 - No Time To Dry
    f. C - Sequence Scores [counting problem]
    g. Postering
9. The Sum of GCD.
10. Well, well, well… that dope virtual contest arranged by Tasmeem Reza.
11. Similar problems:
    a. CF997E
    b. Army of Me (stack, segment tree)
    c. Sonya and Tree
    d. Given a tree, count the number of subgraphs where nods form a connected range.
12. 1428F - Fruit Sequences [given binary string, f(l,r) = max subarray of s[l..r] consisting of only ones, find sum[sum[f(l,r)]]]
13. XX OC GP of Zhejiang Problem D

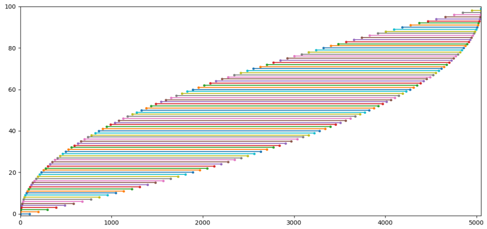
14. Query range mex of subarrays. (static array = offline sweep + segtree. Do a[pos] = val → …. There's a paltorator yt video on it)
15. IZhO 2021 Day 1 Potatoes
16. XXI oc gp of suwon - Colorful Squares
17. USACO 2016 February Contest, Platinum Problem 1. Load Balancing [bs + sweep]
18. USACO 2018 February Contest, Platinum Problem 1. Slingshot [2d absolute value]
19. 1470F - Strange Covering: find min area sum of two axis parallel rectangles that cover all the points.
20. Consider all the ranges of 1,2,3, … , n (choose(n+1,2) of them). Update: activate all the sub-ranges of [l, r]. Query: count how many sub-ranges of [l, r] are already activated.
    a. F[i] = max j such that the range [i, j] is activated (initially f[i] = i-1). Update(l, r): do f[i] max= r for all i in [l, r]. Query(l, r): sum f[i]-i+1 for in [l, r]. Segment tree beats.
    b. Alternatively, maintain the collection of update ranges, sorted by their left end-point, in such a way that if you find two such ranges that one is fully covering the other, delete the smaller one.
21. Valera and Queries [simple, just for implementation practice, count range inside range]
22. Developing Game: Given n tuples (li, vi, ri), find the max size of a set in which every two tuples are compatible with each other. A tuple (lx,vx,rx) likes (ly,vy,ry) iff lx <= vy <= rx and ly <= vx <= ry. [boils down to: given n axis-aligned rectangles in the 2d plane, find a point which is contained in maximum number of rectangles.]
23. 911G - Mass Change Queries [5x]
24. Souvenirs
25. prof pang and shit

Segment Tree

See later: Segment Tree Problems

1. B - Dynamic Diameter.
2. MULTQ3 - lazy + change children.
3. Multiply range, Increase/Decrease range, Query range sum.
4. LightOJ 1204 - Weird Advertisement [area covered by at least k rectangles, propagating down tags may not be a good idea always]
5. CCO '20 P5 - Interval Collection
6. Do we actually need lazy propagation on segment trees? 543E - Listening to Music
7. Given array. Queries: <l,r> find if any number appears more than (r-l+1)/2 times in a[l..r]. Update: change a[x]. [use the trick from IOI Towns]
8. 1500E - Subset Trick [observations (fairly easy to notice, binary search, convex thing, not with it ig?] [subproblem: maintain a set, query(l,r) = sum of f(i) | l <= i <= r where f(i) = sum of smallest i numbers, [offline+segtree, keep cnt, sums/treap+online] [**implementation practice**]





9. COCI 2020./2021. - Round #6 Index [given array, query(l,r) = max x such that there are at least x numbers >= x in range [l, r]] [persistent segment tree]
10. A funny "Trick" for coding RMQ with segment tree [a[i] += x in a range and range max query. For that, you don't need to write another query(l,r) function. For range max of [l,r] first do add(l,r,+BIG_NUMBER), then output t[0]-BIG_NUMBER, then do add(l,r-BIG_NUMBER) again]
11. 1083C - Max Mex [keep paths of a tree in each node of segment tree]
12. IOI '14 - Wall
13. 464E - The Classic Problem [psegtree :)]
14. Graph and Segment Trees (an awesome, detailed, well put together blog)
    a. 786B - Legacy
15. Flight Years: some cht boolshit (okay, maybe not)
16. 811E - Vladik and Entertaining Flags [keep tiny dsu in each node of sgt]
17. 1146E - Hot is Cold [lazy subproblem: updates: range set, range xor. In each node keep a pair(a, b) which means we need to apply set(a) first in this range, and them xor(b).]
18. 760E - Nikita and stack: retroactive stack problem [something to do with the prefix sum of stack operations]
19. 19D - Points: add point to the set, delete from the set, query(x, y) find leftmost point in p in set such that p.x >= x and p.y >= y. Do everything in lg(n)!
20. KGSS: find max sum pair in a range.
21. Problem KQUERYO: just merge sort tree / bit with vector
22. 794F - Leha and security system [keep function in each node, propagation = function composition, but the function is small]
23. 803G - Periodic RMQ Problem [just implicit segment tree]
24. 750E - New Year and Old Subsequence [maintain bullshit in each node]
25. View problem - Wombats (IOI13_wombats) [segtree+big-small tuning]
26. https://codeforces.com/gym/102503/problem/L [phillipine noi, lazy prop]

DP

1. 720D - Slalom
2. Two Dishes (JOI19_dishes)
3. Tsmm sama's note
4. USACO 2020 February Contest, Platinum Problem 3. Help Yourself
5. COCI21_SJECKANJE [save optimal value in range, merge]

6. 1368H2 - Breadboard Capacity
7. 1416E - Split
8. Volcanoes [2d sgt opt, but sgt not needed; set is enough]

DSU

1. 1303F - Number of Components
2. F - Cow and Vacation [given tree, go from a to b, you can use rest stops, you can travel <= k edges at once, determine if you can: big brain idea to use dsu, divide edges into two parts (insert node), from each node launch k radius bfs like thing, also merge two things if they collide]
3. Reachability tree:
   - Russian OI, 06-11 Aprill, 2021, Final stage problem 8: You are given an undirected weighted graph. Find shortest paths from 1 to all other vertices. A path can contain a vertex multiple times, but can have each edge at most once. Cost of the path is equal to minimum weight + maximum weight on that path.
   - 1416D - Graph and Queries

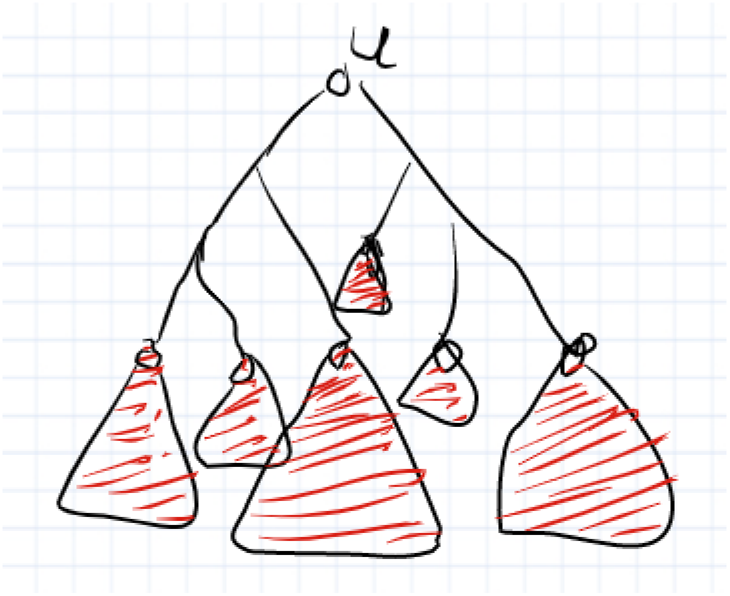
Tree Related DS

1. DFS Trickeries:
   a. F. DFS. For a given pair (u, v), set of nods w such that you can w → v w/o going through u and w → u w/o passing v
2. Basics - Tree updates:
   a. Point update, subtree query, (+ subtree update, point query), (++ subtree update, subtree query) [via tree flattening]
   b. Point update path query, (+ path update, point query) [doable with tree flattening]
   c. (b.), path update path query [via euler tour]
   d. Path update subtree query (+ vice-versa) [HLD, w/o HLD?]
3. DSU tree (IOI 2018 Werewolf, Maximum and Minimum, Clapway)
4. Binary Lifting, No Memory Wasted
5. Tree binarization?
6. Centroid Decomposition. (On Edge).
7. Minimum Spanning Tree Problem.
8. Lecture 17 — April 8, 2010 1 Overview 2 Euler Tour Trees
9. 1413F - Roads and Ramen [flip edge weight, find furthest distance of some path with even weight sum] [meta: any furthest fucking shit ----> first check diameter/its endpoints]
10. 1464F - My Beautiful Madness [observation, **hld**]
    a. Intersection of set of paths: consider the deepest vertex (not only on the paths- for example they took the d-th parent of lowest lca of the paths, **try anything extreme**) Another related example:
    b. Rikka with Intersections of Paths: number of ways to choose k paths so that the intersection is non empty. [fix the top node of the intersection, then at least one

path belonging to the chosen set has to have its lca as that node and all of the paths have their lca above that]

11. 1320E - Treeland and Viruses [just virtual tree]
12. 1495F - Squares [todo]
13. Path max queries on a tree in O(1)
14. A list of important concepts in Tree-based Problems
15. Almost-LCA in Tree
16. Usefulness of the DSU tree over LCT
17. Solution to CCF NOI 2018 Winter Camp Task 1, in English
18. Count total number of k length paths in a tree [the simple dsu is actually O(n): you are merging shallower subtree to deeper subtree and cost of merging is depth of shallower subtree, this is O(n)]
19. Giant Penguin: given graph, each node is part of at most k different vertex-simple cycles. updates : mark node, query: find min dist from given node to a marked node. [centroid decomp on graph? Fuck, 300iq is lit]
20. JOISC 2021 Day 3 Meeting2: Classic centroid decomposition problem
21. Count sum of sizes of minimal subtree containing a set of nodes (i.e. almost virtual tree, but take all the nodes in-between) → count edge contribution rather than node contribution; this is much easier and neater. In the end you need to add some problem specific constant to the answer.
    a. Find sum of choose(n,k) virtual tree (minimal subtree i.e. take all nodes) sizes. [count contribution of each edge. At the end, add nCk to the answer. Contribution of one edge is calculated as follows: you will iterate how many nodes you will select from each of the two subtrees (the two disconnected trees upon removing the edge under consideration). If you think for a moment, this becomes convolution and can be solved using fft]
    b. IZhO 2021 Day 2 Dendrology
22. XXI oc gp of suwon - Another Tree Queries Problem

23. USACO 2018 January, Platinum Problem 2. Cow at Large [centroid decomposition]



Suppose, all of the red nodes satisfy some specific property and it's easier to keep track of them altogether. Nodes are called 'red' in such a way that if a node is red, the whole subtree of it is red. Now say you want to count how many **disjoint** subtrees the set of red nodes consists of. Take $\sum_{u \in R} (2 - \deg(u))$. And, counting this summation is easier.

24. USACO 2018 February Contest, Platinum Problem 2. New Barns [maintain diameter]
25. USACO 2018 US Open Contest, Platinum Problem 3. Disruption [dsu / offline sweep]
26. USACO 2019 January Contest, Platinum Problem 2. Exercise Route: count number of pairs of given paths that are **not** edge disjoint. [Here's a slick little counting trick: split the paths into u--lca and lca--v, and solve the problem completely independently for this set of paths. But there can be overlaps, which can be easily be excluded (note that, a pair of paths only gets counted twice in the subproblem, when their lca is same, but not one of its endpoints]
27. 1470E - Strange Permutation: find the x-th node in the k-th lexicographically smallest path in a dag for each query (x[i], k[i]). The path length can be long, so you cannot really iterate on that. Instead of iterating, do something similar to hld- for each node, mark exactly one outgoing edge as heavy, and others light (if such stuff is possible, obviously). If you can mark edges as heavy/light in such a way that every path has at small amount of light edges, then you can do skipping like hld.
28. Circles of a Tree, Diameter of a set of vertices, Cover of that:
    - Infernape: in each query, you'll be given k circles, find number of nodes that are in at least k-1 of them. [**lemma**: intersection of two circles is another circles centered at a vertex or in the midpoint of some edge]
    - Range diameter sum: find sum{ max(dist(a,b) | l <= a, b <= r) | 1 <= l <= r <= n }
    - My Beautiful Madness

29. 1452G - Game On Tree
30. https://codeforces.com/contest/1527/problem/D: calc ans[k] = number of paths with mex k, for each 0<=k<=n. Boolshit path manipulation.

Binary Trie

1. Tutorial on Trie and example problems - Threads @ IIIT Hyderabad
2. Given an array a, Q queries $query(x) = \max_{1 \le i \le n} a_i \oplus x$.
3. Given array, find max xor sum subarray.
4. Given tree, Q queries $query(u, x) = \max_{v \in Sub_u} a_v \oplus x$. (a) online, (b) offline. Solutions: segment tree + trie, dsu on tree + trie, trie + vector in each node, persistent trie.
5. Given array a, you will be given two types of queries: (1) update the value of a_p by x, (2) find the k^{th} element in range $[l, r]$ in sorted order. Online. MST + trie in each node? Trie + pbds in each node. Mo with update.

Boolshit inequality bash

1. IOI 2007 Pairs
2. JOISC 2021 Day 3 Easiest Task
3. IOI 2016 Shortcut
4. Palembang Bridges (APIO15_bridge)

Old Codechef DS Problems

1. https://www.codechef.com/MARCH17/problems/TUPLES2: find triplets of paths such that either they are pairwise disjoint or pairwise intersecting. **[centroid decomp]**
2. SUMDIS - Editorial - editorial
3. OAK - Editorial - editorial
4. Cloning Problem Code: CLONEME Submit

Dynamic Programming

1. Knuth Optimization
2. Divide and Conquer Optimization
3. Monotone-Queue Optimization
4. WQS Binary Search Optimization
5. Convex Hull Optimization
6. Slope trick. A Codechef problem. Original Resource.
7. Non-trivial dp tricks and techniques.

8. JOISC 2019 Lamps.
9. E. Linear Kingdom Races (bunch of ranges over array and dp) (e.g. F. AND Segments)
10. 626F - Group Projects (open-close interval trick)
11. 1018 - Brush (IV)
12. 262144, Liar, Elevator Rides (Swapping DP State Problems) [well, don't limit yourself trying to use this trick only when one of the states becomes huge, try plugging in anywhere, may yield nice results e.g. lis problem], Collecting Stamps 3 (JOIOC20)
13. F
14. Dango Maker (JOI18_dango_maker) [observation, dp over diagonal]
15. 1442D - Sum [I mean, how to come up with the initial observation? Rest of it is dnc trix]
16. SOS DP:
    - Omkar and Pies [*****] [few cool observations and tricks e.g. apply operations from [l,r] to transform s to t … reverse operations => make independent, after that, the subproblem: given a[.] b[.] k find i, j | j-i>=k and common bits between a[i] and b[j] is maximized]
    - 986C - AND Graph
17. Algorithms Weekly by Petr Mitrichev: An undo week: calculate dp excluding i-th element for all i. Can also be done using dnc.
18. E - Increment Decrement [A Seattle week, A clean slate week] **TODO**[**BIG BRAIN IDEA**]
19. USACO 2016 - Landscaping **TODO**
20. USACO March'12 - Landscaping [cool transformation]
21. USACO 2016 February Contest, Platinum Problem 3. Circular Barn [dp opt, dnc, circular→ linear]
22. USACO 2016 December Contest, Platinum Problem 2. Team Building [PIE in dp]
23. USACO 2017 February Contest, Platinum Problem 2. Why Did the Cow Cross the Road II: modified lis
24. OII Sushi: Given $S\ (\le 10^5),\ a[1],\ a[2],\ \dots,\ a[n]\ (n \le 10^4)$. Find $\min\ (\max_{1\le i\le n} k_i)$ such that $\sum_{i=1}^{n} k_i a_i = S$. Everything is a non-negative integer.
    - $O(\frac{NS}{64} + S\log^2 S)$: first use bitsets to find the mask of what's possible using at most one of each type. Then, we can binary-search-on-answer and use FFT multiplication to take exponents of this bitmask.
    - $O(\frac{(N+S)S}{64})$: Note that in O(S/64), you can add one instance of one item to a knapsack DP, using bitset. Then, we'll cycle through the types and add 1 instance to the bitset one at a time; if it doesn't change the bitset, then we can omit it from future loops. Thus, each addition either increases the number of set bits, or we remove an item, so there are at most N+S runs total.
    - Basically write k[i]s in binary. O(NSlogS/64)
25. Max-flow min-cut theorem: model a max flow solution, max flow = min cut, finding min cut can be done using dp (also, rewrite min cut formula using specific properties of network, basically simplify things)

    - 724E - Goods transportation
    - 1368H1 - Breadboard Capacity

26. 813D - Two Melodies: selecting two disjoint subsequences of an array
27. Problem - 730I: [ez, but educational]
28. https://codeforces.com/contest/1527/problem/E: dumb af application of ds in dp table computation, just for showcase purpose.
29. https://loj.ac/p/2470: isotonic regression

Exchange Argument

1. TopCoder, SRM 502, TheProgrammingContestDivOne
2. Swap Space
3. D - Zabuton
4. Lecture #3 — Exchange arguments (sorting with dp) [all the problems and links are there]
5. 1354F - Summoning Minions
6. 1107F - Vasya and Endless Credits

Bitmask DP

1. USACO 2019 US Open Contest, Platinum Problem 2. Compound Escape
2. 1430G - Yet Another DAG Problem

Slick

1. 1253E - Antenna Coverage [What's so slick about it?]
2. https://codeforces.com/gym/102979/problem/G [**Andreasyan**: Let $dp_{l,r}$ be the answer in segment $[l, r]$. For calculation we need to choose some i from that segment to be maximum. For i it is optimal that $-C_{i,j} + V_{i,j} \cdot q_{l,r,i} + dp_{l,i-1} + dp_{i+1,r}$ is maximized, where $q_{l,r,i}$ is the number of queries from the segment $[l, r]$ containing i. We can pre-calculate q and optimal $-C_{i,j} + V_{i,j} \cdot q_{l,r,i}$ for all possible l, r, i by convex hull trick.] [What's so slick about it? **Swistakk**: The most important trick in this solution is that we do not need to take care of the fact that value at point i is actually bigger than all other values at intervals $[l, i-1]$ and $[i+1, r]$]
3. USACO 2017 January Contest, Platinum Problem 3. Subsequence Reversal: reverse any subsequence at first, then compute lis (non-decreasing actually) of the array. Your goal is to maximize this lis. Find its length. [subsequence reversal = sequence of back to back nesting swaps i.e. swap(i1, jk), swap(i2, jk-1) … swap(ik, j1) where i1 < i2 < .. ik < j1 < j2 < … jk]

Sorting transitions and relaxing DP [this really is a big brain idea]

1. 1497D - Genius: transitions have complete ordering and each subsequent transition must be increasing.
2. Satanic Panic: find # of convex pentagons [sort line segments by angle, segments are transition]
3. 1253E - Antenna Coverage [maybe this too? Maybe not]

Counting / DP

1. Sums and Expected Value — part 1
2. Sums and Expected Value — part 2
3. DIfferentiate and get recurrence.
4. Interpretation of product. The Interpretation.
5. Satanic Panic (notion of expanding sets systematically) (e.g. Neko Rules the Catniverse damn/ these are so cute). **Remark:** Recurrence of euler number is also expanding previously built sets (well, factorial too) but this notion is underrated most of the time.
6. USACO 2020 February Contest, Platinum Problem 3. Help Yourself [power trick]
7. Given n, k, find the number of non-negative integer solutions to $x_1 + x_2 + ... + x_k = n$ where $x_1 \le x_2 \le ... \le x_k$.
    a. This is equivalent to: how many sequences $a_0, a_1, ..., a_k$ are there such that $\sum_{i=0}^{k-1} (k - i) \cdot a_i = n$
    b. If x1 = 0, then we can solve dp[n][k - 1] for remaining numbers. otherwise x1 > 0, then all other numbers must be positive too. we can subtract 1 from each of them and solve the problem for dp[n - k][k]. dp[n][k] = dp[n][k - 1] + dp[n - k][k].

Tree DP

1. DP on Trees Tutorial.
2. Contribution trick: sort by $dp_{v,0} - dp_{v,1}$ and sum largest k $dp_{v,0} - dp_{v,1}$.
    a. 1223E - Paint the Tree
3. Maintaining DP on the go / two times dp (adjust dp value while going to a child).
4. Slope trick. APIO Fireworks.
    a. Given a rooted tree with N nodes. Nodes have values Select some nodes such that any path from root to leaves have at-most K nodes. Maximize the sum of values of selected nodes. Print answer for all K <= N [normal dsu is actually n log n, levels get merged; similar to find k len paths in a tree, see Tree Related DS 17]

   b. Find the maximum sum of lengths of k vertex disjoint paths in a tree. [haven't seen anywhere, but looks very very trivial setup]
   c. XX OC GP of Zhejiang Problem C
5. Barricade Trick.
   a. POI Barricades
   b. E - Attack to a Tree
   c. View problem - Džumbus (COCI19_dzumbus) [todo]
   d. housevisit - CNY Contest 2019 [todo]
   e. NERC - Hard Optimization
6. 997D - Cycles in product (+ centroid decomposition)
7. Dp On Trees *Given a tree T of N nodes and an integer K, find the number of different sub trees of size less than or equal to K. $O(n * K)$, instead of $O(n * K^2)$.*
8. JOISC 2021 Day 4 worst_reporter4
9. 1517F - Reunion [**TODO**]
10. 1146F - Leaf Partition [ez, todo]

Convex Shits

1. XX OC GP of Zhejiang Problem C
2. IOI Aliens
3. APIO Fireworks
4. 713C - Sonya and a problem w/o legend
5. Buy low sell high
6. Ummm… APIO'07 Backup?

Graph

Misc.

1. RectangleArea, APIO 2011 Table Coloring
2. CHN06: Given a weighted bipartite graph with nodes {1…2n} with partites {1…n} and {n+1…2n} and cost function c(u,v) (for 1≤u≤n and n+1≤v≤2n), extend the graph into a complete bipartite graph so that every perfect matching has the same cost. It is guaranteed that there is a unique way to do this. What is the sum of the squares of the edge costs, modulo 10^9 +7? [Nirjhor Sama (uwu): u basically need an NxN cost matrix, c[i][j] = cost of i — n+j, for any permutation you need the (sum cost[i][p[i]]) value to be same, by the exchange thing you have a+c = b+d if there are four values
   a b
   c d
   in other words a-b = c-d, which means difference of any two columns is the same value repeated N times, so the cost matrix must be sth like

a[1], a[1]+d[2], a[1]+d[3], ..., a[1]+d[N]
a[2], a[2]+d[2], a[2]+d[3], ..., a[2]+d[N]
...
a[N], a[N]+d[2], a[N]+d[3], ..., a[N]+d[N]
also introduce d[1]=0. this is the structure of the cost matrix for two sequences a[1...N], d[1...N]. you're given some cost[i][j] = w —> a[i]+d[j] = w so you create edge a[i]—-w—-d[j], and in this sort of graph you already know one value d[1]=0, so you can just run a dfs to compute the rest, the constraints ensure that the graph is consistent (there's no path to get two different values for the same node), and it's connected (unique assignment)]

The Theory

1. Hall's Theorem
    a. 1326E - Bombs
    b. 628F - Bear and Fair Set
    c. 1027F - Session in BSU
    d. 981F - Round Marriage
    e. IOI Teams
    f. 1036G - Sources and Sinks
    g. 611H - New Year and Forgotten Tree [todo]
    h. Task Ice Skates (lyz)
    i. Dates
2. BPM: F - Construction of a tree, 1348F - Phoenix and Memory, ACM ICPC 2014 WF problem I link [mis,geo]
3. Kuhn:
    - Minimize the sum of costs of matched nodes from the left: call dfs(left node) in increasing order of costs.
4. Euler theorem for planer graph:
    a. USACO 2019 US Open Contest, Platinum Problem 3. Valleys

Tree Isomorphism

1. Explanation for 'Tree isomorphism' talk
2. Regular Forestation

Coloring

1. 600F - Edge coloring of bipartite graph (also, Story about edge coloring of graph)

Bicoloring

1. JOISC 2017 Port Facility. Add edge $u \leftrightarrow v,\ v\ \in [l, r]$. Similar problem: COCI 2019/2020 Round 5, Matching.

Shortest Paths

1. Given a weighted graph, go from S to T, but one of the edges is unavailable, you don't know which, the only way to know is to be in either of the ends, minimize the total length of path from S to T in the worst possible scenario. Link.
2. 1163F - Indecisive Taxi Fee
3. Some shortest path problem (from Balkan OI 2012)
4. ICPC Live Archive - 4491
5. Problem - 542E
6. 3621 - Sightseeing Cows
7. Task Sums (sum) - Baza zadań
8. View problem - Travelling Merchant (APIO17_merchant): [at any moment, you can keep at most one item, so think of buying one item from city u, and selling that to v, and meanwhile, you will take the shortest path from u to v, now make another graph where each edge's length is shortest path from u to v and weight is max(0, sell[v] - buy[u]). Now find maximum (weight)/(length) ratio cycle. This is classic. There's a USACO problem I think, but I don't remember the link, maybe check the BDOI 2020 ioi training contests]
9. Problem - 196E (+ mst)
10. Flashing Fluorescents [bfs, observation]
11. SPOJ - MULTII
12. 1473E - Minimum Path
13. COCI 2020-21 Round 4, Task Patkice II [0/1 bfs]
14. Olympic Bus (JOI20_ho_t4)
15. Soccer (JOI17_soccer)
16. E [a[i]=p*q, add(p--q), find shortest cycle]
17. 1495D - BFS Trees [bfs, given two nodes x, y, find number of such spanning trees of the graph that it is also a bfs tree with source x and source y simultaneously]
18. 1450E - Capitalism [given some directed edge and some undirected edge, for each directed edge u--w-->v, add u--w-->v and v--(-w)-->u, and for each undirected edge add u--w-->v and v--w-->u] think in terms of paths… between two nodes makes sense? (TODO, didn't quite get the intuition)]
19. K (<= 50) cells are blocked in a nxm grid (nm <= 2e5). Q queries, each asking the distance of the shortest path between two cells. You can move in 4 directions. [if there's no blocked cell in the query subgrid, then print manhattan. Otherwise, there exists a shortest path that goes through at least one of the 4K cells (all adjacent cells of the blocked cells). Loop on that cell (4K iterations). Also precalc bfs from every one of those 4K cells]

20. IZhO 2021 Day 1 Cities [i-->next_smaller_element(i), i-->previous_smaller_element(i)]
21. 843D - Dynamic https://codeforces.com/contest/1473/problem/EShortest Path [O(V+E+MAX) vector trick]
22. 1307D - Cow and Fields
23. Useful Edges [fix some variables, put those quantities that only depend on fixed variables in one side, then minimize/maximize other side] [initially push all the query paths into a PQ, then take maximum from the PQ, try to append an edge from the original graph to that path, at the end you'll get the optimal path for each edge (a, b)]
24. Problem - 1486E: just an example of (node + another state)
25. XXI OC GP of China: D - City Brain
26. XX OC GP of ByteDance: G - Back and Forth
27. Shortest Path Query
28. 1004. Sightseeing Trip
29. K-th Path [tweaking algo]
30. BOI 14 portals

Euler Tour

1. Senior Postmen (BOI14)
2. 537E - Data Center Drama
3. UVa 10735 - Euler Circuit
4. 1459F - Flip and Reverse ⋆⋆⋆
5. 429E - Points and Segments [virtual edge trick]
6. 723E - One-Way Reform
7. 788B - Weird journey
8. BZOJ 3706: Given an undirected graph with n vertices and m edges, the edges have two colors: black and white. You can perform circuit inverting operations several times, starting from any vertex each time, inverting the color after passing an edge, and returning to the starting vertex. Determine whether you can make all the edges of this picture white through several operations. If possible, find the minimum number of operations. n, m ≤ 1e6.
9. China IOI TST 2018 Notes, (see the last note)

Flow / Cut

1. Maximum density sub-graph
2. Pretty Boxes (min-cost-max-flow, ds, segtree, cc-long) **(*todo)**
3. 1473F - Strange Set
4. 212A - Privatization
5. ExtremeSpanningTrees: + isotonic regression comment

Minimum Spanning Tree

1. 888G - Xor MST (Boruvka's Algorithm)
2. Given a weighted undirected graph, there are at most 5 edges with a particular weight, count how many minimum spanning trees are there.
3. Given a weighted undirected graph, for each edge, find:
    - if it must be in mst
    - if it is in at least one mst
    - if it is never in mst

    Process all edges of same weight(*w*) all at once, make graph:

    for each *(u,v,w)*: add edge *find(u)* -- *find(v)*

    Find bridges.
4. Given an array a, $cost(a) = \sum_{i=1}^{n} \max_{j<i} a_j \oplus a_i$. Rearrange array to maximize cost.
5. 827D - Best Edge Weight
6. 160D - Edges in MST
7. 1184E3 - Daleks' Invasion: for all edges, find max weight for which this edge exists in some mst.
8. USACO 2016 February Contest, Platinum Problem 2. Fenced In
9. Minimum Variance Spanning Tree: subtask T=1 [see 14th article of this]
10. Egalitarianism2
11. Magic Matrix

Games

1. Cop and Robber (BOI14_coprobber)
2. G of this
3. See this later.
4. Winning ways for your mathematical plays?
5. https://codeforces.com/gym/102893/problem/I: given a bunch of points, players have to select a segment st it does not properly intersect with the previous ones. Player who cannot move, loses. Decide whether you want to be the first or second player. Win for that player. [think about the terminal state, what happenes then? How does it look?]
6. https://www.codechef.com/problems/HAMILG [matching, vertex geography, todo]
7. Also, XXI OC GP of Xiaomi problem A

Math

NT

1. 1500B - Two chandeliers [crt]

Probabilities

1. Hypno - GP of Warsaw [well, actually, this is less probability and more about that neat trick of calculating dp values in increasing order of the actual dp[]. Um… I think, i'll have to check]
2. USACO 2018 December Contest, Platinum Problem 1. Balance Beam

Vectors, Basises

1. Codeforces 1163E (*)
2. 1336E2 - Chiori and Doll Picking (E1, E2) (*)
3. 1163E - Magical Permutation (+ gray code)

Counting

Snake oil method.

Recurrence

1. 1746. Hyperrook

Representative counting

1. Given a tree, each node has a certain probability of being destroyed, calculate the expected number of connected components of the forest. Episode 34 - Educational Streaming w/ Errichto

Easy

1. 1332E - Height All the Same $\sum_{i=0}^{N/2} p^{2i} q^{N-2i} \binom{N}{2i}$

Inclusion-Exclusion

1. Sky Full Of Stars
2. 1342E - Placing Rooks

Combinatorial

Geometry

1. Given finite number of points inside a right triangle, prove that, there is a way to draw a polyline which visits every point exactly once (so the number of segments is one less than the number of points) and whose sum of squares of segment lengths is at most the square of the hypotenuse length. (Topcoder SRM 771).
2. 996E - Leaving the Bar

Geometry

1. https://codeforces.com/blog/entry/88635?#comment-770603 [given set of points, form C(n,2) line segments joining pair of points, find which of them does not properly intersect with any other line segments]
2. APIO 2010 Signaling
3. 1284E - New Year and Castle Construction
4. Fiber Shape
5. USACO Nov'13 - Line of Sight [tangents? Analysis, also, see this]

Random Problems

1. Given a set of points. Each point will stay alive with probability p_i. Calculate the expected number of points of the convex hull. Calculated the expected area of the convex hull. (count contribution) Episode 34 - Educational Streaming w/ Errichto
2. Given a set of points. Count the number of pairs of completely disjoint triangles which can be formed by using the given points. (bijection, every pair of convex objects, on 2d plane, has exactly two interior tangents) Episode 34 - Educational Streaming w/ Errichto

Contests

1. 300iq Contest 1
2. XX Open Cup: GP of Kazan, contest (300iq contest 2)
3. 300iq Contest 3
4. XX Open Cup: GP of Zhejiang.pdf (discussion) (contest)
    a. this problem
5. Codeforces Round #588 (Div. 1) (by Radewoosh)
6. XX Open Cup GP of Tokyo
7. XX Open Cup GP of Bytedance pset.pdf

8. XX Open Cup: GP of Nanjing cfgym
9. 2020-2021 Winter Petrozavodsk Camp, Day 9 Contest (XXI Open Cup, Grand Prix of Suwon)

Papers

1. Isotonic Regression via Partitioning
2. (PDF) Weighted isotonic regression under the L 1 norm
3. Linear Time Algorithms for Knapsack Problems with Bounded Weights
4. Chinese IOI TST Papers
5. Optimal Edge Ranking of Trees in Linear Time

Blogs

1. Fresh and unconventional algorithms & ideas for competitive programming

TODO

1. G:Geolocation
2. The Detailed Editorial for AtCoder World Tour Finals 2019-E (dp + integration?)
3. [Tutorial] Matroid intersection in simple words
4. Nim arithmetic
5. Randomized weighted majority algorithm

FFT/Polynomials

1. $\prod_{1\le i<j\le N} |a_i - a_j|$ modulo M. Link.
2. A problem collection of ODE and differential technique
3. [Tutorial] Generating Functions in Competitive Programming (Part 1)
4. [Tutorial] Generating Functions in Competitive Programming (Part 2)
5. Pentagonal number theorem - Wikipedia
6. TreeDistance [interpolation, matrix tree theorem, eigenvalue]
7. USACO 2020 US Open Contest, Platinum Problem 2. Exercise [egf]

Fwht

1. I. Bitwise Magic

Matrices [lin alg?]

1. Sherman Morrison Formula
2. XX Open Cup GP of Bytedance K
3. https://codeforces.com/gym/102979/problem/F

Hard Problems

1. Codechef NOV19 - Pretty Boxes [idk about that]

Communication / Two Step problems

XOR, mapping/bijection, bruteforce, **BITMASK SHIT (BINARY REPRESENTATION)**
Interactive and Communication Problems · USACO Guide

Interactive

1. Ceoi carnival
2. https://docs.google.com/document/d/1B2qX2_AhJ9y1x8DfjsMtFVTtZHp1-9g33y5jAGtyJsU/edit
3. E - Find the vertex
4. https://vjudge.net/contest/316360
5. 3 Hard Interactive Problems
6. Weighted cd + bs on tree problem from 300iq contest

USACO

2015-2016

1. Landscaping
2. High card low card
3. Lights out - [hmm, ok. Statement was kinda ambiguous]

2016-2017

1. Robotic Cow Herd

2. Switch Grass
3. COWBASIC

2017-2018

1. Push a Box [impl is nasty]
2. Sprinklers [ugly summation problem]
3. Cow at Large [centroid decomposition, let l(x) be the distance to closest leaf from node x. For each u, ans(u) = sum{ (2-deg(v)) | |uv| >= l(v) }
4. Lifeguards [dp, somewhat standard, will prolly implement]
5. Cow Gymnasts: dunno, some nt problem? Fuck it

2018-2019

1. The cow gathering: that weird ordering + tree problem
2. Train Tracking 2: siding window counting problem, I don’t think it’s a cool problem
3. Mooing Mischief: big-brain dp optimization problem, todo
4. Valleys:euler formula [will implement]
5. Compound Escape: bitmask dp, will implement prolly
6. Exercise Routes: neat pair of paths counting trick
7. Dating App: was kinda wtf moment for me, but has a nice pss

2019-2020

1. Equilateral Triangles [practice for 45 deg rotation?]
2. Falling Portals [cool convex hull problem
3. Circus [find condition → DCP?]

HARD Segment trees

1. Little Pony and Lord Tirek
2. The Child and Sequence
3. Gorgeous Sequence (2015 Multi-University Training Contest 2) (soln)
4. Bear and Bad Powers of 42
5. Nagini
6. Julia the snail
7. 【UR #11】元旦老人与数列- 题目
8. Box Operations
9. And or Max
10. DZY Loves Colors