

# Introduction

The 2018 ICPC Asia Jakarta Regional Contest is a competitive programming contest held in BINUS University, Jakarta, Indonesia. This contest is part of a network of multiple regional competitions around the world of which the winners will get a chance to participate in the 2019 ICPC World Finals. In competitive programming contest, we are given several data structure and algorithms problems and we are only judged by the number of problems solved and the time we need to solve the problems.

Throughout November and December 2018, NUS sent eight teams to three different overseas regional contest in Asia Pacific and South East Peninsula region, and Jakarta site first one. There are three NUS teams sent to this site. Our team comprises of:

1. Phan Duc Nhat Minh, from NUS School of Computing, as coach.
2. Jerrell Ezralemuel, 2nd year undergraduate student from NUS School of Computing, majoring in Computer Science, as contestant.
3. Samuel Henry Kurniawan, 4th year undergraduate student from NUS Faculty of Science, majoring in Life Science, as contestant.
4. Jonathan Mulyawan Woenardi, 4th year undergraduate student from NUS Faculty of Science, majoring in Mathematics, as contestant.

## Day 1: Arrival Day

Our flight to Jakarta was "smooth" (we were quite rushing since we almost late to arrive in the airport) and we arrived at Soekarno-Hatta International Airport. Since it was already late afternoon, we had a lunch first at the airport. We then took a taxi to go to BINUS Square, the host university guest hotels where we stayed for four days. There were no scheduled activities for the day, so we had a rest and had a dinner at nearby mall. We also finalized our reference sheet and printed them at night. We enjoyed the day as we ate a lot of delicious Indonesian food.

## Day 2: Opening Day

The opening day was quite a normal day. We went to BINUS University using bus provided by the committee. We finished registration and got to meet other teams, e.g. teams from University of Indonesia and BINUS University. We then proceeded to the opening ceremony.

After attending opening ceremony and lunch, we go to practice contest. We solved three out of four questions in the first 45 minutes, and our code was quite buggy in the last one. Instead of finding the bug, we preferred to use the time to test the time limit using python and c++. We also tried the printing services. The committee ensured the printing services will be fast, because last year the printing services had some problems. Nevertheless, we familiarized ourselves well with the contest environment.

There was no more activities after the practice session. At the end of the day, we had practice contest debrief and strategy discussion for the contest day tomorrow.



Figure 1. Team photo at the opening day.

### Day 3: Contest Day (Contest)

We were sleeping peacefully in the night before the contest and had a good breakfast. The contest starts at 10am GMT+7. In the contest part, we will have three parts, where each part is the contest experience from each individual perspective, because we think there are a lot of things that we can share from different perspective.

### Jonathan's Perspective

I would prefer to refer to myself in 3rd person, even though this is supposed to be my perspective.

#### The Early Game (Minute 0 to Minute 90)

As we usually did in our weekly practice, Samuel read from the beginning and Jonathan read from the end. We gave Jerrell extra time to prepare certain code and configurations at the beginning. He will start reading from the middle afterwards. Ideally, Samuel and Jonathan should be fast enough to give solution for easy problems for Jerrell to implement.

In the first few minutes, Jonathan attempted problem L and found a greedy solution that looks convincing, but he cannot prove its correctness. As Jonathan is a very careful person (sometimes overly careful), he did not immediately accept his own solution and check many possible cases. Meanwhile, Samuel and Jerrell had already one wrong-answered attempt on problem A. Samuel requested Jonathan to continue with problem A.

Problem A is a tricky problem involving edit distance of strings. Given a binary string of length  $N$ , we are asked to construct another binary string of length  $N$  with edit distance greater than or

equal to  $N/2$ . The solution was pure constructive, and it took some time for Jonathan to finalize the solution. Fortunately, our team did not stuck because of this problem. Samuel AC-ed problem I followed by us AC-ing problem A. We had two problems AC-ed in 22 minutes and we were not too far behind other teams.

After listening problem D's description from Samuel, Jonathan found a 'general' solution for the problem D. It is 'general' in the sense, it does not work for small  $N$  (which is  $N = 2$ ). We did not realised this until we submitted our solution later. There are two options at that time:

1. To implement our solution for problem L, an unproven greedy solution
2. To implement our solution for problem D, a simple ad-hoc solution with some edge cases (thus some if-else branching).

Jonathan decided to let Jerrell implement problem D first while he prove his solution for problem L. We got Wrong Answer verdict on our first attempt for problem D. This was the moment we realised that we need a specific solution for case  $N = 2$ . However, we still got Wrong Answer. Jonathan then swapped problem to focus on with Jerrell. Jerrell implement solution for problem L, and Jonathan re-check the correctness of his solution for problem D. We then AC-ed problem L in minute 75. We fixed a bug and AC-ed problem D in minute 81.

Looking back, there were some mistakes regarding our handling on problem D and L:

1. The greedy solution for Problem L is correct from the beginning. Jonathan should not be too meticulous and insist on proving the correctness of his solution. This argument is supported by the fact that there are already many teams that solved L.
2. Implementation of solution of Problem D should not be delegated to Jerrell as it could easily become a messy solution with a lot of edge cases.

Nevertheless, we were relieved that we managed to solve the first 4 problems. We saw the scoreboard and we understood that these 4 problems are the easy ones, and the next challenge is to enter the medium range problems.

#### The Mid Game (Minute 90 to Minute 220)

We started to attempted different problems. Looking from the scoreboards, H and J seems to be the next problems other teams attempted. Some teams solved F too. Jonathan first had a look at problem H and J. Problem J is a problem related to string and data structures where the solution could varies from string techniques to even graph. Jonathan tried to model problem J into some graph problem, but failed to have anything result. Problem H is the kind of pure data structure problem that Jonathan is not very good at. Therefore, Jonathan focus on problem F instead, while Samuel and Jerrell worked on problem H and J.

Jonathan first found solution for problem F involving binary search the answer (BSTA) and simulation of the main process. The implementation involved so many details in the simulation aspect. In addition to the numerical answer of the BSTA process, we are also required to print out sequence of numbers that represent the simulated process that represent the answer. This adds to the complexity of the implementation. We did not immediately implement the solution of problem F, and Jonathan joined in attempting problem J. However, due to the pressure of not solving anything, we decided to start implementing our solution for F. Jonathan is not confident in coding the solution, thus Jerrell coded it.

The implementation of problem F seems to take a very long time. Fortunately, after so many weird attempts (most will not pass the time limit), Jonathan found a solution for J. This time, the solution should pass the time limit and Jerrell was confident in implement this. This is a huge reinvigoration for Jerrell and Jerrell started implemented solution for problem J. Jonathan started to read other problems such as problem C and G because he had no idea for problem H. Samuel

continued with problem H. Problem C is a slight variation of classic De-Bruijn sequence, and Jerrell said he knew how to generate the regular De-Bruijn sequence. After reading problem G, Jonathan calculated the theoretical time complexity of the most naive solution, and found that it is  $O(N^3 \log N)$ . With  $N = 500$ , Jonathan thought his solution should exceed time limit.

After finishing the implementation of solution of problem J, Jerrell tested it using sample test case and got a wrong answer. Samuel worked together with him to debug it. Fortunately, we managed to find the bug and AC-ed the problem in minute 219. Even though it took us too long time to finally AC-ed another problem, it is still a huge relief for us. Looking back, one big mistake that happen is Jonathan not implementing problem F on his own. The whole process took a lot of focus and energy from Jerrell, of which he could use to solve other problems.

#### The Late Game and (Minute 220 to Minute 300)

We only had 80 minutes left. To summary, what we had at the moment is:

1. We have some ideas on problem C. Later on, we learned that we just need to glue them together to get the solution.
2. We had unfinished implementation of solution of problem F.
3. We have a naive solution for problem G. Later on, we learned that the intended solution is the naive solution, carefully implemented to not have a big constant factor on the speed performance.
4. We had so many desperate attempts on problem H.

We first tried to combine our ideas on unsolved problems. However, in the end we chose to continue the implementation of problem F. There are some parts of the implementation that are not clear, so Jonathan coded some parts. In the end, we managed to finish a rather messy implementation made by Jerrell and Jonathan. Unexpectedly, it got a Wrong Answer verdict.

Looking back, some of our mistakes are:

1. We were working on too many problems at the same time. We should just focus on 2 or 3 questions at a time.
2. We could have tried to attempt problem G using the naive solution. At the contest, we were quite indecisive on what we to attempt.

#### Conclusions

There are so many things that we learned from this contest, as presented in the previous subsections. In our last few practices before ICPC Singapore, we will try to fix our mistakes and hopefully can achieve better result in ICPC Singapore. I am particularly proud of my team as this is our first ICPC regional experience and there has been improvement since we started doing weekly practices.

## Jerrell's Perspective

### In Early Game

Actually our early start is quite good with solving problems as projected although below my expectation. But we are still on the trajectory such that we still have the chance to be on the top 12 we are doing well. As I am the main coder of this group I did most of our problems at start I config the machine to streamline the coding process and familiarizing myself with the keyboard at the same time. On the early game we solve ADIL in under 90 minutes although not the best start, it's a good start. it took me too much time coding D to my standard as the first solution given is not the right solution and it keeps me back and forward finding up why, but it's okay.

The second "HALF"

After this everything went... bad (?) And the time comes for me to code the problem F in which is the problem that I don't really like, simulation and which gets complicated as in filled with "ifs" in short amount of time. And I should've give this problem to samuel instead of doing it myself I get lost easily as the number of ifs gets high, from time to time I always get away from this kind of problem and I forced myself doing this. Although I am pretty sure Samuel would've done this faster as he is quite weird as he likes these kind of problems and doesn't really care about writng a bunch of branches here and there. And I am getting back and forth F and L because of "new ideas" from the team but on solving L we encounter a simple bug that took us too much time on that as well. But it's fine anw. It wasn't that bad and everything went downhill from there.

Reflection

I should've train more on doing this kind of problem and maybe learn to delegate task to others better. As I realized the solution of C and I could've maybe solved H in short amount of time as the solution wasn't that hard. The problem C And I should've solved the problems instead of doing menial task of "F" as I realized after exiting the room and I solved problem "D" without any use of random and what not. I have the knowledge of C the De Bruijn sequence, and I am more than familiar on how to construct the graph and doing the Hierholtzer to find the Eulerian walk on the De Bruijn graph resulting on the De Bruijn sequence itself. And solving C should be super easy and I like this kind of problem as this kind of problem comes with generality and I can code it once and there won't be lot and lot of branches.

I haven't actually find my balance in coding, I don't have a good sense on how messy can I code on contest, I used to code to clean on contest which is a bad thing and a good thing on itself, but sometime when the code gets messy I couldn't cope up with the bad code and it annoys me too much. I hate seeing my code and I see codes with a lot of branches as dirty and emotionally consuming for me.and I would be exhausted as I finish doing the problem.

The other problem I think me and Samuel we sort of have different ideas on how to code and I am very efficient on how do I write my code, and very concise in writing the code, and sometime our code annoys each other he couldn't understand my code because it's too concise or I dont' know why, or I like to give an elegant data inside my array / ds / anything just to reduce the number of branches and simplifies the code. But I don't know why it annoys them in a certain way. And I sometimes hate seeing Samuel code as it is too messy for me and it drives me nuts to see how the data is handled and how much do I need to clean up to just debug it. Hence we have to decide early on who should code and can't just give the code to each other.

But in the end, I think I should have be able to code all of the solution during the ICPC if I prepped well.

Well, I'm a good guy so here's a TL;DR

I hate simulation, I shouldn't have done it, pick someone else. I should've solving another problem instead of coding another problem to get at least 8 AC on this contest which is very feasible. I am not strong enough to cope with "if-fy" problem , but I should've.

I don't really have much to say, but hey! It was fun. Thanks !

## Samuel's Perspective

Intro

So actually I am the burden on the team (I am very happy that they did not kick me from the team). As a non-cs student that algorithms experience is only from CS1010S and CS2040C and a bit of self-CP3-read, usually I am unable to get the idea by myself nor code non-complete search

code from medium question onwards. So my role in the team is to help setting the initial pace, reading the question from first to last, helping explain the question, solving and coding the few easiest question if necessary, and help throwing ideas and debugging (as the talking rubber duck for rubber duck debugging) for later part. We are also decided not to go for the time because it will become very stressful and hinder us for thinking clearly and just try our best to win in terms number of questions AC.

### Before the Contest

Ok, I feel I slept very well as I take the telephone cord off so no one can wake me up at 430 am (the previous day someone morning-call me at 430 am and I cannot sleep anymore after that). Since I can eat very fast, after breakfast I go to some quiet place alone to start warming up with kattis (I solved one 2040 level question with satisfying time imo). I also drank coffee to make myself in the most excited state and I think I am well prepared for the contest now.

### In the Contest

When the timer says "Start!", immediately I open the first question and read the question A. Question A is a very tricky question about edit distance for binary number, I was thinking a solution to just flip the bits. Since it looks easy and all the sample test cases are about flipping the bits (and if it is wrong we only get 1 penalty and if it is right we might get first to solve so its worth trying). So I immediately told jerrel (jerrel is the one handling the keyboard, switching people handling the keyboard may need some time) "read the characters, if it is 0 print 1, else print 0", and we got WA of course, we fell to the judges' trap. Then I realized that 010101 and 101010 is just two edit distance instead of 6. After thinking a while, I got the idea if the 0 and 1 is not the same amount, all 0 or all 1 will be helpful. But I have not thought of what happen if the number 0 and 1 is the same amount. At the same time, Jonathan' seems to have solved I and explained the solution to Jerrel, so I told him (I am thinking he should be able to solve this easily since he is one of the best mathematicians I know) "read A" and I told him my partial solution.

Few seconds after that, we AC-ed I and we are continuing to read the questions. Few minutes later, Jonathan told Jerrel the A solution and we AC-ed A as well. He also told his 90% sure solution of L, and Jerrel is coding for a while for that. Then suddenly he said to me "read D", after I read, he has found the solution by putting some edge cases then the general cases and discussed that to me and I think that is workable. So we told Jerrel "code this first" since this is an obvious solution. To be honest I think this is my mistake, this is an "if-else" problem that Jerrel does not like to code, I was thinking for offering myself to code instead to not letting Jerrel get fatigued physically and mentally to code multiple "if-else" condition, since this is a 1010-level question once the solution is found . But I was stupidly thinking "ok, jerrel can code faster than me so I should not hinder the team by making myself code". Then we submit D and got a WA. After some time, Jonathan's realized that our L solution is correct because most of other teams have solved it as well and Jerrel said he prefer code L first since it is "less if-else".

After AC-ing L, we continuing with D, and we found one case that we have not covered, and after that, we AC-ed D as well. I was thinking OK, 1.5 hours 4 AC, its a good start. Then since we have not solved any other problems, I was continuing to read the problems while trying to think the solution of H and J, since those two problems look like the easier problems since most teams AC-ed it. Then Jonathan found solution of F, since it just a simulation problem, and Jerrel is coding the backbone of F for some time until Jonathan found the solution of J and Jerrel said he prefer code J first since he think it is easier to code. I was finishing reading all the problems around this stage and trying to think for H. My solution for H is a complete TLE with  $O(nk)$  complexity so I was trying to think for this (and occasionally other problems as well).

After around 45 minutes, Jerrel said he is done and ask me to run the sample test case while he is eating stuffs outside. Then the sample test case is wrong, and I told him that when he came back to the seat. After around another 30 minutes, I was wondering why Jerrel looks very tired and start to stare the screen quite blankly for some time. I thought he is still in the middle of implementing the bug repair. So I did not pay much attention while continue thinking for the solution of H. After another 30 minutes, I realized that for the past 30 minutes Jerrel does not code a lot, most of the time he stares the screen while scrolling, so I was asking "what's wrong" and he said the sample test case is buggy when ran in the code, and he cannot found the bug. Then I offered myself to become the rubber duck to debug. After a very short while, thanks to rubber duck debugging he found he mistyped  $i+1$  to  $i$ , causing the DP to fail. Then we repair that and got an AC. I was very sad knowing we only AC-ing one problem in the past two hours, and I was cursing myself not to help as the rubber duck for rubber duck debugging earlier.

For the last 90 minutes, because we did not solve any other question, Jonathan suggested that we focus on F since it is just a simulation problem. Around this time Jerrell said that problem C is a De Bruijn, but since the solution is not formalized yet, we still continue to do F. For the last 45 minutes, our F is buggy and Jonathan took over the coding position since Jerrel looks very tired already. We solved the sample test cases in the last few minutes and submitted it at min 298, and it is wrong answer from some edge cases. Then we gave up and go out from the contest room when the contest ended.

#### Self-Reflection

My biggest regret is that I do not have much experiences since I just started taking CS1010S in my year 3 semester 1. Actually I was intending to do a lot of practice for the semester, but I was miscalculating my workload as I have some non-skippable commitment, and somehow I was accepted at CS3216 (I was just randomly applying since it is nothing to lose to apply and I prepared my workload with the assumption that my CS3216 application got rejected), which takes most (if not all) my individual practice time. However, this is my last year (I think, if I can graduate on time) joining ICPC and I do not have another chance to fight. Hence, although I really regretting my own incompetence, I was also grateful to have this kind of experience in the ICPC.



Figure 2. Inside the contest area (Jerrell is getting some snacks outside).























RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J	K	L
1	 <b>Deobureo Minkyu Party</b> Korea Advanced Institute of Science and Technology	11 1261	2/16	1/266	1/66	2/109	2	5/144	2/156	3/61	1/8	2/97	1/93	1/45
2	 <b>earlybird</b> Kyoto University	10 1338	1/65	0	2/233	1/23	0	1/81	1/282	2/175	1/11	1/140	1/252	1/36
3	 <b>3Sophomores</b> National University of Singapore	10 1581	1/31	0	9/260	5/73	0	1/280	4/179	2/103	1/9	1/83	2/168	1/55
4	 <b>CMP</b> Korea Advanced Institute of Science and Technology	9 1032	2/25	0	2	2/33	0	1/162	2/204	2/112	1/11	1/75	2/252	2/38
5	 <b>Supir Tayo</b> Universitas Indonesia	9 1405	2/133	0	3/278	3/52	0	4/248	2/238	1/169	2/6	1/59	0	1/22
6	 <b>map</b> University of Engineering and Technology	8 727	1/36	0	0	2/49	0	1/163	1/201	4/118	1/4	1/52	7	1/24
7	 <b>NTUSECURE</b> Nanyang Technological University	8 888	3/12	0	0	3/45	0	1/139	7/247	2/99	1/9	1/70	9	2/27
8	 <b>YouR Lovely JatengPRiDe</b> Universitas Indonesia	8 1151	3/72	0	0	1/90	0	1/198	7/275	3/108	1/16	2/162	7	1/10
9	 <b>semoga ayas juara versi empat</b> Universitas Indonesia	8 1218	2/95	0	0	5/74	0	2/238	3/256	1/130	1/10	2/189	0	1/46
10	 <b>Kth-D Hyperprism</b> Bina Nusantara University	8 1274	6/55	0	1/248	2/50	0	20	7/288	4/134	1/8	1/70	0	5/41
11	 <b>1T</b> Nanyang Technological University	8 1409	4/45	0	0	1/29	1	6/181	5/282	1/203	1/10	1/128	3	7/171
12	 <b>ReFreshPHD</b> National University of Singapore	7 532	2/14	0	3	2/85	0	2/152	8	1/73	1/10	1/110	0	1/28
13	 <b>1S22S1</b> Bina Nusantara University	7 899	3/33	0	1/265	2/60	0	4	1	3/208	1/11	2/132	0	3/30
14	 <b>Yang Penting AC</b> Bina Nusantara University	7 1028	4/49	0	0	2/65	0	6/285	20	1/217	1/10	2/178	1	1/24
15	 <b>semoga ayas juara versi tiga tit</b> Universitas Indonesia	7 1154	6/95	0	0	5/115	0	2/299	0	7/233	1/8	1/58	9	1/26
16	 <b>Ganteng Ganteng Tukang Sula</b> Institut Teknologi Bandung	6 629	2/23	0	0	3/54	0	4	0	4/172	1/11	1/131	0	2/98
17	 <b>I See the One</b> Institut Teknologi Bandung	6 730	3/35	0	0	2/92	0	0	3	4/232	2/12	2/155	0	1/44
18	 <b>mad as rabbits</b> Universitas Indonesia	6 736	3/97	0	0	2/64	0	3	0	3/276	1/11	1/120	0	2/48
19	 <b>Welshite</b> Bina Nusantara University	6 1024	6/88	0	0	1/123	0	0	0	4/272	1/11	1/204	0	4/106
20	 <b>Sultan Halim</b> National University of Singapore	5 465	2/22	0	0	3/81	0	1	0	0	1/8	1/219	0	1/75

Figure 3. Final scoreboard (we are at the twentieth place).

## Day 3: Contest Day (After Contest)

After the contest finished, we had a late lunch, since it was already 3pm. However, the contest problems still haunted us because we kept discussing about our attempts and solutions with other NUS teams. After everyone had lunch, we then proceeded with the closing ceremony. Everyone was eagerly waiting for the winners announcement. As our team did not get into the top 12 teams, we only went to the stage to take photos with other NUS teams.

After the closing ceremony, there was one more scheduled activity that had been a tradition for ICPC Jakarta site: a gala dinner with a lot of Indonesian foods. We also took some photos with friends before we went back to hotel.



Figure 4. All NUS teams.

## Day 4: Departure Day

We did not join the excursion as our flight was scheduled at 11am GMT+7. We went directly to the airport after having breakfast at BINUS Square. The flight was smooth and we arrived at Changi Airport and went back to NUS. Even though our journey in ICPC Jakarta has ended, we still had ICPC Singapore on December 13. We will continue our practice, and see you on our next report!

## Closing and Acknowledgements

We would like to thank NUS School of Computing and our various corporate sponsors, including our diamond donors, Indeed and Sea, as well as our bronze donors, Jump Trading and DRW, for all the administrative and financial support.

We would like to thank Dr. Steven Halim and Phan Duc Nhat Minh for guiding us in Jakarta and for the training and coaching.

We would like to thank you all the committee of The 2018 ICPC Asia Jakarta Regional Contest for the wonderful opportunity to participate in this contest. We truly enjoyed the food and the contest prepared by the committee. We would also like to give special thanks to the photography team that provides the good photos inside this report.