ICPC 2025 Asia Bangkok

Contest Report

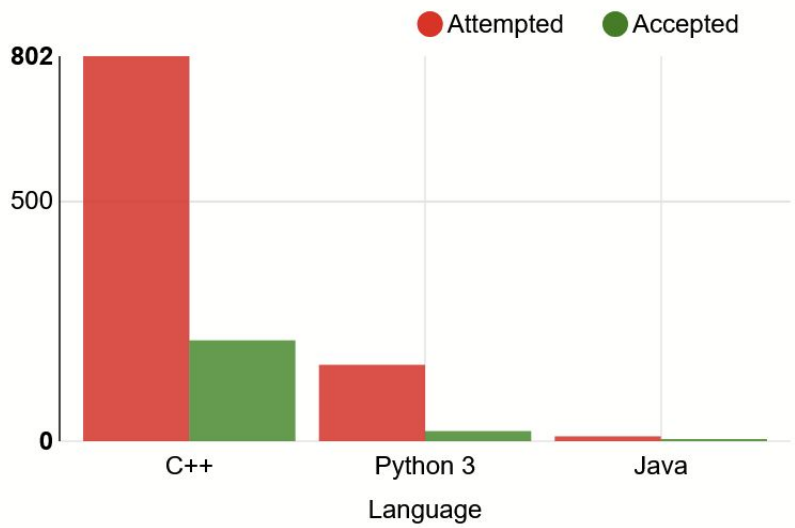
Technical Report

Contest Summary

53	Teams (68 contest stations)
990	Total Submissions
248	Accepted Submissions
15	Submissions that wait more than 5 seconds
	The longest wait is 12 seconds.
10	Judgehost Machine

Language Summary

- C++: Used by 43 Teams
  solve 12 problems
- Python3: Used by 16 Teams
  solve 3 problems
- Java: Used by 1 Team
  solve 1 problem



Incident

Start Exactly on 8:00

During the first 5 minutes

- Public Clarification & Announcements of last-minutes changed in Task Statement
- 3 PCs was reported some hardware glitches, flickering monitor, temporarily freezing.
  - One team request for extension but the judge ruling is to reject.

At 8:30

- Detected mix-up color of the balloon for problem A and B. Announcement were made.

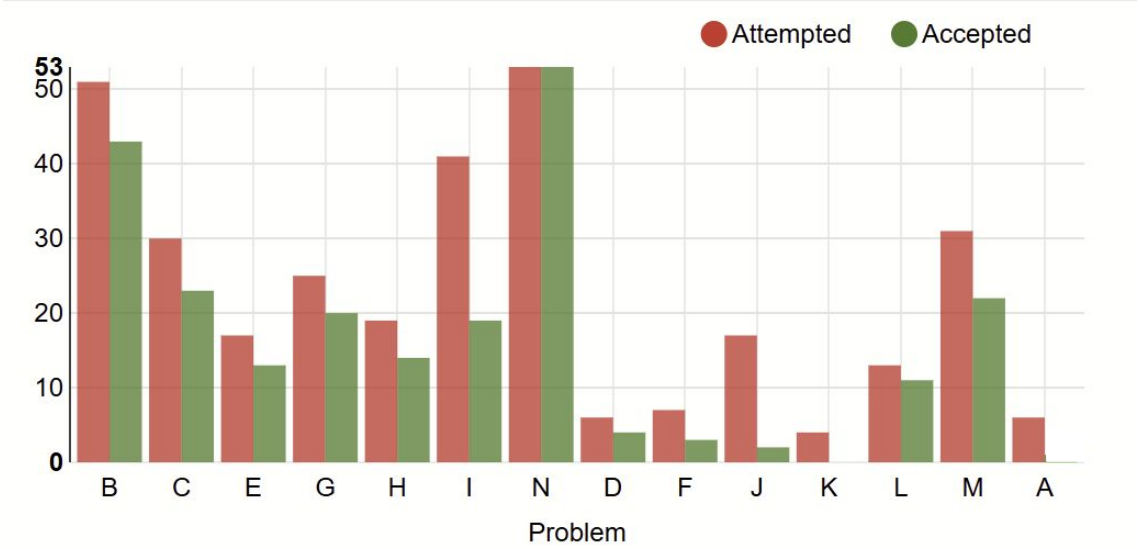
Problem Set Report

Problem Set Incident

- 9:32 A Clarification request of problem G regarding the case when there is no positive satisfactory was asked. We published public clarification
  - No submission was affected by this clarification
  - No announcement was made.
- 11:41 Question regarding Problem A was asked. The judges detect ambiguous in the statement and a public clarification was made.
  - No submission of this problem is submitted before this announcement.
  - No announcement was made

Problem Summary

- Easy Problem
  - N, B
- Medium Problem
  - C, E, G, H, I, L, M
- Hard Problem
  - D, F, J
- Luv
  - K, A



A: Among Us

Author: Mattanyu Tangngekkee

A: Among Us

If in the hidden permutation p[u] = v, then both u and v cannot be both not-**sus** at the same time.

You can ask random permutations many times, each time tracking the pair (i,j) that are both not-**sus** and mark them impossible to form an edge in the permutation graph. And track the crewmate that is always sus, as they are the one accused themself.

After enough queries, there would (very likely) be only two candidates for p[i] for any i. After that, you only need one more query to determine the direction of all cycles.

B: Bring it to Back

Author: Natapong Sriwatanasakdi

B: Bring it to Back

Case N = 2

- Parity: [1, 2], [2, 1], [1, 2], …

Case N > 2

- Case M < N:
  - [N, N -1, …, N - M + 1, 1, 2, 3, …, N - M]
- Case M >= N:
  - [N, N -1, N -2, …, 3, 2, 1]
    - N = 3, M = 3: [3, 2, 1] => [2, 1, 3], [1, 3, 2], [1, 2, 3]

B: Bring it to Back

C: Challenge to the Reader

Author: Mattanyu Tangngekkee

C: Challenge to the Reader

X can be written as 1+2+...+N - 2 x (a1+a2+a3+...+ai) where a1,a2,.... > 1

For example

-6 = 1+2+3-4+5-6-7 = (1+2+3+4+5+6+7) - 2 x (4+6+7)

So in the other way you can ask that is there a subset S of {2, 3, ..., N} such that sum(S) = (N(N+1)/2 - X)/2

You can slowly check from the smallest N that |X| <= N(N+1)/2

C: Challenge to the Reader

There are 3 impossible cases (you have to check the next N instead):

N(N+1)/2 - X is odd

(N(N+1)/2 - X)/2 = 1

(N(N+1)/2 - X)/2 = N(N+1)/2 -1

Otherwise you can greedily select the value for set S from largest to smallest (beware of the case where the last remaining value is 1, then use the second largest instead)

D: Dungeons and Dragons

Author: Poonyapat Sriroth

D: Dungeons and Dragons

First part, calculate grundy number for all 0 <= i <= n

From, Sprague-Grundy theorem

grundy[i] = mex(grundy[i - 1], grundy[i - 2],.., grundy[i-p[i]])

for each value of grundy number, maintain the maximum index that the value is presented MX[v]

grundy[i] = minimum integer K such max MX[K] < i-p[i]

This can be done by either Fenwick Tree or Segment tree

D: Dungeons and Dragons

Second part, use the polynomial $A = a_0x^0+a_1x^1+...+a_nx^n$

where a_i is the number of indexes that grundy value = i

$B = b_0x^0+b_1x^1+...+b_nx^n$ where b_i is the number of indexes that j such that (j + k > n or grundy[j] = grundy[j+k]) and grundy[j] = i

The answer is coefficient of 0-th degree of (A^R-B^R) by multiplying polynomial with XOR degree instead of summation
This can be done by XOR convolution(FFT)
Time Complexity: O(nlog(n) + nlog(R))

E: Elena and Travel Passes

Author: Natapong Sriwatanasakdi

E: Elena and Travel Passes

- Sort edges by pass level
- Do Floyd-Warshall but uses an edge as a medium instead of vertex

```
for (u, v, p, w) in edges
    for i in range 1 to N
        for j in range 1 to N
            sp[i][j] = min(sp[i][j], sp[i][u] + sp[v][j] + w)
```

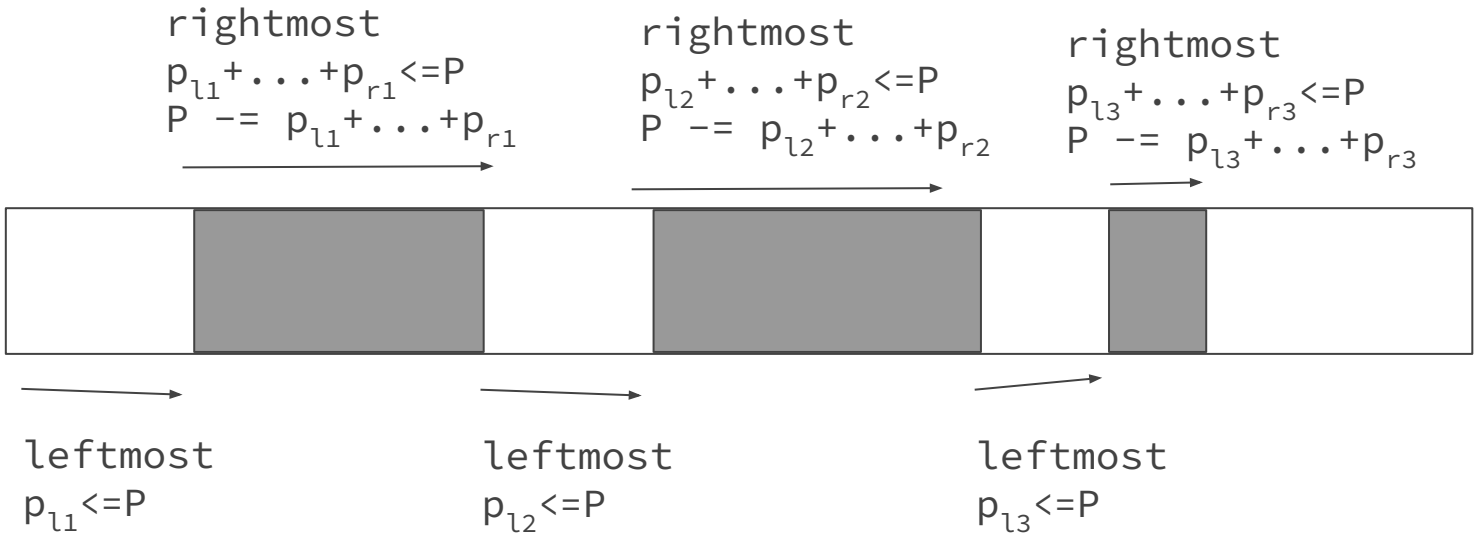
E: Elena and Travel Passes

- When all edges in a pass level is used
  - Memorize the maximum sssp from each city to answer type 1 query
  - Find the minimum value among vertices tie breaking by vertex number to answer type 2 query
- Binary search the memorized value
- Note: Should process only after all pass level is used

F: Festival Stroll

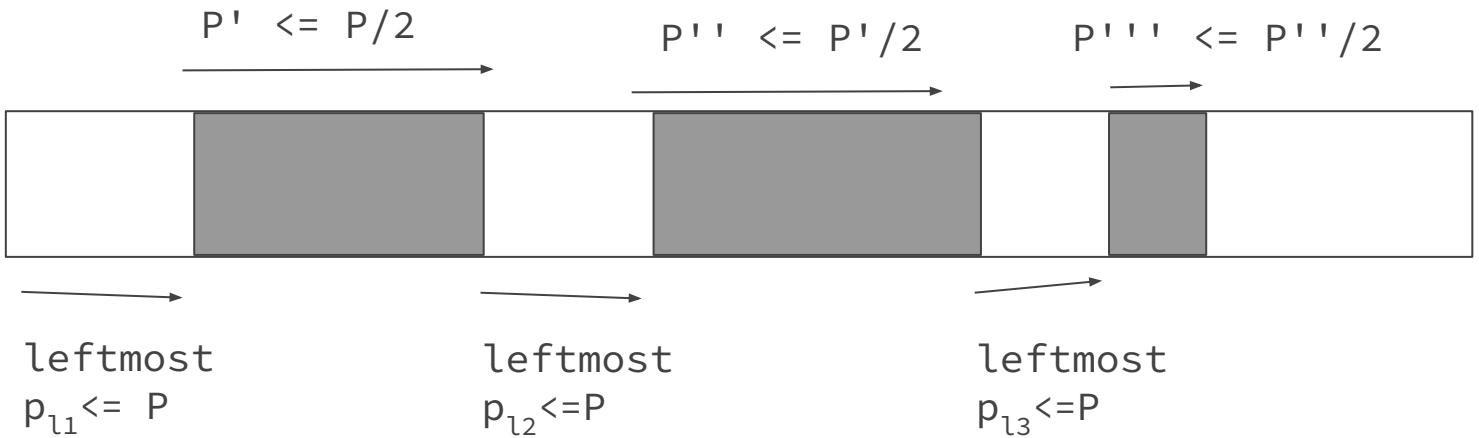
Author: Natapong Sriwatanasakdi

F: Festival Stroll



How to optimize this?

F: Festival Stroll



- Update/query with fenwick tree or segment tree

G: Galactic Adventure Agency

Author: Papangkorn Apinyanon

G: Galactic Adventure Agency

---

important observation

$|x_i-x_j|+|y_i-y_j|+|z_i-z_j|=\max((-1)^u (x_i-x_j)+(-1)^v (y_i-y_j)+(-1)^w (z_i-z_j))$

for all integers $0 <= u, v, w <= 1$

It's enough to maintain the maximum value of tree depth +

$(-1)^u (x_i)+(-1)^v (y_i)+(-1)^w (z_i)$ for every mask (u, v, w) in subtree and do dp on tree

Time complexity: $O(2^d n)$ where d is the number of dimensions

H: Home Workout Playlist

Author: Natapong Sriwatanasakdi

H: Home Workout Playlist

- DP[n][m] = min(s) | S_m=n, s = A[S_m] - A[S_{m-1}]
- Naive: DP[n][m] = A[n] - max(DP[n -i][m -1])
  - A[n -i] + DP[n -i][m -1] < A[n]
- Optimize: Use fenwick tree/segment tree to keep suffix max value
- Time complexity: O(n lg(MaxValue) * SubsequenceLength)
  - Subsequence length is in O(sqrt(MaxValue))
  - ~ 447

I: ICPC Extractor

Author: Mattanyu Tangngekkee

I: ICPC Extractor

You can use binary search on answer and greedily pick the substring as follows:

Iterate from front to last: greedily pick substring "IC" X times.

Iterate from last to front: greedily pick substring "PC" X times.

Match "IC" and "PC" from front to last.

If any of this step is impossible, then X is greater than the answer.

Otherwise, X is less or equal to the answer, and the binary search continues.

There is also a linear time algorithm that works too.

J: Joyeuse

Author: Araya Luanseng

J: Joyeuse

Sort the array, then consider the following.

$$\sum_{i=2}^{n}\sum_{j=1}^{i-1}\sqrt{a_i+a_j}=\sum_{i=2}^{n}\sqrt{a_i}\sum_{j=1}^{i-1}\sqrt{1+\frac{a_j}{a_i}}=\sum_{i=2}^{n}\sqrt{a_i}\sum_{j=1}^{i-1}f(\frac{a_j}{a_i})$$

where $f(x)=\sqrt{1+x}$ is evaluated in range $x\in(0,1]$.

We can **approximate** f(x) with Maclaurin series in the form $a_0x^0+a_1x^1+a_2x^2+a_3x^3+a_4x^4+.....$

J: Joyeuse

To evaluate $a_0x^0+a_1x^1+a_2x^2+a_3x^3+a_4x^4+.....$ quickly

precalculate prefix sum of x^i and evaluate till the k-th degree which yield enough precision

The judge solution uses 3 Maclaurin series at x = 0,0.5,1 and evaluate up to 5^{th} degree

Note: Single Maclaurin may not have enough precision

Time complexity: O(nlog(max(a)) with high constant

K: Kickshot Tournament

Author: Mattanyu Tangngekkee

K: Kickshot Tournament

Let K be the number of steps you move from the starting position to reach one of the corner.

K can be figured out with extended euclid algorithm.

The number of coin that is crossed two times is equal to the number of pair (x,y) such that x>=1 y>=1 and

(M-1)x + (N-1)y <= K which can be count using **floor-sum**.

L: Laser

Author: Atittarn Buathep

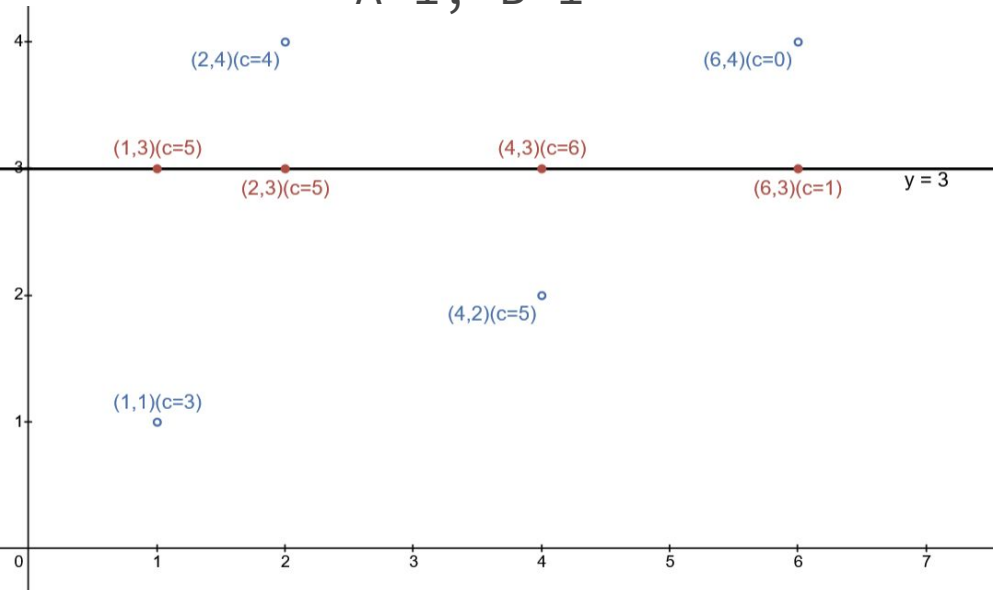
L: Laser

The problem is formulated as
minimize($A * |s_i - x_j| + B * |t_i - y_j| + c_j$)

To choose these point efficiently, we will project each point to a line that gradually move from y=0 to $y=10^9$

L: Laser

A=1, B=1



After that, we can use 2 1D segment trees to calculate minimum distance from left and right side of each point

Time Complexity: O(nlog(n))

M: Meticulous Manipulation

Author: Jonathan Irvin Gunawan

M: Meticulous Manipulation

Do operation in reverse order

For i = n to 1

Find the position of i in the array and move suffix from i to prefix and remove i from the array

By performing this operation, the order in the array doesn't change if we view the array as circular, so we just need to maintain the correct head of array

To find the position of i and remove index quickly from array, we can use Fenwick Tree or Segment Tree

Time Complexity: O(nlog(n))

N: No Distance is Too Far Apart

Author: Mattanyu Tangngekkee

N: No Distance is Too Far Apart

Alice is at position A+1

Bob is at position N-B

There are |(A+1)-(N-B)|-1 people between them.