# ICPC Asia Hong Kong Regional Contest Analysis

By Zhejiang University

2024.12.22

# Overview

This contest will be used as a stage (hopefully, the next stage) of Universal Cup Season 3.

Please do not discuss or distribute this analysis in public places before this stage of Universal Cup.

# Overview

| | Easiest | | | | | | | | | | | | Hardest |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Idea | C | K | L | H | G | B | F | E | D | A | J | I | M |
| Coding | C | K | A | G | L | B | J | F | H | E | I | D | M |
| Summary | C | K | L | G | H | B | F | E | A | D | J | I | M |

# C. The Story of Emperor Bie

Shortest Judge Solution: 343 Bytes

# Description

You can build a sequence according to the following process:

1. The sequence initially contains an arbitrary positive integer.
2. In each step, append an integer to the left end or the right end which is no greater than the other end.

Given the final sequence, find all possible initial positions.

- $n \leq 5 \cdot 10^5$.

# Solution

- The possible initial positions are all $p$ such that $a_p = \max_{i=1}^{n}\{a_i\}$.
- Necessity: You can never add an element greater than the initial element, so you must start with an element equal to the maximum value.
- Sufficiency: Consider the operations in reverse order, you can always keep the initial element not deleted if it is one of the maximum elements.
- The time complexity is $\mathcal{O}(n)$.

# K. LR String

Shortest Judge Solution: 959 Bytes

# Description

You are given a string $s$ consisting of "`LR`". Answer $q$ queries, the $i$-th query contains a string $t_i$, find out whether $s$ can be changed to $t_i$ by processing several following operations: Choose a string character, delete the closest character on the direction according to the character.

- $\sum |s|, \sum |t_i| \leq 10^6$.

# Solution

- There are three conditions that $t_i$ must meet:
  1. $t_i$ is a subsequence of $s$
  2. If $s$ begins with "`R`", then $t_i$ must begin with "`R`"
  3. If $s$ ends with "`L`", then $t_i$ must end with "`L`"
- We can prove that these conditions are sufficient by greedy. Find the next "`L`" and "`R`" for each position, we can check each answer in $\mathcal{O}(|t_i|)$.
- The complexity is $\mathcal{O}(|s| + \sum |t_i|)$.

# Problem L. Flipping Paths

Shortest Judge Solution: 1991 Bytes

# Description

Given a $n \times m$ black-white grid. You can flip a path of cells from $(1, 1)$ to $(n, m)$ going only right and down. Find a way to paint all the cells the same color within $400$ operations.

- $n, m \leq 200$.

## Solution

- Firstly, we can enumerate the final color of the grid.
- For each $2 \times 2$ subgrid, the top right cell and the bottom left cell can be flipped together using two paths.
- Therefore, for all minor diagonals, the number of wrong-colored cells must have the same parity.

## Solution

- Next, we can consider each of the major diagonals one by one, trying to flip these major diagonals into the correct color.
- The path can be chosen by going to the top left corner of the diagonal directly below it, then choosing the positions with the wrong color to go up to the designated diagonal. Finally, go from the bottom right corner of the diagonal directly to the bottom right corner of the grid.
- The time complexity is $\mathcal{O}(n^2)$.

# Solution

- There is another solution by greedily choosing an upper-right "wrong color monotonic stack" each time and flipping a path through these cells.
- Such a path can be found by prefix sums.
- The time complexity of this alternative solution is $\mathcal{O}(n^3)$.

# G. Yelkrab

Shortest Judge Solution: 1205 Bytes

# Description

Given $n$ strings $s_1, s_2, \ldots, s_n$, calculate $\bigoplus_{j=1}^{i} f(i,j)$ for all $1 \leq i \leq n$.

$f(i,j)$ denotes the maximum total value of choosing several groups of size $j$ among the first $i$ strings. The value of a group is the length of the longest common prefix of strings in this group.

- $n \leq 5 \cdot 10^5$.
- $\sum |s_i| \leq 10^6$.

# Solution

- Let's try to calculate the answer for some strings and group size $k$. Repeatedly group the $k$ strings that the value is as large as possible gives us the correct answer.
- Proof: Assume that in a better grouping, the $k$ strings have LCP of length $w$, and are in groups $G_0, G_1, G_2, \ldots, G_m$, where $w \geq val(G_1) \geq val(G_2) \geq \ldots G_m$, and $G_0$ contains the strings does not form any group.
  If we put the $k$ strings in group $G_1'$, and for group $2 \leq i \leq m$, use strings in group $j < i$ to pad the size to $k$ to form groups $G_2', G_3', \ldots, G_m'$, we have $val(G_i') \geq val(G_i)$, which contradicts with grouping into $G_1, G_2, \ldots, G_m$ is a better grouping.

# Solution

- We can speed up this process by inserting all strings into a trie. Then enumerate nodes from deep to shallow, group strings as much as possible, and pass the rest of the string to the parent node.
- If $sz_x$ strings are in the subtree of node $x$, there will be $\left\lfloor \frac{sz_x}{k} \right\rfloor$ groups formed in the subtree of node $x$. So the answer is $\sum\limits_x \left\lfloor \frac{sz_x}{k} \right\rfloor$.
- When $sz_x$ increases by 1, $\left\lfloor \frac{sz_x}{k} \right\rfloor$ increases by 1 for all $k$ that $k$ is a divisor of $sz_x$. Preprocess the divisors of $x$ for all $1 \leq x \leq n$ will help us update the answer quickly.
- The time complexity is $\mathcal{O}(\sum |s_i| \log n)$.

# H. Mah-jong

Shortest Judge Solution: 3034 Bytes

# Description

Given $n$ Mah-jong tiles, find the number of consecutive intervals that the tiles in the interval form a "Mah-jong".

The "Mah-jong" contains several *Pong* and *Chow*.

- $n \le 10^5$.
- $a_i \le 8$.

# Solution

- To form a "Mah-jong", three same "Chow" $\{x, x+1, x+2\}$ can be changed into "Pow" $\{x, x, x\}, \{x+1, x+1, x+1\}, \{x+2, x+2, x+2\}$.
- So each Mah-jong can be represented as $c_x (0 \leq c_x \leq 2)$ "Chow"s $\{x, x+1, x+2\}$ and several "Pongs".
- Enumerate $3^6 = 729$ possible multiset of "Chow"s, assume that there are $d_i$ tiles $i$ in this plan. If a Mah-jong contains $e_i$ copies of tile $i$, then it must satisfy that:
  1. $d_i \equiv e_i \pmod 3$
  2. $e_i \geq d_i$

# Solution

- 1. $d_i \equiv e_i \pmod 3$
  2. $e_i \geq d_i$
- Enumerate the right endpoint $r$, there exists an integer $u$ that for the second condition, it is satisfied if and only if $l \leq u$.
- Use two pointers to find $u$ for each $r$, and use buckets storing prefix sums to count the possible $l$ satisfying the first condition.
- The time complexity is $\mathcal{O}(3^6 n)$ if implemented carefully.

# B. Defeat the Enemies

Shortest Judge Solution: 1597 Bytes

# Description

There are $n$ enemies, each with shield $a_i$ and health $b_i$. You can deal $1 \le x \le k$ damage to all enemies with cost $c_x$. The shield will absorb the damage first, then the health will be damaged. Exceeded damage on the shield will not affect the health. Find the minimum total cost to defeat all enemies.

- $m \le 10^4$.
- $k \le 100$.

# Solution

- Enumerate the total damage $s$.
- For an enemy $(a_i, b_i)$, it is defeated if after an attack, the damage falls in range $[a_i, s - b_i]$.
- Let $f_i$ be the minimum cost when the damage is $i$. This can be calculated in $\mathcal{O}(mk)$ time. Notice that when $s \geq k + \max\{a_i + b_i\}$, all the enemies will definitely defeated. So there are only $\mathcal{O}(k)$ total damages need to be enumerated. Notice that to calculate the number of ways, you need to enumerate $s$ to $2k + \max\{a_i + b_i\}$.
- The time complexity is $\mathcal{O}(mk^2)$.

# F. Money Game 2

Shortest Judge Solution: 2181 Bytes

# Description

There are $n$ players sitting in a circle. The $i$-th player has $a_i$ deposits initially. In each round, a player can give half of his deposit (rounded down to an integer) to an adjacent person. Let $f(i)$ be the maximum deposits player $i$ can have, calculate $f(x)$ for all $1 \leq x \leq n$.

- $n \leq 5 \cdot 10^5$.
- $0 \leq a_i \leq 10^9$.

# Solution

- Assume people are numbered from $0$ to $n-1$ clockwise. WLOG, assume we are calculating $f(0)$, in the optimal plan we can find some $y \neq 0$, letting people numbered $x \leq y$ give the deposits to the people on the counterclockwise direction one by one, and letting people numbered $x > y$ give the deposits to the people on the clockwise direction one by one.
- One conjecture is that the people numbered $\lceil \log a \rceil < x < n - \lceil \log a \rceil$ have no influence on $f(0)$. This conjecture is wrong because there are counterexamples like $0\ 1\ 1\ 1\ \ldots 2$.

# Solution

- If we only consider $n$ people passing deposits to the left one by one, the total deposit passed at last will be $\left\lfloor \frac{\sum_{i=1}^{n} a_i \cdot 2^{n-i}}{2^n} \right\rfloor$. When the $a_x(\lceil \log a \rceil < x)$ differs, the deposit passed will change at most 1!
- Let $L(x, y)$ be the deposit passes to $x$ considering the $y$ closest people in the clockwise direction, there will be only $\mathcal{O}(\log a)$ different value for $L(x, \cdot)$. Defining $R(x, y)$ similarly, we can find $f(x)$ by using two pointers on $L(x, \cdot)$ and $R(x, \cdot)$. Calculating $L(x, \cdot)$ from $L(x + 1, \cdot)$ by storing the minimum $y$ for the $\mathcal{O}(\log a)$ values lead us to the $\mathcal{O}(n \log a)$ time solution.

# E. Concave Hull

Shortest Judge Solution: 3122 Bytes

# Description

There are $n$ points on the plane. Calculate twice the sum of the area of all the Concave Hulls of the given set of points.

Concave Hull： Each of the $n$ points is either inside the polygon or one of the vertices of the polygon. Exactly one of the interior angles of the polygon is greater than $\pi$, while all the other angles are less than $\pi$.

- $n \leq 2000$.

# Solution

- Enumerate the point of the reflex angle, assume it is $p_i$. It can be seen that $p_i$ can not be one of the points on the convex hull.
- If we sort all points by polar angle respect to $p_i$, the reflex angle must consists of two adjacent points.
- If the other two points are not adjacent, the polygon will self-interact.

# Solution

- Run Graham Algorithm to find the convex hull. Enumerate the vertex of the reflex angle. We can run Graham on polar angle order towards both directions. This runs in $\mathcal{O}(n^3)$.
- If the Concave Hull intersects with Convex Hull, we can stop the process as the answer remains the same. We can run Graham on every interval of polar angle order between edges on Convex Hull once to calculate the answer.
- The time complexity is $\mathcal{O}(n^2 \log n)$.

# A. General Symmetry

Shortest Judge Solution: 983 Bytes

# Description

An integer sequence $s_1, s_2, \ldots, s_m$ is called $k$ - symmetric if and only if $|s_i - s_{m-i+1}| \leq k$ for all integers $i$ ($1 \leq i \leq m$).

Given a sequence of length $n$, find the longest $k$ - symmetric consecutive subsequence centered on each possible place.

- $n \leq 2 \cdot 10^5$.

# Solution

- Fix the center, binary search to find the longest $k$-symmetric consecutive subsequence.
- We can find all the answers in $\mathcal{O}(n \log n)$ if we can access $F(l, r)$ in $O(1)$, where $F(l, r)$ denotes whether $[l, r]$ is $k$-symmetric. We have
  - $F(l, l) = \text{True}$, and
  - $F(l, r) = F(l+1, r-1) \land [|a_l - a_r| \leq k]$ $(l < r)$.
- Denote $G(i, j) = [|i - a_j| \leq k]$, then we have $F(l, r) = F(l+1, r-1) \land G(a_r, l)$.
- Iterate $r$ from 1 to $n$, then $F(\cdot, r) = F(\cdot, r-1) >> 1 \land G(a_r, \cdot)$.
- Hence we can compute all pairs of $F(l, r)$ via Bitset in $\mathcal{O}(\frac{n^2}{w})$.

# Solution

- Two more bonus optimization tricks:
  1. Instead of storing $\mathcal{O}(n^2)$ pairs in memory, we can only store $F(\cdot, i \log n)$ for all possible values of $i$ ($1 \le i \le \frac{n}{\log n}$). Binary search for the value of $\lfloor \frac{ans}{\log n} \rfloor$, then extend with no more than $O(\log n)$ steps to fix the answer.
  2. In reality, the length of the Bitset $F(\cdot, r)$ is $r$ instead of $n$. So we can cut down the constant by half.
- The total time complexity is $\mathcal{O}(\frac{n^2}{2w})$.

# D. Master of Both IV

Shortest Judge Solution: 3815 Bytes

# Description

You are given a tree with $n$ vertices. Each vertex initially has an empty sequence. Support $q$ queries of the two types:

1. `A` $x\ y\ z$: Append $z$ to all sequences of vertices on the shortest path from $x$ to $y$.
2. `D` $x\ h$: Assume the sequence on $x$ is $z_1, z_2 \ldots, z_m$, find the smallest $t$ such that there exists $c_t, c_{t+1}, \ldots, c_m$ where $c_k \in \{0.5, 1\}, c_k \cdot c_{k+1} \geq 0.5$ and $\sum_{k=t}^{m} c_k \cdot z_k < h$.

   Answer each query of the second type.

- $n \leq 5 \cdot 10^5, q \leq 3 \cdot 10^5$.

# Solution

- Let's solve the problem for $n = 1$.
- We can use Dynamic Programming to calculate the minimum total damage. Let $f_{i,0/1}$ denote the minimum total damage when considering the last $i$ attacks and the $i$-th attack's damage is not reduced/reduced by half. The transition can be written as multiplying a matrix of $2 \times 2$, where $\{+, \times\}$ is replaced by $\{\min, +\}$.
- By building a segment tree where the indices are the queries and maintaining the product of matrices for each interval, we can do binary search on segment tree to quickly calculate the answer.

## Solution

- Solving the problem on a tree is similar. We still build segment tree where the indices are queries and try to maintain segment tree for each vertex.
- Differentiate the operation on the path between $x$ to $y$ into inserting operations on $x$ and $y$, deleting operation on the parent of $\mathrm{LCA}(x, y)$, the operations on $x$ is the union of all operations in the subtree of $x$.
- Run Depth First Search on this tree, and merge all segment trees of the children of $x$, then apply operations on vertex $x$. This allows us to do binary search on segment trees to find the answers.
- The time complexity is $\mathcal{O}(n + q \log q)$.

# J. Reconstruction

Shortest Judge Solution: 1650 Bytes

# Description

You are given two trees, $T_1$ and $T$. For each $u$ ($1 \leq u \leq n$), determine whether $T$ can be a possible "centroid tree" of $T_1$ if $T$ is rooted at vertex $u$. Note that for a tree, we can choose an arbitrary point as its "centroid".

- $n \leq 5 \cdot 10^5$.

# Solution

- For every edge $(u_i, v_i)$ in $T$, assume that this edge is cut. Then, every point (except $u_i$ and $v_i$) on the path between $u_i$ and $v_i$ in $T_1$ must remain in the same component of $T$, while the root of $T$ must be in the other component. Thus, there are $n-1$ conditions.
- The points that satisfy all these conditions can be the root of $T$.
- The time complexity is $\mathcal{O}(n \log n)$.

# I. Ma Meilleure Ennemie

Shortest Judge Solution: 3138 Bytes

# Description

Given a method to divide $1, 2, \ldots, n$ into groups and color them with one of $1, 2, \ldots, m$. Find the number of different colorings.

- $n, m \leq 10^{18}$.

# Solution

- The groups are actually some modular arithmetics. When there are $n$ elements in a group and $m$ colors to color the elements, the method actually works like this:
- Grouping: Choose a divisor $d$ of $n$, divide the problem into $d$ subproblems where there are $\frac{n}{d}$ elements and $\left\lfloor \frac{m}{d} \right\rfloor$ colors.
- Coloring: Choose an integer $1 \le c \le m$ where $\gcd(c, n) = 1$ to color the elements.
- In the grouping session, instead of choosing a divisor $d$, we can choose a prime divisor $p$.
- The number of ways to color a group in one color is $\sum_{c=1}^{m} [\gcd(c, n) = 1] = \sum_{d|n} \mu(d) \left\lfloor \frac{m}{d} \right\rfloor$ by the inclusion-exclusion principle.

# Solution

- We need inclusion-exclusion principle because that there are cases like: dividing a group of size $18$ into groups of size $3$ is counted twice, since it is equivalent to first divide into groups of size $6$ and then divide into groups of size $3$, or first divide into groups of size $9$ and then into groups of size $3$.
- We can also use the inclusion-exclusion principle to count. Assume $n = \prod p_i^{k_i}$ and $f(n, m)$ denote the answer for $n$ elements and $m$ colors.
- We have $f(n, m) = \sum_{d|n} \mu(d) \left\lfloor \frac{m}{d} \right\rfloor - \sum_{d|n, d>1} \mu(d) f(\frac{n}{d}, \left\lfloor \frac{m}{d} \right\rfloor)^d$

# Solution

- By only enumerating divisors $d$ such that $\mu(d) \neq 0$, there are $\sum_{d|n} 1$ states and $\sum_{d|n} 2^{\omega(d)}$ transitions.
- $\sum_{d|n} 2^{\omega(d)}$ is at most about $3 \cdot 10^7$ for $n \leq 10^{18}$.
- Use Pollard-Rho to factorize $n$ and carefully enumerate all transitions, and use Pohlig–Hellman algorithm and BSGS to speed up power calculation, the time complexity is $\mathcal{O}\left(n^{\frac{1}{4}} \log n + \sum_{d|n} 2^{\omega(d)}\right)$, which is capable of passing the problem.

# M. Godzilla

Shortest Judge Solution: 9492 Bytes

# Description

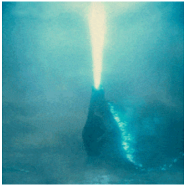
Given a grid of $n$ rows and $m$ columns. Godzilla will visit every cell exactly once. When visiting cell $(i, j)$, Godzilla must do one of the following three things:

1. Make no attack.
2. Spend $e(i, j)$ units of energy to make a horizontal attack to hit all the cells in the same row, causing $d(i, j)$ units of damage.
3. Spend $e(i, j)$ units of energy to make a vertical attack to hit all the cells in the same column, causing $d(i, j)$ units of damage.

# Description

For $k = 0, 1, 2, 3$, try to maximize $\sum d(\cdot, \cdot)$, subject to:

1. $(\sum e(\cdot, \cdot)) \bmod 4 = k$.
2. Every cell is hit by exactly one horizontal attack.
3. Every cell is hit by exactly one vertical attack.



- $n, m \leq 75$.

# Solution

- Build a graph with $n + m$ vertices, denoting each row and each column.
- For a potential attack made at cell $(i, j)$, add an undirected edge $(d(i, j), e(i, j))$ between row $i$ and column $j$.
- We should select $n + m$ edges and decide the directions for them, such that the in-degree of each vertex is exactly $1$.
- This is equivalent to selecting $n + m$ edges to form a pseudoforest, such that every connected component has at most one cycle.
- Hence we are required to find a basis of bicircular matroid.

# Solution

- Let's denote the optimal basis satisfying $(\sum e(\cdot,\cdot)) \bmod 4 = k$ as $OPT(k)$.
- We can find the global optimal basis $\mathcal{B}$ greedily like Kruskal.

**Theorem**

Under $\mathbb{Z}_4$, when $OPT(k)$ exists, we can always find $OPT(k)$ such that $|\mathcal{B} \setminus OPT(k)| \leq 3$.

- Enumerate and remove $t$ $(1 \leq t \leq 3)$ edges from $\mathcal{B}$, there will be exactly $t$ tree components, other components are all 1-trees. Select $t$ more edges to form a new pseudoforest.

# Solution

- Let's run a kernelization to reduce the number of candidate edges.
  1. For every tree component, find the edge with maximum $d(\cdot,\cdot)$ from it to any 1-tree for $e(\cdot,\cdot) = 0, 1, 2, 3$, respectively.
  2. For every tree component, find the edge with maximum $d(\cdot,\cdot)$ whose two endpoints are all inside it for $e(\cdot,\cdot) = 0, 1, 2, 3$, respectively.
  3. For every pair of tree components, find the top two edges across them for $e(\cdot,\cdot) = 0, 1, 2, 3$, respectively.
- Hence we will get a kernel problem with extremely small size, we can easily compute $OPT(\cdot)$ in $\mathcal{O}(1)$.

# Solution

- Finally let's see how to fetch top edges from different components after removing $t$ edges in $\mathcal{B}$.
- Note that each component in $\mathcal{B}$ is 1-tree.
- Ignore any edge on the circle, run DFS on the tree.

**Lemma 1**

A component can always be represented as $\mathcal{O}(t)$ consecutive intervals in DFS order.

# Solution

- Enumerate two intervals $[a, b]$ and $[c, d]$, we are looking for the top two edges, one of whose endpoint is located at $[a, b]$ and another is located at $[c, d]$.
- Let's treat an edge $(u, v)$ as a 2D point $(dfn_u, dfn_v)$.
- It turned out to be a 2D RMQ problem.

**Lemma 2**

2D RMQ can be answered in $\mathcal{O}(1)$.

- The total time complexity is $\mathcal{O}((n + m)^3)$.

# Thank you!