



Trademark Contest 2 Analysis

Oleksandr Kulkov

Osijek, Croatia

February 1, 2026

General info

Contest authors: Tyler Marks, Sachin Sivakumar (UCF)

Onsite winner: UWaterloo: Waterloo Black

Unsolved problems: 1

L. Trimming Hedges

Statement

You're given $h_1, \dots, h_n$.

You can pick k , h and $r < k$, and do $h_{kt+r} = \min(h_{kt+r}, h)$ for all t .

Make height the same for all h_i .

Topics: Ad Hoc

- If $h_1 = \dots = h_n$, the answer is 0;
- If $\gcd(i_2 - i_1, i_3 - i_1, \dots) > 1$ for $h_{i_k} > \min h_i$, the answer is 1;
- Otherwise, the answer is 2 (pick $k = 2$ on odds and evens).

A. Black Friday

Statement

You're given $a_1, \dots, a_n$ and $d_1, \dots, d_n$.

You apply $x \mapsto (x - a_i) \cdot d_i$ in any order.

Minimize final x , assuming it's initially very large.

Topics: Sorting

Let's compare if (a_1, d_1) or (a_2, d_2) is better to apply first:

$$d_1(d_2(x - a_2) - a_1) \stackrel{?}{<} d_2(d_1(x - a_1) - a_2)$$

$$-d_1d_2a_2 - d_1a_1 \stackrel{?}{<} -d_1d_2a_1 - d_2a_2$$

$$a_2d_2(1 - d_1) \stackrel{?}{<} a_1d_1(1 - d_2)$$

$$\frac{a_1d_1}{1 - d_1} \stackrel{?}{>} \frac{a_2d_2}{1 - d_2}$$

Use it as a comparator for sort.

C. File Your Taxes

Statement

A tax system is defined as follows:

- It has n tax brackets, defined by $b_1 < \dots < b_{n+1}$ and $t_1 < \dots < t_n$;
- Income in $[b_i, b_{i+1}]$ is taxed at $t_i\%$;
- $b_1 = 0$ and $b_{n+1} = m$.

There are two tax systems. You'll report a total income x_i .

You choose $y_i \leq x_i$ and report

- $[0, y_i]$ to the first system;
- $[y_i, x_i]$ to the second.

Find minimum amount of taxes for all x_i .

Topics: Piecewise linear functions

- Let $f_i(x)$ be tax on x and $g(x) = f_2(x) - f_1(x)$;
- They are all piecewise linear;
- Start with $f_2(x)$, and find $y \leq x$ that maximizes $g(x)$;
- The answer is $f_2(x) - g(y)$;
- Sweep line by x , events:
 - New tax bracket in system 1 or 2;
 - New query point;
- Keep track of $f_1(x)$, $f_2(x)$ and max. value of $g(y)$ for $y \leq x$.

J. Return of Puzzle Boy

Statement

Two players add words to dictionary:

- All words must be m letters long;
- $w_i = d_i$ for some position i and dictionary word d ;
- $w_i \neq d_i$ for some position i and all dictionary words d ;
- If you can't add a word, you lose.

Count winning first moves in the game.

Topics: Games, Combinatorics

Game on stones

- Let a_i be number of unused letter at i ;
- Convert to m piles with a_i stones;
- A turn is picking k piles for $1 \leq k < m$ and removing one stone.

Win criterion

- All piles are even $\implies$ losing by symmetry;
- All piles are odd $\implies$ losing, because:
 - Can't pick all stones to make all even;
 - The other player can pick complement to make all even.
- Otherwise, the game is winning.

First move needs to pick either of:

- All non-zero even piles;
- All odd piles.

Count moves

Let I be indices of even piles. The answer is then:

$$\prod_{i \in I} a_i \prod_{i \notin I} (c - a_i) + \prod_{i \notin I} a_i \prod_{i \in I} (c - a_i)$$

F. K-Star Sets

Statement

Given a tree and k , find number of subgraphs that are a forest of k -stars.

Topics: DP on trees, NTT

DP states

Let's use $dp[v][c]$, where

- $c = 0$ if v is not connected to p (0 or k children);
- $c = 1$ if p is the center of k -star (0 children);
- $c = 2$ if v is the center of k -star ($k - 1$ children).

DP transitions

Transitions of interest:

- No child connections for $c \in \{0, 1\}$: add $\prod_u dp[u][0]$;
- Connect to a center-child if $c = 0$: add $dp[w][2] \prod_{u \neq w} dp[u][0]$;
- $t \in \{k, k-1\}$ children for $c \in \{0, 1\}$: add

$$[x^t] \prod_u (dp[u][0] + x dp[u][1]).$$

Find the latter in $O(n \log^2 n)$ with D&C and NTT.

E. IPv6 Address Compression

Statement

You have a hexadecimal string split in blocks of size k .

You can pick c , and

- Remove all occurrences of c ;
- Reduce all consecutive colons to at most 2.

Find smallest resulting string while assigning c_i to $[l_i, r_i]$.

Topics: Segment tree

Segtree nodes

Use a segment tree for each character. Store L , R , c_L , c_R and t :

- The range $[L, R]$ of the node;
- Number of colons on the left/right side;
- Number of characters.

Segtree of c tells us answer if c was deleted.

Segtree update

If updating a range to c :

- $t = 0$,
- $c_L = \min(2, \lfloor \frac{R}{k} \rfloor - \lfloor \frac{L}{k} \rfloor)$ and $c_R = 0$.

Otherwise:

- $t = R - L + 1 + \lfloor \frac{R}{k} \rfloor - \lfloor \frac{L}{k} \rfloor$,
- $c_L = [(L > 0) \wedge (k|L)]$, and $c_R = 0$.

Segtree merge

- If left total is 0, merge its colons into right's c_L ;
- If right total is 0, merge its colons into left's c_R ;
- Otherwise,
 - Merged c_L is left's c_L ;
 - Merged c_R is right's c_R ;
 - Merged t is $t_L + t_R + \min(2, c_R^{(L)} + c_L^{(R)})$.

Pick minimum over c . Works in $16(n + q \log n)$.

D. Hurricane Mapping

Statement

First run:

- Given info about n sensors, add one extra sensor;
- It is guaranteed each edge already has a center.

Second run:

- Can query point whether it is in convex hull;
- Find the area of the convex hull from first run in $\leq 10^4$ queries.

Topics: Geometry, Run twice

Extra point

- Add extra point in the corner that maximizes resulting area;
- Analysis shows the resulting area is at least $\frac{1000^2}{2}$ (see text tutorial).

Querying area

- The new convex hull covers at least $\frac{1}{2} \frac{1000^2}{5000^2} = \frac{1}{50}$ of the area;
- With 800 randoms, chance of not finding any in polygon is 10^{-7} ;
- Walk to the left to find point on the boundary (1000 queries);
- Walk over the perimeter (8000 queries).

Use endpoints from the perimeter to get area.

G. LCM Subset Count

Statement

Given factorizable $n, k \leq 10^{18}$.

Count subsets of $[n]$ with LCM a multiple of k .

Topics: Zeta / Möbius transformations, PIE

Counting non-divisibles

Let $k = p_1^{a_1} \dots p_m^{a_m}$, note $m \leq 15$.

- Let a_x be the count of numbers up to n divisible by primes in x :

$$a_x = \left\lfloor \frac{n}{\prod_{i \in x} p_i^{a_i}} \right\rfloor$$

- Let b_x be count of multiples of primes in x , not divisible by others:

$$b_x = \sum_{y \supseteq x} (-1)^{|y|-|x|} a_y$$

- Let c_x be integers not divisible by any prime not in x :

$$c_x = \sum_{y \subseteq x} b_y$$

We can find b_x and c_x as subset/superset sum DP.

Getting the answer

- There are $2^{c_x} - 1$ non-empty sets not divisible by prime powers in x ;
- Let d_x be subsets divisible only by prime powers in x :

$$d_x = \sum_{y \subseteq x} (-1)^{|x|-|y|} (2^{c_y} - 1)$$

Repeatedly use Möbius / Zeta transforms to compute efficiently.

B. Collecting Strawberries

Statement

You can move in 4 directions in a grid.

You can put obstacles of cost $c \cdot x$ between two adjacent cells.

It will increase time needed for traversal by x .

Find max. shortest time after spending k on obstacles.

Topics: LP duality, min cut in planar graphs

Optimal solution

$$\begin{array}{llll} \max_x & c^\top x & (\text{max}) & \xLeftrightarrow{\text{dual}} \min_\lambda & b^\top \lambda & (\text{min}) \\ \text{s.t.} & Ax \leq b & (\text{need } \leq) & & \lambda \geq 0 & (\text{var } \geq) \\ & x \geq 0 & (\text{var } \geq) & & A^\top \lambda \geq c & (\text{need } \geq) \end{array}$$

- Find min cut on the cost graph;
- Increase all min cut edges uniformly;
- Can be proven via LP duality (see text tutorial for details).

Min cut on planar graph

- Typical flow algorithms should be too slow;
- Consider dual planar graph:
 - Faces are vertices;
 - Edges between vertices are now edges between faces.
- Min. cut is shortest path in the dual graph (see Timus 1664).

Solves the problem in $O(mn \log mn)$.

K. Trapping Rain Water

Statement

Given a 2D grid, find which positions “trap water” inside.

Topics: Dijkstra, BFS

- Find min. max escape height from each cell h_{ij} with Dijkstra;
- Cell can retain water if $h_{ij} > n - i$;
- Simulate water flow with BFS:
 - If cell beneath is retaining or a wall, flow to left and right;
 - Otherwise, flow downward;
 - If current cell is retaining, also flow upward.

Solution time: $O(mn \log mn)$.

H. New Password

Statement

You're given a string.

Sort all substrings (with repetitions) and concatenate.

Print substring $[L, R]$.

Topics: Suffix structures

- Build a suffix tree;
- Count occurrences for each node;
- Consider lexicographic traversal;
- When processing an edge, we have:
 - Strings whose length is $l, l + 1, \dots, r$ each repeat k times;
 - Total of $k \frac{(l+r)(r-l+1)}{2}$ new characters per edge;
- Skip an edge, or find first relevant character;
- Iterate characters.

I. Pirates on a Boat

Statement

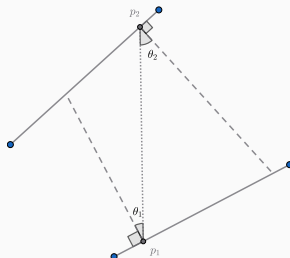
There are two polylines with increasing x , describing shores. Pick:

- A uniformly random point on bottom polyline;
- An angle θ with probability density $\frac{\cos \theta}{2}$;

Find probability you will (only) cross the other shore.

Topics: Probability, Integration, Physics

Pairs of points



Probability of landing in dp_2 is

$$V(p_1, p_2) \frac{\cos \theta_1}{2} \frac{\cos \theta_2 dp_2}{r},$$

where $V(p_1, p_2)$ is obstacle indicator, $\frac{\cos \theta_2 dp_2}{r}$ is the angular measure of dp_2 , and $\frac{\cos \theta_1}{2}$ is the probability density at that angle.

Reciprocity

We need to find

$$f(L_1, L_2) = \frac{1}{|L_1|} \iint V(p_1, p_2) \frac{\cos \theta_1 \cos \theta_2}{2r} dp_1 dp_2$$

We should note that $f(L_1, L_2)|L_1| = f(L_2, L_1)|L_2|$.

Shortest paths

Assume L_1 and L_2 are upper / lower convex hulls.

- A and B endpoints of L_1 ;
- C and D endpoints of L_2 ;
- $s(q, r)$ shortest distance between q and r without crossing polylines.

Consider space formed by L_1 , and also $s(A, C)$ and $s(B, C)$:

- $f(L_1, s(A, C)) + f(L_1, s(B, C)) = 1$;
- $f(s(A, C), L_1) + f(s(A, C), s(B, C)) = 1$;
- $f(s(B, C), L_1) + f(s(B, C), s(A, C)) = 1$.

If we start on any line, we have to cross one of the other.

Applying reciprocity

Apply reciprocity:

- $|L_1|f(L_1, s(A, C)) + |L_1|f(L_1, s(B, C)) = |L_1|;$
- $|L_1|f(L_1, s(A, C)) + |s(A, C)|f(s(A, C), s(B, C)) = |s(A, C)|;$
- $|L_1|f(L_1, s(B, C)) + |s(A, C)|f(s(A, C), s(B, C)) = |s(B, C)|.$

From this,

$$2|L_1|f(L_1, s(A, C)) = |L_1| + |s(A, C)| - |s(B, C)|,$$

$$2|L_1|f(L_1, s(B, D)) = |L_1| + |s(B, D)| - |s(A, D)|.$$

Finale

Consider region formed by L_1 , L_2 , $s(A, C)$ and $s(B, D)$:

$$f(L_1, L_2) + f(L_1, s(A, C)) + f(L_1, s(B, D)) = 1$$

Plugging in the previous formulas, we get

$$f(L_1, L_2) = \frac{|s(B, C)| + |s(A, D)| - |s(A, C)| - |s(B, D)|}{2|L_1|}$$

Substitute starting polyline with its hull to get the answer.
The result is known as **Hottel's crossed strings method**.