ICPC Asia Seoul Regional Contest 2025

Solutions Presentation

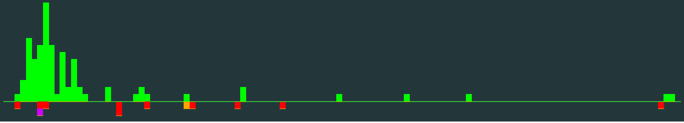
November 22, 2025

icpc.foundation

icpc Asia Seoul

M: Triple Fairness

Proposer: Nah Jeounghui, Setter: Youn Gyojun



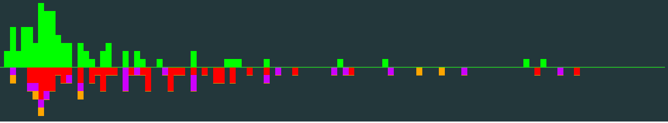
Problem: Given n **not divisible** by 3, find **any** Fair Problemset sequence of length $3n$.

Solution: Print 1 to n in order 3 times.

Complexity: $\mathcal{O}(n)$.

L: Segments

Proposer: Koo Jaehyun, Setter: Sim Jeong Seop



Problem: Given n horizontal segments, find the **maximum extension** required for all segments to reach a vertical query line $x = p$ for each of q queries.

Constraints: $n, q \leq 2 \times 10^6$

Solution:

- y-coordinates are redundant.
- Answer is determined by
  - the **maximum of left** endpoints $=: x_l$, and
  - the **minimum of right** endpoints $=: x_r$.
- $\text{Answer} = \max\left(x_l - p, p - x_r, 0\right)$.

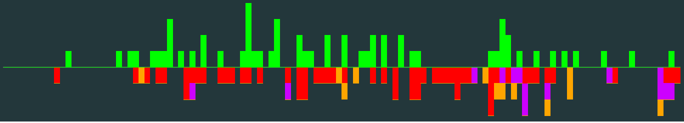
Pitfalls:

- Remember to apply $\max(0, \cdot)$.

Complexity: $\mathcal{O}\left(n + q\right)$.

C: Bay

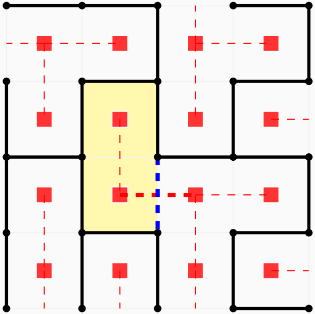
Proposer: Cho Hwan-Gue, Setter: Cho Hwan-Gue



Problem: Given a spanning tree of a grid graph, find all the non-tree edges that create cycles (bays) of area S.

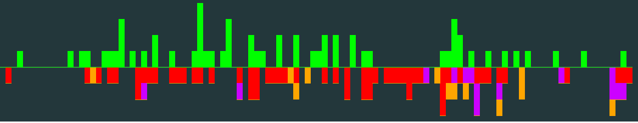
Solution:

- Treat grid cells as vertices to form a rooted forest.



C: Bay

Proposer: Cho Hwan-Gue, Setter: Cho Hwan-Gue



Solution:

- Set the vertices connected to the exterior of the grid as roots.
- Use DFS to count subtrees of size S.

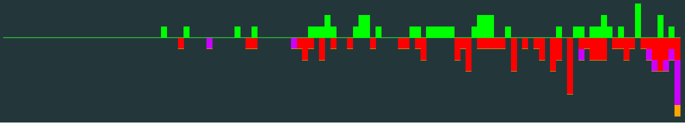
Pitfalls:

- You should count **subtrees**, not **cut edges**.
- Cells form a grid of $(n-1) \times (n-1)$.
- You **cannot** apply Pick's theorem; the number of interior lattice points may not be zero.

Complexity: $\mathcal{O}\left(n^2\right)$.

I: Magic Door

Proposer: Kim Soohwan, Setter: Kim Soohwan



Problem: Given $n \times m$ board, simulate a complex Match-3 game with bombs, gravity, and chain reactions. Count removed gems.

Solution:

- Implement rules exactly (Match-$3 \to$ Gravity $\to$ Bombs).
- Handle simultaneous removals carefully.
- Continue processing even if bombs are active but no matches exist.

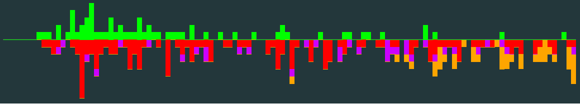
Pro-tips:

- Strict linear-time optimization for every step is **not** required; the total time complexity is naturally bounded by $\mathcal{O}\left((nm)^2\right)$.

Complexity: $\mathcal{O}\left((nm)^2\right)$.

G: Extraterrestrial Creatures

Proposer: Kim Yeonghyeon, Setter: Kim Yeonghyeon



Problem: Simulate X button presses where the smallest value increases. Find final values.

Constraints: $n \leq 5 \times 10^5$; $X \leq 10^{12}$

Solution:

- Let $f(t)$ be presses needed for all numbers to reach $\geq t$.

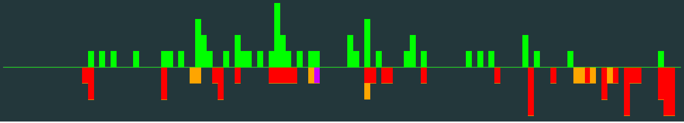
$$f(t) = \sum_i \left\lceil \frac{\max(A_i - t, 0)}{J_i} \right\rceil$$

- f is monotonely increasing; **binary search** for t such that $f(t) \leq X < f(t+1)$.
- Simulate the remaining $(X - f(t))$ presses on the smallest IDs.

Complexity: $\mathcal{O}(n \log(nX))$.

E: Clean Arrangements

Proposer: Shin Chan-Su, Setter: Shin Chan-Su



Problem: Find a linear arrangement of a rooted tree to **minimize total edge length**, such that no edge covers the root (clean).

Solution:

- Define D_v as the minimum of “subtree T_v cost plus overlap cost”.
- Sort child subtrees by size (descending).
- Place subtrees alternately to the left and right of v.

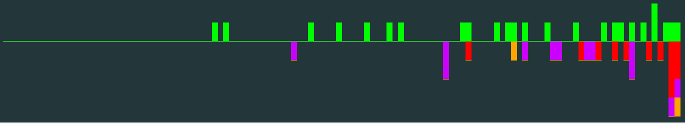
Pitfalls:

- The case where v is the root should be handled slightly differently.

Complexity: $\mathcal{O}(n)$.

K: Quadrants

Proposer: Bae Sang Won, Setter: Bae Sang Won

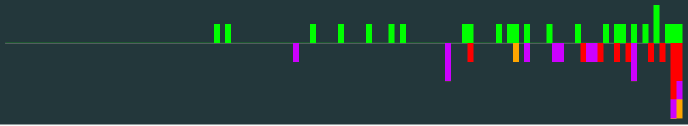


Problem: Given n points, count the number of k-quadrants (quadrants containing exactly k points) for all k.

Constraints: $n \leq 2\,000$

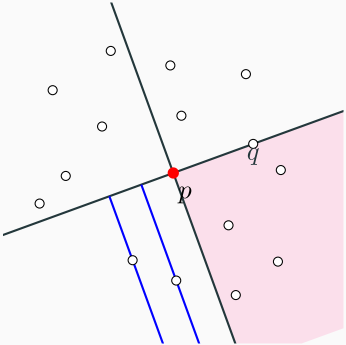
K: Quadrants

Proposer: Bae Sang Won, Setter: Bae Sang Won



Solution:

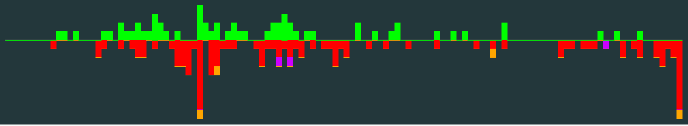
- Fix a point p, and rotate perpendicular axes.
- The figure below shows k-quadrants with $k \in [4, 6)$.



Complexity: $\mathcal{O}\left(n^2 \log n\right)$.

D: Bookshelf

Proposer: Kim Jae-Hoon, Setter: Kim Jae-Hoon



Problem: Can a favorite book be moved to a target position given specific rules about moving books into empty intervals?

Constraints: $n \leq 10^5$

Solution:

- Calculate free width $w := L - \sum A_i$.
- Books with width $\leq w$ are free to swap; those $> w$ maintain relative order.
- Anchor the favorite book and test feasibility of placing large books to its left/right.

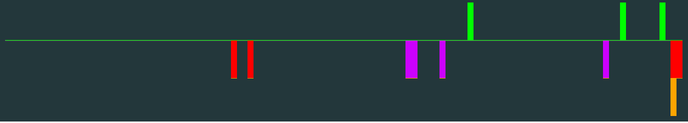
Pitfalls:

- You should separately handle whether the favorite book is large or not.

Complexity: $\mathcal{O}(n)$.

F: CPEquivalence

Proposer: Han Yo-Sub, Setter: Han Yo-Sub



Problem: Min edits to make array y CP-equivalent to x.

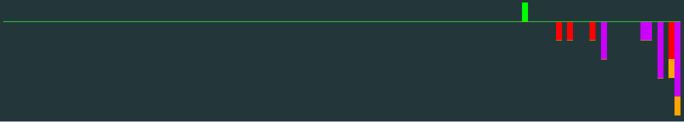
Constraints: $n, m \leq 40$

Solution:

- CP-equivalent $\equiv$ Maximum Cartesian trees are isomorphic.
- Build the Cartesian tree T of x.
- DP state $D_{[s,e],c,T_v}$:= Minimum possible value at node v for subarray $y[s \cdots e]$ of y with $\leq c$ edits.
- Optimize the DP using monotonicity and two-pointer techniques.

Complexity: $\mathcal{O}\left(40^5/6\right)$

J: Mex Culpa



Proposer: Koo Jaehyun, Setter: Koo Jaehyun

Problem: Given two sequences a and b, compute $f_i = \operatorname{mex}(\cdots)$.

Constraints: $n \le 2.5 \times 10^5$

Solution:

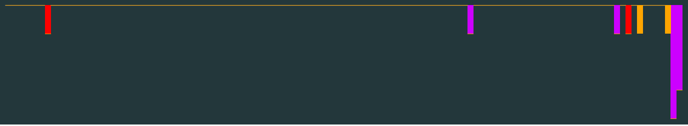
- Define $I_i := [a_i, a_i + b_i]$.
- Computing $\operatorname{mex}(\cdot) \equiv$ Greedily coloring interval graph I_*.
- Find smallest i such that $f_i = 0 \equiv$ Find smallest i such that $I_i \subseteq J$.
- Recursively find all $f_* = 0$, remove these, and repeat for $f_* = 1, 2, \cdots$.

Pro-tips:

- k-d tree $(\mathcal{O}(n\sqrt{n}))$ instead of 2D segment tree $(\mathcal{O}(n \log^2 n))$ also works.

Complexity: $\mathcal{O}(n \log^2 n)$.

A: Adventurer Dabi



Proposer: Youn Gyojun, Setter: Youn Gyojun

Problem: **Interactive**. Navigate an unknown grid to find a key, then reach the treasure in the minimum number of steps. Handle unknown teleports.

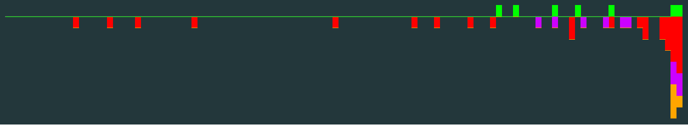
Solution:

- Move to an arbitrary cell adjacent to a wall.
- Discover all walls using the right-hand rule.
- Identify teleport cells and recover connectivity; Invertible moves allow backtracking.
- From the key cell, use BFS to find the shortest path to the treasure.

Pro-tips:

- The command limit is very loose; Wise and smart randomized strategies work.
- Use mock interactor and web visualizer wisely.

H: Fair Problemset



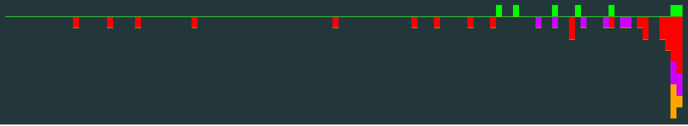
Proposer: Nah Jeounghui, Setter: Nah Jeounghui

Problem: Count Fair sequences of length $3n$ for every $n = 1, 2, \cdots, 10^6$.

Solution:

- Let P_i be the i-th problem.
  - Contiguous partitions:
    - $G_{1,*} := \{P_1, P_2, \cdots, P_n\}$
    - $G_{2,*} := \{P_{n+1}, P_{n+2}, \cdots, P_{2n}\}$
    - $G_{3,*} := \{P_{2n+1}, P_{2n+2}, \cdots, P_{3n}\}$
  - Jump partitions:
    - $G_{*,1} := \{P_1, P_4, P_7, \cdots, P_{3n-2}\}$
    - $G_{*,2} := \{P_2, P_5, P_8, \cdots, P_{3n-1}\}$
    - $G_{*,3} := \{P_3, P_6, P_9, \cdots, P_{3n}\}$
- Let $G_{i,j} := G_{i,*} \cap G_{*,j}$.
- Count sequences $\sigma_1, \sigma_2, \cdots, \sigma_n \in \Sigma_3$ so that place P_i at $G_{j,\sigma_i(j)}$.

H: Fair Problemset



Proposer: Nah Jeounghui, Setter: Nah Jeounghui

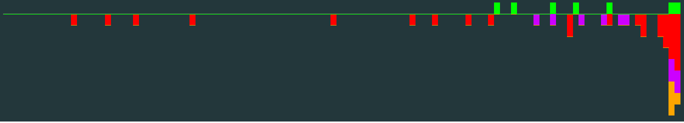
Solution:

- Observe that
  - Use $(1\,2\,3)$; $(\alpha - x)$ times.
  - Use $(2\,1\,3)$, $(1\,3\,2)$, $(3\,1\,2)$; x times.
  - Use $(2\,3\,1)$, $(3\,1\,2)$; $(\beta - x)$ times.

  , where $n = 3q + r$; $\alpha = q + \delta_{1,r}$; $\beta = q + \delta_{2,r}$; $0 \le x \le q$.
- The answer is

$$A_n = (\alpha!)^3\,(\beta!)^6 \sum_{i=0}^{q} \binom{n}{\alpha - i; i; i; i; \beta - i; \beta - i}^3$$

H: Fair Problemset



Proposer: Nah Jeounghui, Setter: Nah Jeounghui

Solution:

- To compute all A_*'s, you can
  - Make a generation function for A_* and use FFT,
  - Find recurrence relation of $B_q := \sum_{i=0}^{q} \binom{q}{i}^3$ and use B_*'s, or
  - Compute A_* for $n \leq 200$ naïvely and find p-recursive relation of order ≤ 20.

Complexity: $\mathcal{O}\left(10^6\right)$.

B: Badge Relay

Proposer: Lee Seung Yong, Setter: Lee Seung Yong

Problem: Calculate the minimum time for selected among n employees to cross a bridge with a single badge (capacity 2). Process q range queries.

Solution:

- Greedy strategy is optimal: Use 1 or 2 fastest to shuttle.
- For $n \geq 3$ people with times $t_1 \leq t_2 \leq \cdots \leq t_n$, the answer is

$$(n-k-2)t_1 + (2k+1)t_2 + \sum_{i=3}^{n-2k} t_i + \sum_{i=0}^{k-1} t_{n-2i}$$

  , where $m := \max\{j : t_j \leq 2t_2 - t_1\}$ and $k := \lfloor (n-m)/2 \rfloor$.
- Apply Mo's algorithm to the queries.
- Maintain a segment tree to sum odd-ranked values ($1^{\text{st}}, 3^{\text{rd}}, \cdots$) efficiently.

Complexity: $\mathcal{O}\left(q\sqrt{n}\log n\right)$.

Thanks to:

Problem proposers & setters & reviewers:

- Cho Hwan-Gue
- Kim Soohwan
- Kim Jae-Hoon
- Bae Sang Won
- Shin Chan-Su
- Sim Jeong Seop
- Han Yo-Sub
- Koo Jaehyun
- Kim Yeonghyeon
- Kim Jaehong
- Nah Jeounghui
- Youn Gyojun
- Lee Seung Yong

System operators:

- Hong Seokju
- Park Suhyun
- Shin Seungwon