wf24 report

nvmdava
Bui Hong Duc Nyamdavaa Amar

rama_pang
Rama Aryasuta Pangestu

September 2024

1 Preparation

We do a few contests for practice.

Anecdote: we need to find contests whose difficulty is roughly the same as that of WF — UCup (Universal Cup) is often too hard, while anything else is often too easy — so we come up with a few methods:

- make a userscript to hide the actual problems solved while looking at the scoreboard, essentially to find out the number of problems solved by top teams.
- then the difficulty of the problem set can be roughly judged by how many problems the X-th team solved during the contest.
- remember to include virtual participants, because some contests may only have weaker teams — which would lead to unreasonably easy contest.
- one way to measure the difficulty is to find out if there are other ICPC WF teams (from universities which regularly gets medal) who have done the contest, and judge the difficulty based on their performance.

Overall, we did like 4 to 5 (?) contests in person, and 4 contests online on weekends after `nvmdava` went to London. Practicing online sometimes leads to low morale (some members of the team often fell asleep if alone), but in 2 person practice (without `nvmdava`, that is) our performance doesn't suffer much. Judging based on practice performance, we are somewhat hopeful that medal is possible if everything goes well.

We prepare the TRD very last minute, cramming centroid decomposition and weighted general matching last minute before the deadline. Then we stay at COM until almost 11pm (last bus) to print the notebook. `rama_pang` buys some file folder to contain it (sorry 💸)

Our final notebook can be found at `https://github.com/rama-pang/icpc-trd/releases/tag/icpc-wf-2024`.

We got a little funny issue where rama_pang tries to include "cutting edge" (C++20) techniques (e.g. zip() and enumerate()), but I decide they're not worth the time spent typing them and throw them out. Though this deep-dive-ish to C++20 std::ranges features end up being somewhat slightly useful to type binary search faster in contest...

We retain YComb (as always) and included a more generic operator<< and operator>> for "ranges" (vector, tuple, pairs, array, set, map, and so on which is a std::ranges::range) though (though we didn't type this in contest).

2 Before the contest

We have two chances to get used to the keyboard — one during ICPC Huawei Challenge, and another during Dress Rehearsal. This was quite helpful to get typing speed up on an unfamiliar keyboard.

During Dress Rehearsal, we noticed that all the samples are conveniently downloaded to ~/Desktop/samps/ for us, so we just move it to the home directory. We also noticed that the interactive problem conveniently has a testing_tool.py, which is very helpful for debugging in the actual contest.

Post-contest, it turns out there are quite a few teams who didn't notice the existence of testing_tool.py for the interactive problem (Problem L), which might contribute to the low number of solves as it's quite inconvenient to test the problem manually and implementation heavy to implement your own "interactor".

3 During the contest

(written with Hong Duc in the first person voice)

As usual, we start with me reading from the front, rama_pang reading from the back, and nvmdava typing the template. I read problem A, recall that it comes from CS4261, but did not immediately recall the solution. I move to B, worked on it for some 10 minutes and solve it.

I modify the .vimrc template a bit to read the input file from %:r/1.in instead of i. It took a bit of vim documentation lookup to figure out I need :r.

Then rama_pang takes over the computer to implement F. Meanwhile nvmdava tries to solve problem E, but it is actually more complicated than it looks. He tries to explain the idea to me, and I note that we need some sort of knapsack built in the solution, which might be done by digit-by-digit DP. But a rough calculation of time complexity shows digit-by-digit DP might be too slow, so we give up on it.

rama_pang got the idea for problem I (some tricky and prone-to-edge-cases dynamic programming) and started implementing, but I notice problem C is relatively simple, so I interrupt rama_pang to implement it. After problem C is accepted, rama_pang continued implementing problem I and got accepted.

Afterwards, because we have no solved problems, I recall the solution from CS4261, go ahead and implement problem A. `nvmdava` and `rama_pang` works on problem D (I previously read the statement but did not come up with any good idea), and after some time they come up with the solution.

Meanwhile I got into a little trouble because the straightforward implementation has too large time complexity. I need to comment out something like 50 lines of code (?), implement the two pointer idea and submit.

Unfortunately this gets TLE, so I give up the computer for `rama_pang` to implement D (which got AC quite quickly). After calculating the time complexity I see that the binary search is called too many times, then I derive the closed-form formula (solve a quadratic equation) and code it in. There's a corner case where the quadratic equation degenerates to a linear equation which needs to be taken care of, but fortunately I notice the issue early.

At this point `rama_pang` has some ideas on problem J (some implementation-heavy dynamic programming in segment tree states) and problem L (some implementation-heavy interactive problem), and decided to go for L.

Next, `rama_pang` works on L while we try to come up with ideas for H and K on paper. Initially some of us misread the statement of H (that the demon can move along the wall, which is not the case). When the demon still cannot move, it simplifies the problem significantly, and after some further simplification we reduces it to a Gaussian elimination of a 400×40000 matrix modulo 2. . . which sort of fits in the time limit, or so we think until we realize the large dimension might be up to 16000000, which definitely won't fit.

We had a little debate whether to implement it anyway (for all we know the number of distinct points possible may not be that large) or put it off, I decide to put it off because it's too risky.

Meanwhile, `rama_pang` generates a test case by hand, and the answer appears to be wrong. I tell `rama_pang` to look at our idea for problem H, while I debug his code. After some debug printing, I notice that the answer on the generated test case is actually correct.

```
1 5
..#.#
1 1 N
```

The printed answer by the program is "no", I realize this is because the first region has rotational symmetry. We just submit the solution, but unfortunately we get WA.

rama_pang mentions something to the effect of "maybe a 3×3 square. . . ?", and I quickly realize that the problem is rotational symmetry does not necessarily kills the hope of determining the location — you just need to move to the center.

```
3 3
...
...
```

```
...
1 1 N
```

Then I work out how to modify the code to detect that case as well as optimizing the number of steps needed in the DFS phase, and `rama_pang` implements the path-finding with BFS.

We try to generate some other test cases by hand, such as

```
5 5
.....
.###.
.#.#.
.###.
.....
1 1 N
```

after everything passes, we submit and got another AC. Before getting the AC, I decide to print out the solution because "print delivery is so slow, might as well print in case it WA" — turns out this would be unnecessary.

(Actually I'm not sure if we ever printed J (the whole problem was handled by rama_pang), but we did print L like that.)

`rama_pang` continues to implement J while me and `nvmdava` discusses solution for K. `nvmdava` knows how to use dynamic programming, which helps with coming up with the solution. I proceed to implement it after `rama_pang` is done with problem J — turns out this would be our last AC in the contest.

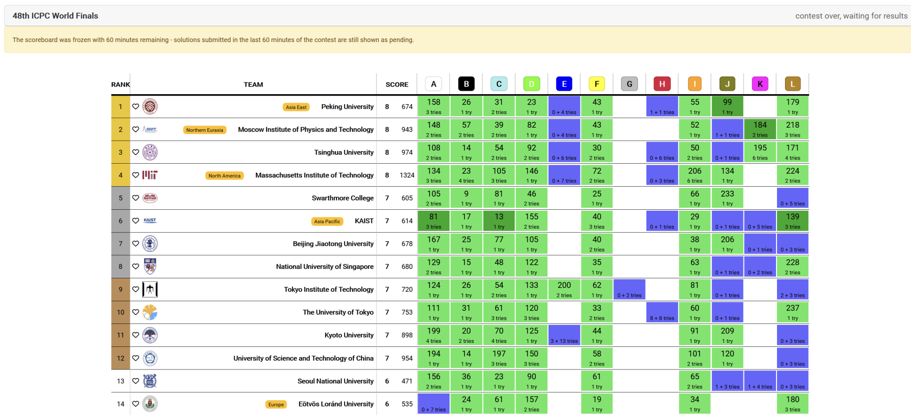
There was quite a few corner cases for problem K — namely, what if the first pile is empty, what if the third pile is empty, what if the state after the big "move" is actually valid — but fortunately we notice these quickly. We implement it but got WA anyway.

`nvmdava` meanwhile tries to work on it on the background, and notice another corner case. Unfortunately this is not sufficient to get AC.

In the end, we print out our solution, intending to debug it later.

4 After the contest

In total, I solved A+B+C+(half of L), `rama_pang` solved (half of D)+F+I+J+(half of L), `nvmdava` solved half of D, and tried to work on E, H, and K. E is unfortunately a difficult problem that is deceptively easy. `nvmdava` tried to explain to me the idea within the contest, but I think it'd need at least digital DP to solve the knapsack part — and even that doesn't fit in the time limit. I feel `nvmdava`'s idea for K *sounds* correct, but unfortunately we did not manage to implement it during contest.



48th ICPC World Finals — contest over, waiting for results

The scoreboard was frozen with 60 minutes remaining - solutions submitted in the last 60 minutes of the contest are still shown as pending.

RANK	TEAM	SCORE		A	B	C	D	E	F	G	H	I	J	K	L
1	Asia East Peking University	8	674	158 3 tries	26 1 try	31 2 tries	23 1 try	0 + 4 tries	43 1 try		1 + 1 tries	55 1 try	99 1 try		179 1 try
2	Northern Eurasia Moscow Institute of Physics and Technology	8	943	148 2 tries	57 2 tries	39 2 tries	82 1 try	0 + 4 tries	43 1 try			52 1 try	1 + 1 tries	184 2 tries	218 3 tries
3	Tsinghua University	8	974	108 2 tries	14 1 try	54 2 tries	92 2 tries	0 + 6 tries	30 2 tries		0 + 6 tries	50 2 tries	0 + 1 tries	195 6 tries	171 4 tries
4	North America Massachusetts Institute of Technology	8	1324	134 3 tries	23 4 tries	105 3 tries	146 1 try	0 + 7 tries	72 2 tries		0 + 3 tries	206 6 tries	134 1 try		224 2 tries
5	Swarthmore College	7	605	105 2 tries	9 1 try	81 1 try	46 2 tries		25 1 try			66 1 try	233 1 try		0 + 5 tries
6	Asia Pacific KAIST	7	614	81 3 tries	17 1 try	13 1 try	155 2 tries		40 3 tries		0 + 1 tries	29 1 try	0 + 1 tries	0 + 5 tries	139 3 tries
7	Beijing Jiaotong University	7	678	167 1 try	25 1 try	77 1 try	105 1 try		40 2 tries			38 1 try	206 1 try	0 + 1 tries	0 + 3 tries
8	National University of Singapore	7	680	129 2 tries	15 1 try	48 1 try	122 1 try		35 1 try			63 1 try	0 + 1 tries	0 + 2 tries	228 2 tries
9	Tokyo Institute of Technology	7	720	124 1 try	26 1 try	54 2 tries	133 1 try	200 2 tries	62 1 try	0 + 2 tries		81 1 try	0 + 1 tries		2 + 3 tries
10	The University of Tokyo	7	753	111 1 try	31 1 try	61 3 tries	120 3 tries		33 2 tries		8 + 8 tries	60 1 try	0 + 1 tries		237 1 try
11	Kyoto University	7	898	199 4 tries	20 2 tries	70 4 tries	125 1 try	3 + 13 tries	44 1 try			91 1 try	209 1 try		0 + 3 tries
12	University of Science and Technology of China	7	954	194 1 try	14 1 try	197 3 tries	150 3 tries		58 2 tries			101 2 tries	120 1 try		0 + 3 tries
13	Seoul National University	6	471	156 2 tries	36 1 try	23 1 try	90 1 try		61 1 try			65 2 tries	1 + 3 tries	1 + 4 tries	0 + 3 tries
14	Europe Eötvös Loránd University	6	535	0 + 7 tries	24 1 try	61 1 try	157 2 tries		19 1 try			34 1 try			180 3 tries

Figure 1: Scoreboard before freeze. Can also be found in `https://web.archive.org/web/20240919110101/https://scoreboard.icpc.global/2024/scoreboard/`.

Over the course of the first four hour of the contest, we managed to consistently stay in medal range, with the scoreboard ending up looking as above. After calculating the penalty difference to other teams, Prof. Steven was quite confident that we get at least a medal.

On the closing ceremony, it turns out we managed to get Rank 6 — a silver medal. This is the first ICPC WF medal gained by NUS after two decades of trying, and also the second ICPC WF medal and first silver medal gained by any university in South East Asia (the first medal being a bronze medal by University of Engineering and Technology — VNU at ICPC WF 2021, which was Hong Duc's old team).

48th ICPC World Finals — final standings

RANK	TEAM	SCORE		A	B	C	D	E	F	G	H	I	J	K	L
1	Asia East Peking University	9	935	158 3 tries	26 1 try	31 2 tries	23 1 try	4 tries	43 1 try		241 2 tries	55 1 try	99 1 try		179 1 try
2	Northern Eurasia Moscow Institute of Physics and Technology	9	1212	148 2 tries	57 2 tries	39 2 tries	82 1 try	4 tries	43 1 try			52 1 try	249 2 tries	184 2 tries	218 3 tries
3	Tsinghua University	9	1218	108 2 tries	14 1 try	54 2 tries	92 2 tries	6 tries	30 2 tries		6 tries	50 2 tries	244 1 try	195 6 tries	171 4 tries
4	Asia Pacific Tokyo Institute of Technology	9	1322	124 1 try	26 1 try	54 2 tries	133 1 try	200 2 tries	62 1 try	2 tries		81 1 try	240 1 try		282 5 tries
5	KAIST	8	868	81 3 tries	17 1 try	13 1 try	155 2 tries		40 3 tries		1 try	29 1 try	254 1 try	5 tries	139 3 tries
6	National University of Singapore	8	934	129 2 tries	15 1 try	48 1 try	122 1 try		35 1 try			63 1 try	254 1 try	2 tries	228 2 tries
7	Beijing Jiaotong University	8	960	167 1 try	25 1 try	77 1 try	105 1 try		40 2 tries			38 1 try	206 1 try	1 try	282 1 try
8	The University of Tokyo	8	1031	111 1 try	31 1 try	61 3 tries	120 3 tries		33 2 tries		16 tries	60 1 try	278 1 try		237 1 try
9	Seoul National University	8	1112	156 2 tries	36 1 try	23 1 try	90 1 try		61 1 try			65 2 tries	287 4 tries	5 tries	254 3 tries
10	Zhejiang University	8	1166	178 1 try	29 1 try	66 2 tries	130 1 try		107 1 try			84 1 try	258 1 try		294 1 try
11	North America Massachusetts Institute of Technology	8	1324	134 3 tries	23 4 tries	105 3 tries	146 1 try	7 tries	72 2 tries		3 tries	206 6 tries	134 1 try		224 2 tries
12	Swarthmore College	7	605	105 2 tries	9 1 try	81 1 try	46 2 tries		25 1 try			66 1 try	233 1 try		5 tries
13	Kyoto University	7	898	199 4 tries	20 2 tries	70 4 tries	125 1 try	16 tries	44 1 try			91 1 try	209 1 try		3 tries
14	University of Science and Technology of China	7	954	194 1 try	14 1 try	197 3 tries	150 3 tries		58 2 tries			101 2 tries	120 1 try		3 tries
15	Europe University of Zagreb	7	982	119 3 tries	7 1 try	42 3 tries	253 2 tries		39 2 tries		4 tries	126 3 tries			216 2 tries
16	Jagiellonian University in Krakow	7	1000	290 5 tries	32 2 tries	62 1 try	82 1 try		60 2 tries		172 2 tries	122 3 tries	5 tries		1 try

Figure 2: Final scoreboard.

A few days after contest, it turns out that we missed problem K by a few lines. Another team which has the same idea as `nvmdava` for problem H also managed to get accepted post-contest (their team got TLE in contest due to missing a slight optimization `nvmdava` mentioned).