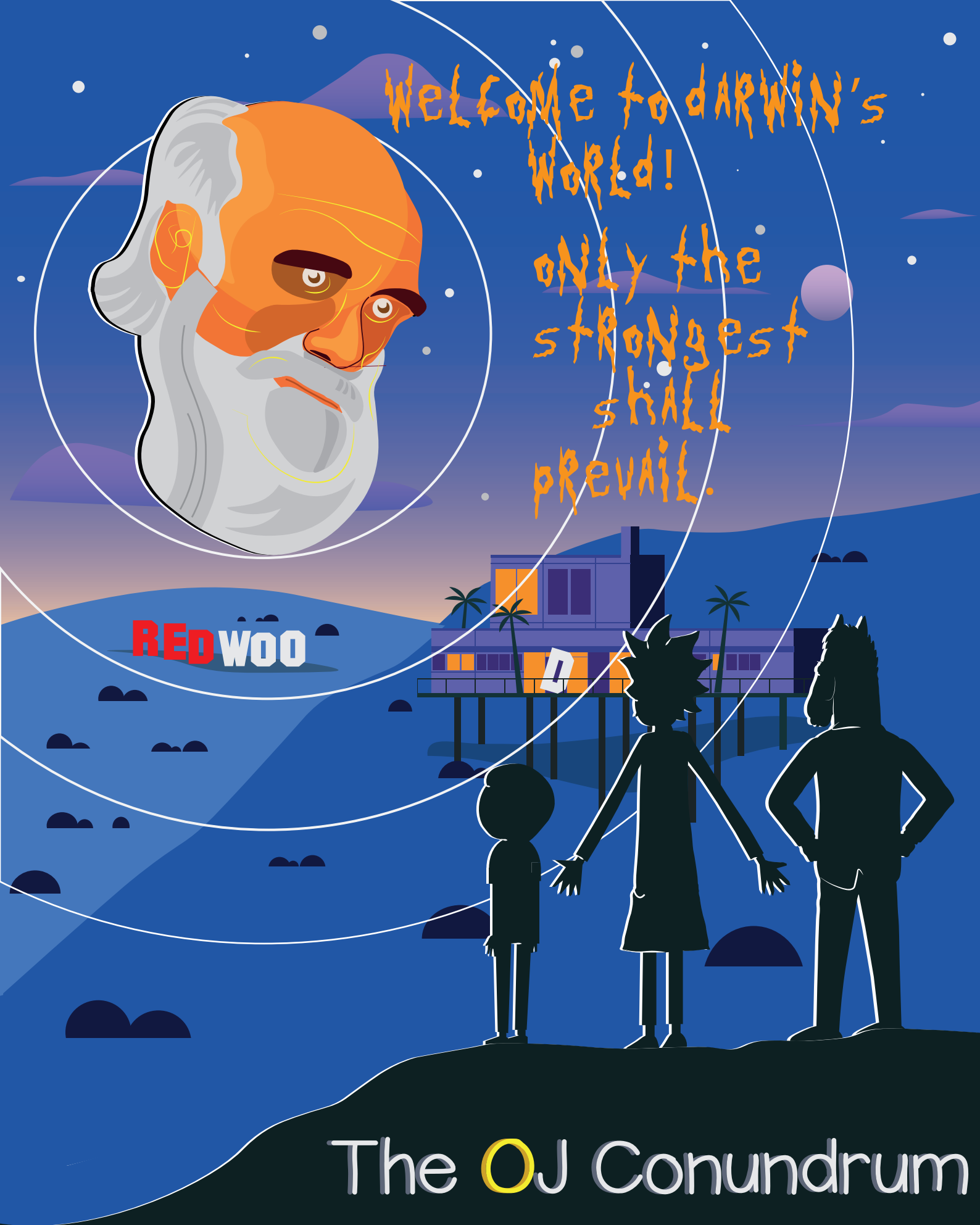


The OJ Conundrum

"Get GREENfty? What the hell is that?"
"It's our world's best effort, that's what"
PING!

Written by Shashwat Goel
Data Visualisations by Jyoti Sunkara

void Introduction ( )

{

Among the most uniquely implemented courses in IIIT-H's undergraduate curriculum are Computer Programming and Data Structures and Algorithms. While they are standard Computer Science offerings, they have a special place in our hearts. What distinguishes them is the unmatched focus on solving problems and the flavour of competitive programming (hence-forth referred to as CP[1]).

Before I begin, here goes a disclosure. I write this as someone who took admission from the Olympiad mode, or more colloquially, an 'IOI'[2]. I have been doing Competitive Programming off-and-on since Grade 9. It has been what introduced me to and developed my interest in CS.

No, I'm not going to fanboy over OJ. I won't brain-wash you into thinking your experience is totally your responsibility. Much rather, the opposite. The goal of this article is to compile long-standing issues with OJ, analyzing what causes them and proposing how they can be fixed.

}

the iiit c-pro/dsa course why did I study these?

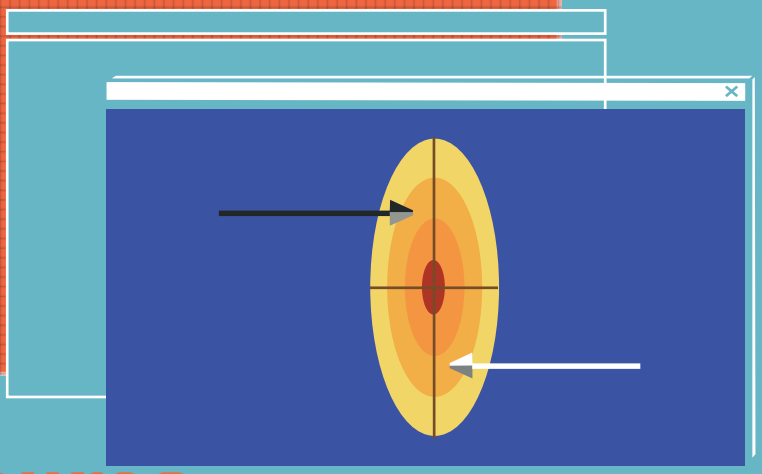
```
sort (reasons.begin(), reasons.end(), byPriorityDesc());
```

```
/*
    1. DSA questions have a significant (disproportionate, in India) weightage in Job Interviews, and will probably
       decide your package.
    2. Internships…
    3. ICPC…
        .
        .
        .
1e9+6. CF Rating...
1e9+7. The concepts taught are fundamental to, and the foundations of, all aspects of computing.

*/
```

[1] https://codeforces.com/blog/entry/67253 - Links without context.

[2] Please try not to call all Informatics Olympiad admits 'IOI's. Fun fact: only a few of us actually made it to the Indian Team that goes for IOI, and for some of us who didn't it only brings back sad memories :P



Course Goals

So IIIT chose to not beat around the bush. Most of the course-weightage is allocated to coding-contest-ish assignments and exams. While lectures are mostly theoretical, students spend the majority of their time on labs, assignments, and CP. In fact, IIIT keeps out the analysis part for a future course, Algorithms and Analysis in II-I.

The course goals include motivating the writing of modular code (OOP), introducing some nuances of C/C++ (IO buffers, directives etc.), good coding practices, improving problem-solving and computational-thinking abilities etc. The syllabus covered is enough to crack the DSA part of most job interviews. Students mostly end up needing additional prep at that time though (no, our cache can't hold this stuff for 3+ years).

My main focus will mainly be the assignments and exams, popularly combined into the single term - "OJ".

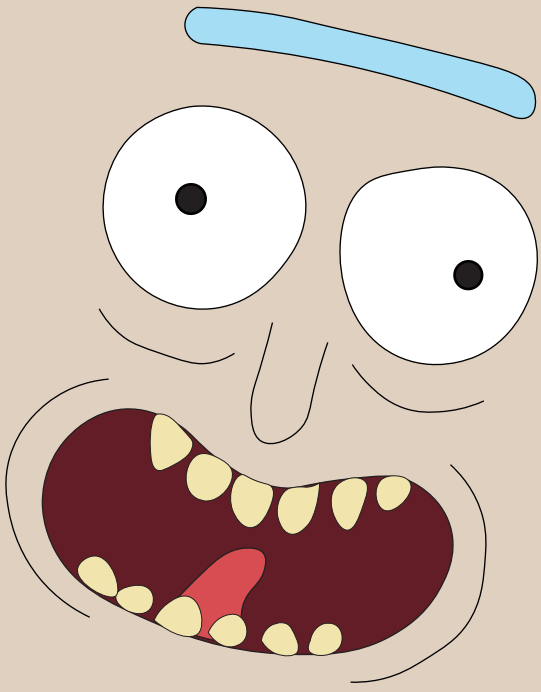
Assignments

THE ONLINE JUDGE PLATFORM

Imagine going through messy codes of 300 mostly-clueless students on a regular basis. Doesn't work right? An Online Judge automates this process. Outputs of submitted code are evaluated on pre-decided test inputs using a validation script that checks for correctness and run-time.

+ Advantages

1. Students can make multiple submissions for every question, with full feedback on correctness. This is unlike other assignments where you can't fix your mistakes.
2. It simulates coding contests, online platforms etc.
3. A hands-on approach. Instead of writing code on paper, student's get to go through the complete process of algorithm design, testing and debugging.
4. It saves a lot of time and effort for TAs.



Disadvantages

1. No feedback on coding practices, approach, possibilities of making the code easier etc.

2. Binary grading. A small bug could mean an absolute 0. This is somewhat mitigated by advantage 1 above. It can further be fixed with the introduction of subtasks. A more general issue, however, is an inherent lack of subjectivity in the grading. In most practical scenarios, codes and algorithms are often not “right”, or “wrong”. Rather, they lie on a spectrum based on efficiency, accuracy and readability.

3. No way to check if the student actually understands the code submitted and how much help was taken.

Frankly, the disadvantages are also shared by assignments in other courses to an extent. This article is indeed not about the use of an Online Judge. Instead, I shall focus on the content of the assignments, it’s scope, and how an OJ with problems is not enough to achieve the previously stated course goals.

Content

The OJ problems, both for assignments and exams, are largely picked by Teaching Assistants (TAs). There is little oversight from professors. Most of the TAs chosen every year actively take part in CP contests both online and ICPC, or have done so in the past. The importance of this dynamic cannot be overstated! It gives birth to the excessive influence of CP problems in assignments. The disconnect of people who regularly do CP and other students is highlighted throughout the survey graphs we present. It clearly shows how people who only do CP being TAs is a harmful phenomenon.

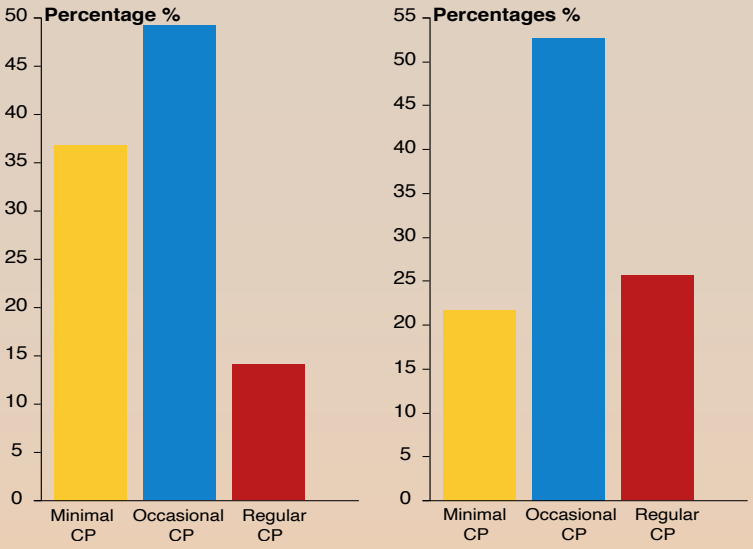


Fig. Types of people based on programming experience: *Results form 2 surveys have been used for this article. One was conducted after CPro and received 114 responses (Graph 1). The other was conducted after DSA and received 74 responses (Graph 2). In the respondents, a small increase in the amount of CP done is seen from CPro to DSA. The lower number of responses in DSA could be because after CPro, more students responded hoping survey feedback will be adopted for DSA. The results were made available informally to TAs and UG1 members of the Student Parliament, but no further action was noticed.*

Contrary to popular belief, CP forms a small part of IIITians intellectual interests, especially after the first year. Most students indulge in the sport only due to the prevalence of similar questions in the hiring process. Even hiring usually involves classic questions that can be

found on dedicated interviewing platforms like LeetCode. Most future coursework and research (unless related to algorithms) too only require a basic understanding of algorithmic concepts. This is unlike typical CP problems that come up in OJ assignments which often involve nuanced tricks and mathematical observations. Rarely is the depth and style of problem-solving in OJ of practical use to most students.

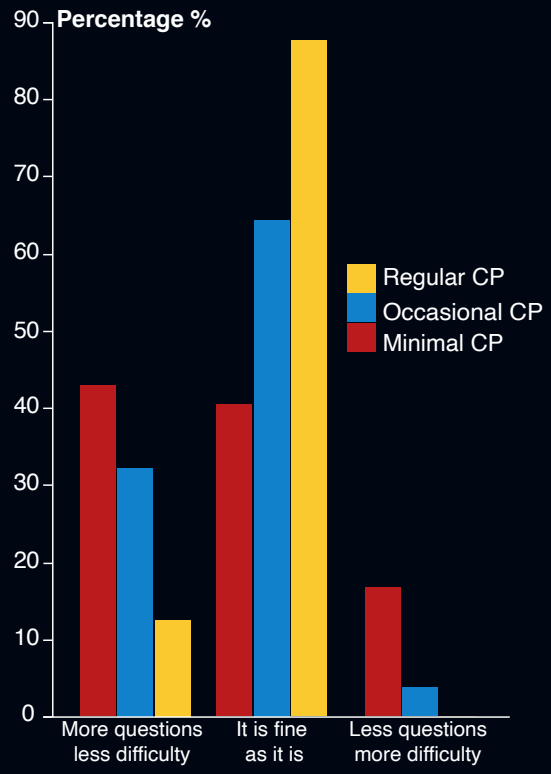


Fig. C-PRO

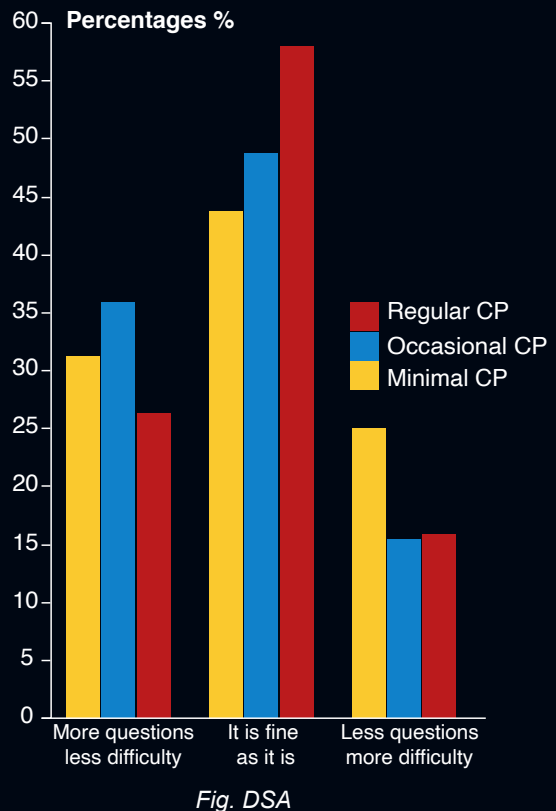


Fig. DSA

***Fig. Number of Questions and Difficulty**: In CPro (first graph), Most students who do CP actively think the questions/difficulty ratio is fine already. However, a significant portion of those with less experience think there should be more questions with less difficulty. Notice the change in DSA (second graph), with an increased number of students agreeing difficulty should be lower.*

Here's a simpler, more relatable way to understand the problem with CP being the focus of these courses:
Go back in time a few years, and imagine being taught for RMO (Regional Mathematics Olympiad) to ace JEE[3] Maths(my sincere apology to DASA students). In fact, imagine your teacher hardly focusing on JEE questions, instead, hoping you'll ace them if you practice for RMO anyway. I think you'd agree you'd be furious at the teacher for the rest of your life. You'd pass the blame for successfully pulling down your JEE rank, which led to you joining IIIT… I digress.

It is true that lectures mostly consist of theory and tutorials/labs help with the basics. Regardless, OJ dominates the focus, and what students take-away from the course. The practical reality is that theory taught is often just ignored by students. At least until the written exams, which carry lesser weightage anyway. Since UG1 students devote a majority of their "acads-time" to OJ, it is imperative to get it right!

DIFFICULTY:

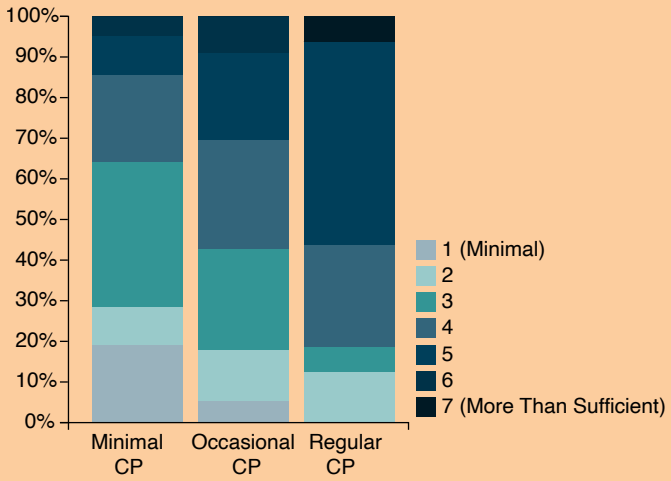
A BAPTISM BY FIRE

At this point, it might feel I'm arguing for a complete pause on drawing questions from CP platforms. Rather, as someone who loves it as a hobby, I think CP has great potential to be used as a tool for teaching. Some OJ problems do tend to be of great educational value. These clarify some very elegant details and applications of the algorithm in question. However, one must not forget (as TAs often do) that its role should be of a tool, and becoming good at CP is not the goal.

[3] Ofcourse, JEE too is party to the practice of forcing students to learn way beyond what they mostly need but the point is that's not the right way and this practice has to stop somewhere.

WHY HARD-CORE CP?!

My opposition is to problems that are hard applications of niche tricks (like re-rooting Tree DP) that are unlikely to surface outside the fantasy world of CP. These are neither taught to students before-hand nor easy to come up with independently, even for seasoned competitive programmers.



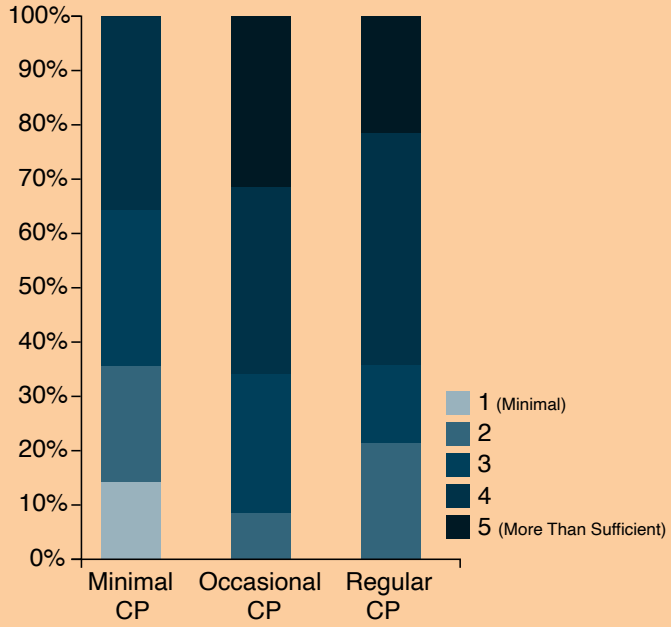


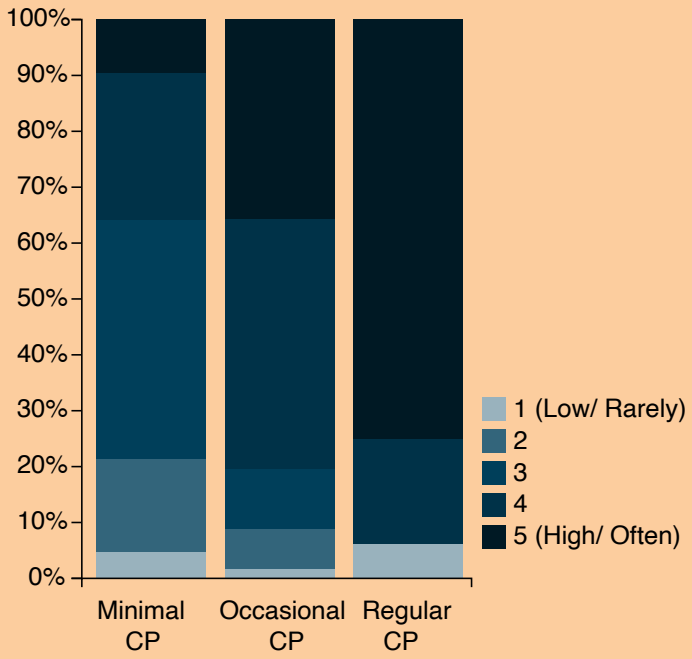
Fig. To what extent are prerequisites for problems covered in Lab/Lectures? : *In the CPro survey (first graph), the 'average' mark was 4/7, and in DSA 3/5 (second graph). The percentage of people who found prerequisite coverage less than sufficient, halves from 80+% to 40% in CPro and 60+% to 30% in DSA. Notably, active competitive programmers (future TAs) mostly don't take labs and tutorials seriously, and since they know the concepts, don't realize how prerequisites are often not covered.*

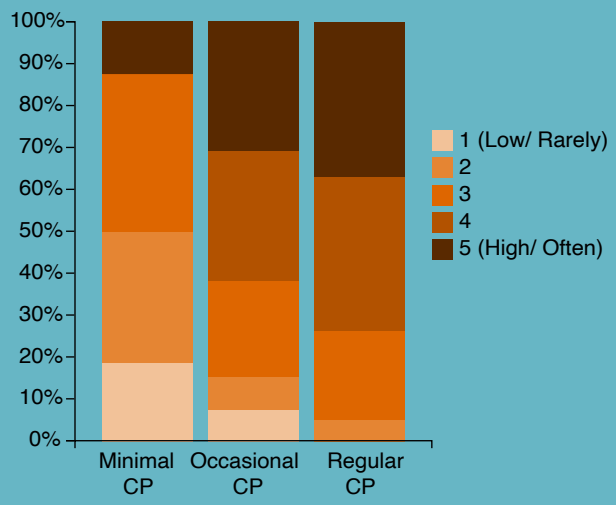
It is good that problems are not directly from what is taught, but in such cases, it is also important to ask: *"Will everyone really ever need such thinking in their life?!"* Some Examples:

1. A problem that required the IOI 2016 Aliens DP Trick. An in-depth understanding of the solution requires knowledge of Lagrange Multipliers, a concept in Multivariable Calculus. At IOI 2016, only 1 participant solved the problem for a full 100 points. 40+ participants got stuck at 60 points, which could be achieved without the use of this trick. For those thinking "well, that's a high-school contest", top IOI participants tend to be Orange/Red on Codeforces, a feat held usually by less than 10 IIIT-H students, none during UG1.

2. Multiple OJ problems can be traced back to Div1D and harder problems on Codeforces and or major international contests (IOI, ICPC). The irony is that even at the end of the course, most students are not comfortable with solving Div2D problems.

3. This is especially prominent in DSA. The first code one writes (unless they do CP) on a non-trivial topic like Tries can require significant modification from the vanilla data structure taught in lectures. Not to mention, a tricky observation-based solution precedes the coding.





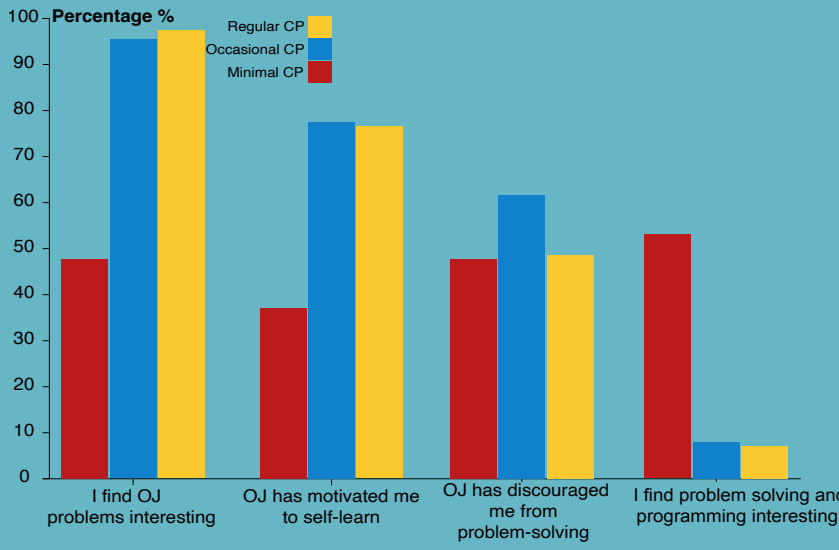


Fig. Given the logic, what percentage of questions students implement independently: *If students don't do CP, they find it hard to even implement more than half the problems even after being told the logic. The numbers naturally get worse across categories in DSA (nearly 40% who even do CP occasionally can't implement half the problems).*

A lot of ex-TAs, alumni etc. (as seen on Confessions@ IIIT) think that questions being this tough promotes learning. The explanation is that if one would really work hard, they would probably do well anyway.

Justification for OJ difficulty is often made citing "Army Training". Once again: The army *needs* that training to survive, but IIIT students have little use for advanced CP tricks! Perhaps IIIT's timely response to the COVID crisis comes due to its experience with exponential learning curves...

Basically, the argument isn't against difficult questions. Students are often welcome to challenging tasks provided completing them is beneficial. Their efforts should instead be conserved for and channelized towards academics that are more applicable to research, jobs, or more broadly, the real-world.

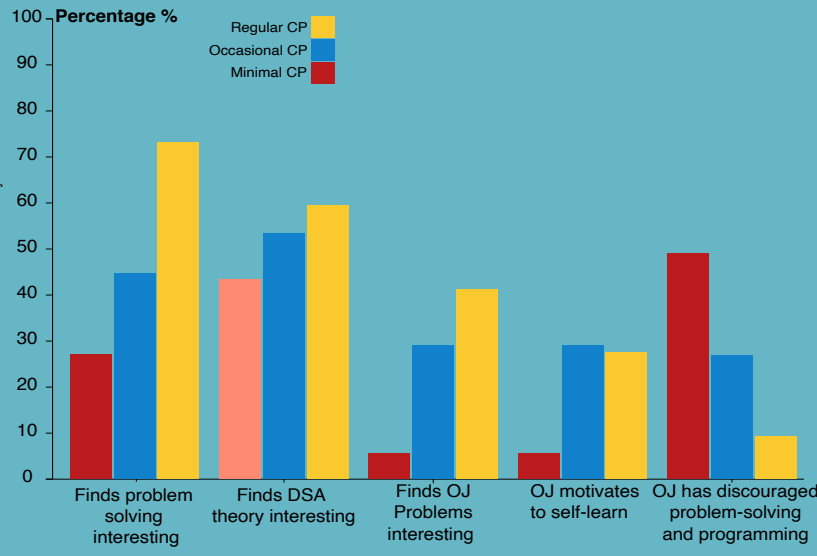


Fig. CP Practiced on other platforms: *There is a lot to unpack here, and these graph say the most about the adverse outcomes of OJ. The first graph is after CPro, while the second is after DSA. There is a clear drop in the percentage of people who find problem solving and programming interesting, mainly due to DSA notably among those who do CP occasionally, it fell from 85% to 50%. This is not seen among those who do CP regularly. In DSA, Clearly the majority doesn't find OJ problems interesting, and neither does OJ motivate them to self-learn. On both these metrics, the feedback after CPro was still decent. More than half of even those who don't do CP regularly do find the theory interesting, in-fact, almost as much as those who do CP regularly. However, a significantly high number of students (After DSA, 60+% of those who do minimal CP, and 30% who do it occasionally) say OJ has demotivated them from problem-solving and algorithms! This is really sad considering these students are committed to a career in CS in the near future.*

It is natural to lose interest in something given little time to fully understand and appreciate concepts. The difficulty curve discourages one not just from algorithms, but being first-year students, from CS as a whole. It may not be rational, but students do tend to feel scared, stressed and disheartened.

int alternatePerspectives()

Moreover, it is important to highlight that the course is taken not just by CS students, but also ECE and ECD. The irrelevance to them grows as questions get more specific to CP. On similar lines, an interesting perspective is that of Lateral Entry students. They don't have to go through the OJ grind[4], yet they don't face any disadvantages in coursework and hiring. This indicates much of the scope of OJ is unnecessary.

[4] Programming is covered as part of a course during the summer to bring them up to speed.

The difficulty level often leaves students with little options. One's first demossing experience soon sows the seed for rampant plagiarism for the rest of their courses at IIIT.

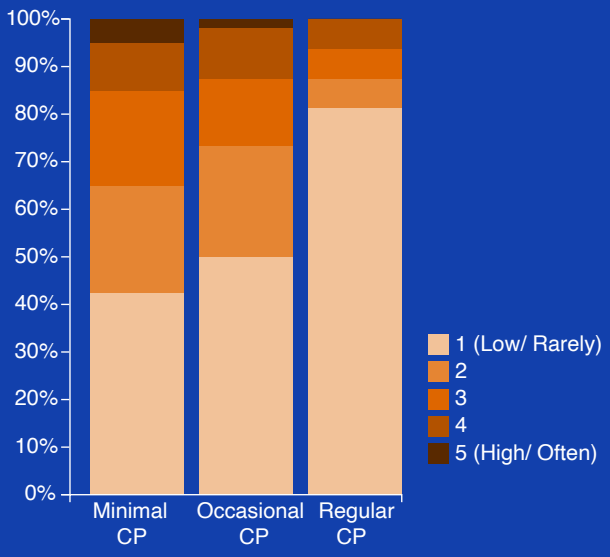
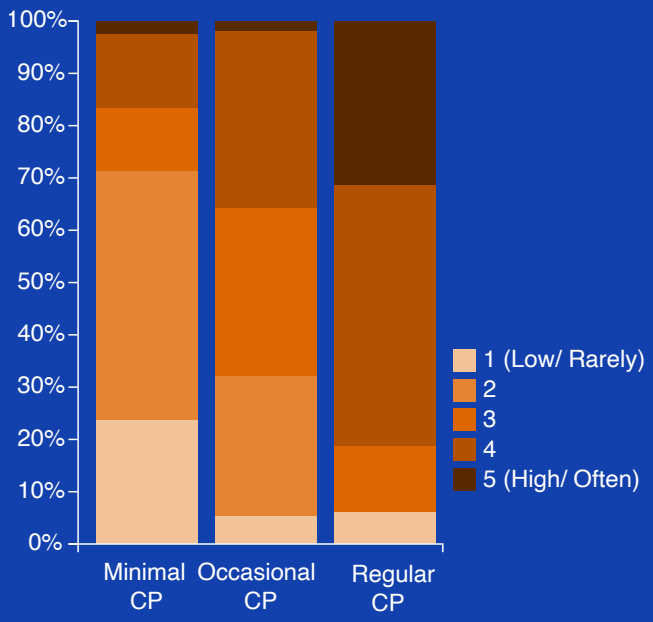


Fig. How often have students been tempted to plagiarize intelligently? : *More than half the students who don't take part in CP contests actively feel pushed towards plagiarizing. Note that this survey was collected after CPro before grades came out, and respondent E-mail IDs were collected, which probably means students were not open enough despite our assurance that the data will be de-identified. The numbers are definitely much worse.*

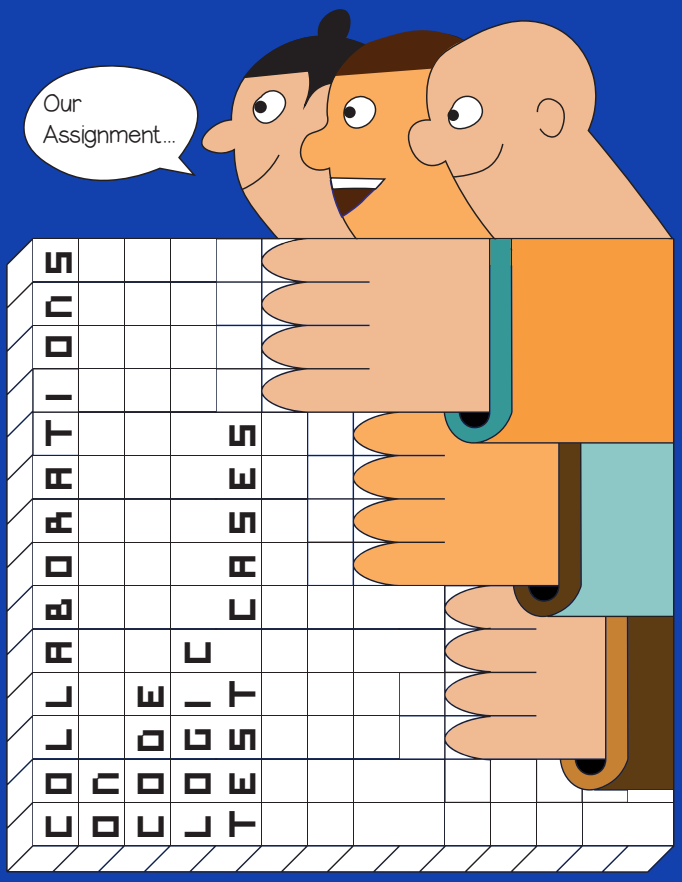
plagiarize. Lab exams end up providing an essential eye-opener to course administrators and students alike. What's shocking is that students have been caught using unfair means even in Lab-exams this semester!



THE ACADEMIC HONOR CODE

How excessive collaboration leads to harder problems

It goes without saying that the student body is equally responsible for some problems. Professor oversight in assignments is often limited to tracking the final leaderboard. Professors seem to be aware of the problems with plagiarism, but perhaps not the extent. A disproportionately large number of accepted submissions are made on the last day. Many of these are demossed, or even coded by someone else. Most of these cases cannot be caught, as the art of beating MOSS is one that many IIIT students master to perfection. Seeing high scores, TAs keep making the problems tougher and professors don't realize what's happening. This pushes an even larger number of students to have no choice except to



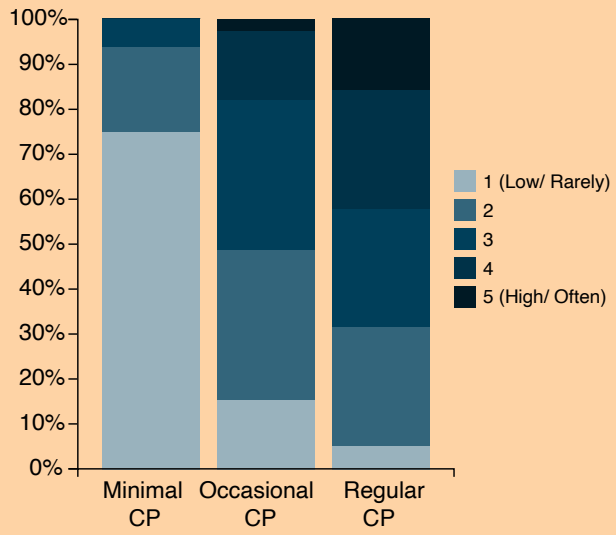


Fig. How well is a student able to solve without any hints?: *Most students end up needing hints to solve problems even in CPro (Graph 1).The problem compounds significantly in DSA (Graph 2) with as many as 75% students who are not interested in CP saying they can rarely solve the problems without hints. Among those who do CP occasionally, almost 50% require hints for most problems. Even amongst active CPers, less than 40% can solve half the problems without hints. Given that these 'hints' are peer-based, it's not easy to quantify whether they are just observations related to the problem, entire solutions or even coding/ debugging help.*

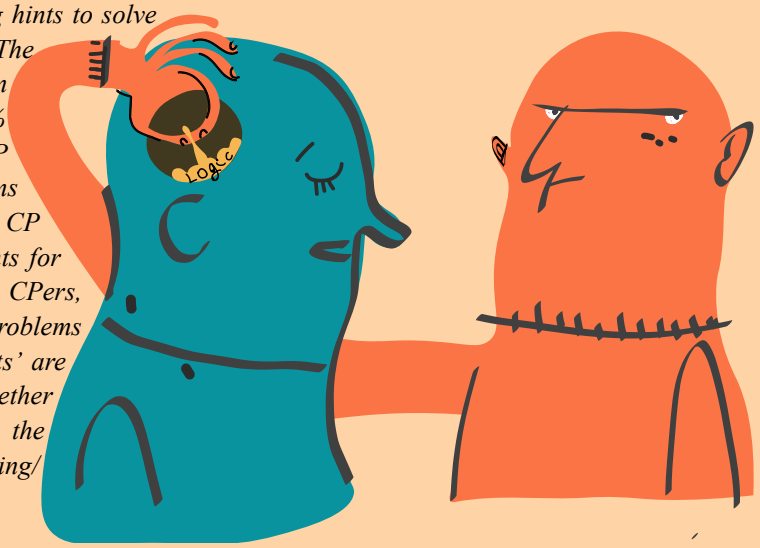
It is not like IIIT inherits students who always plagiarized. In-fact, it only selects students who have done well in high-school. Back in JEE coaching, such students looked down upon plagiarism themselves. They had an acute realization of how it's futile in the long run. People actually fought for extra assignment sheets for practice. So why does IIIT (and in general Indian colleges) have an excessive collaboration problem?

The main reasons I believe are a lack of motivation, passion and trust in the curriculum. CPro and DSA are a little different in this issue than most other IIIT courses though. Students are fully aware that the skills will help them later in interviews. They want to be good at this. They trust their TAs to get things right, considering active CPers in the community often do well in getting offers. So what goes wrong?

Universities across the world tackle plagiarism by instilling an almost religious belief in the Academic Honor Code amongst students, an idea alien to most at IIIT. Specifically in CPro/DSA, the short-term incentives to plagiarize are high with low penalties if caught. Many students get a Demossing 101 from seniors (in exchange for DLF trips, Felicity work and the likes) or through public resources as soon as they enter IIIT.

WHY DEMOSS?

A typical OJ question can take anywhere between 1-4 hours of dedicated effort. This is assuming you don't get stuck at some point (which you do, multiple times). Given that being at IIIT you probably have another deadline the same week, it's attractive to cut down significant parts of the above process through peer assistance. When left with too little time for this on the last day, demossing is the final fallback.



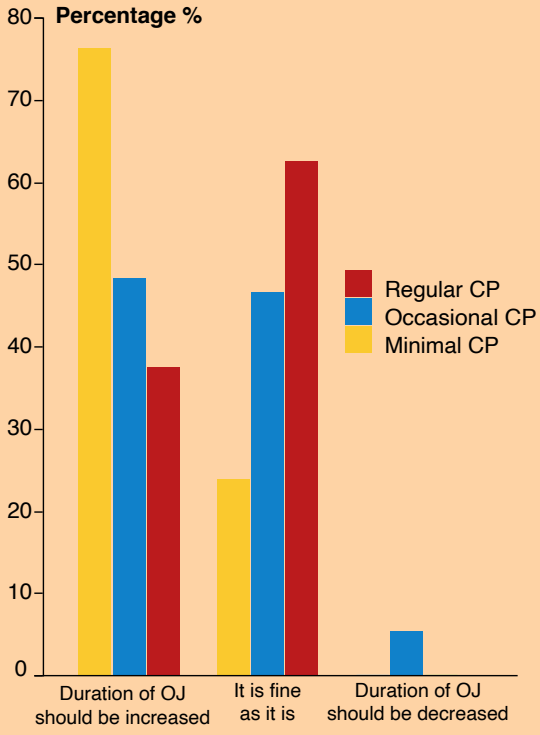


Fig. Duration of Online Judge:*A majority of the students, especially those who do less CP believe that the duration of OJ should be increased.*

It's also hard to catch up once you've been left behind. Due to the overlap in concepts, many questions become unsolvable if you haven't understood the previous assignment's solutions thoroughly.

The leaderboard further fuels anxiety as leaving a few questions puts you below the median. You are tempted to think: what's the worst that can happen if you deMOSS? It's hard to be careless enough to somehow fail a demoss attempt. Even then, TA generosity will probably let you off with points only lost in that question alone. Points you wouldn't get if you didn't demoss anyway. No escalations, no record beyond memories to reminiscence on, no institutional action whatsoever!

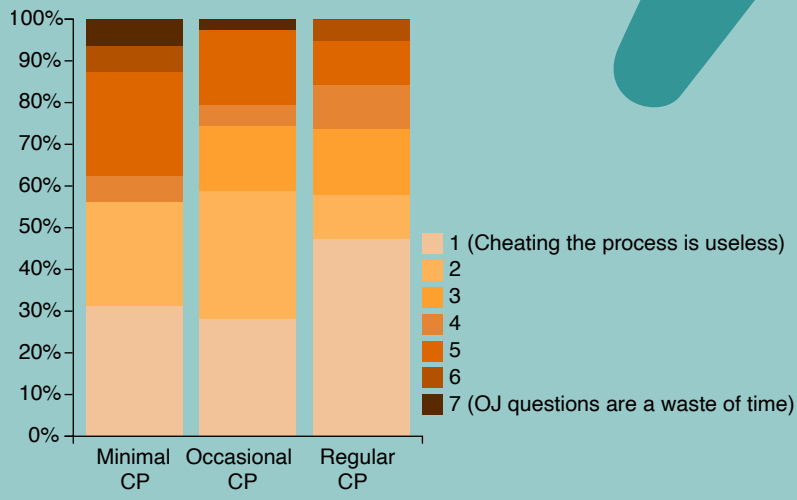


Fig. To what extend do students think that demossing OJ saves actual time for learning: *Ideally, each bar in this graph should be completely dark purple. Across categories, over 40% of students believe that demossing does save time to some extent (3/7 or more). Again the disparity between students who already do CP actively and those who do it lesser is evident.*

Letting them know would've prevented stress and allowed them to plan which questions to leave. The justification given was 'But then you wouldn't try the hard questions', to which my simple retort is that it's not like most did it themselves anyway. Forcing students to plagiarise is unfortunately seen as a fair trade-off for motivating them to attempt a very hard problem.

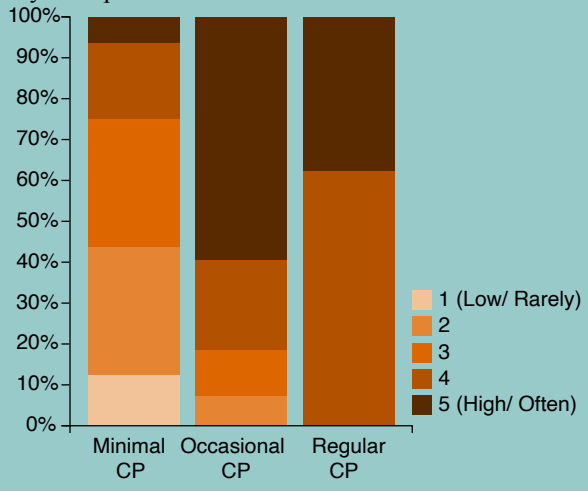


Fig. Percentage of OJ Questions students are positive of solving from start to finish without any assistance today: *This graph indicates that people aren't really confident about their ability to solve all of the OJ problems (including both CPro, DSA), after the end of both courses. The much higher confidence of the middle-category, people who solve problems only occasionally seems unintuitive. Or, could it be the Dunning Kruger effect?*

NEED: BEING ROBUST TO PLAGIARISM

Keeping all this in mind, to their credit, the TAs did enforce a best-of-30 (out of 35) policy for DSA this semester. Essentially, solving more than 30 questions would mean no extra weightage. Alas, students were not informed of this till before the very last OJ. By then most had already dabbled in the dark arts to stay afloat.

Of course, if all students could stick to their abilities/ interests and leave questions, relative grading would ensure they get the grade they deserve. This is easier said than done though. It must be realized that this forms an "unstable equilibrium." If a small group of students over-step this line, it feels unfair to the rest how their honesty will be rewarded with lower grades. I agree this is not a rational ideology, and the focus really needs to be on learning. However, it is important to stop hoping students are saints and analyze where they are coming from.

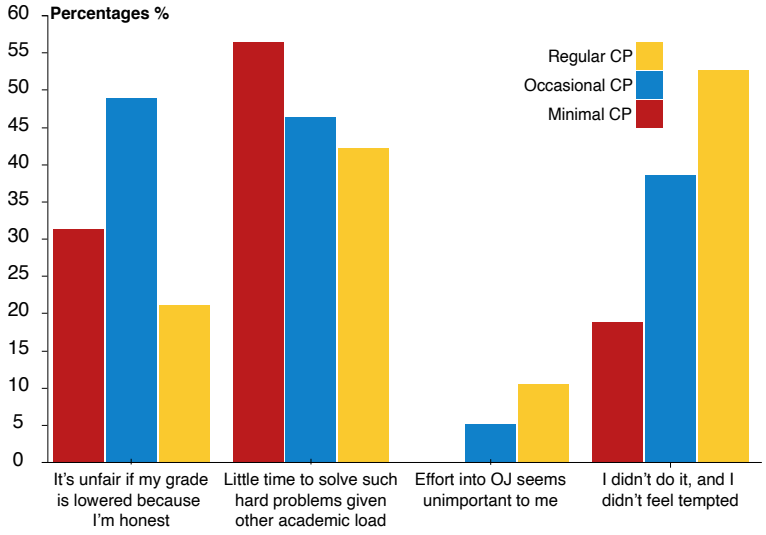


Fig. Why are people motivated towards excessive collaboration/demossing: *This survey was taken after DSA. It is interesting to see that almost 80% of those who do minimal CP, 60% of those who do it occasionally, and 45% of even those who do it regularly refrained from saying they haven't partaken in excessive collaboration. This survey was anonymous, and hence it seems to better represent the actual picture than the CPro one (also, DSA was much harder). The reasons agree with our observations too. Academic load (almost 50% agree) and seeing others do it (~35%) are the main factors that lead to it. Very few people (and among them, mainly who already do CP regularly) feel that putting effort into OJ seems unimportant, which I think would be a more popular option to justify plagiarism in other courses.*

Leaderboard

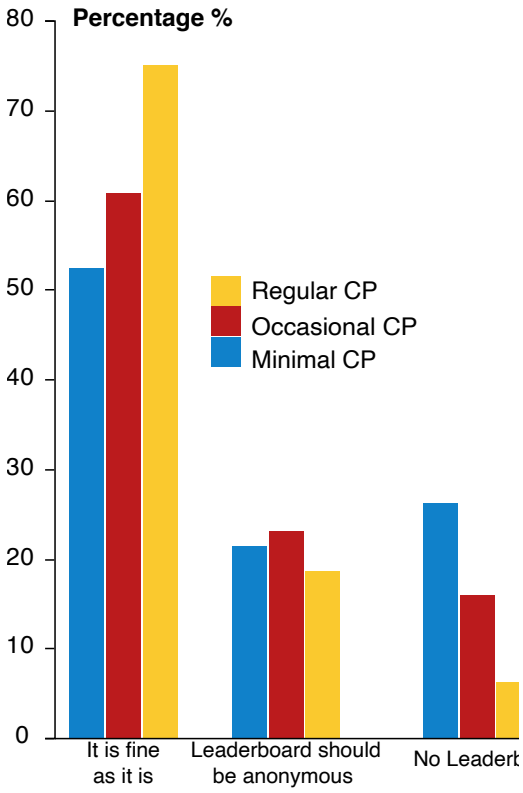
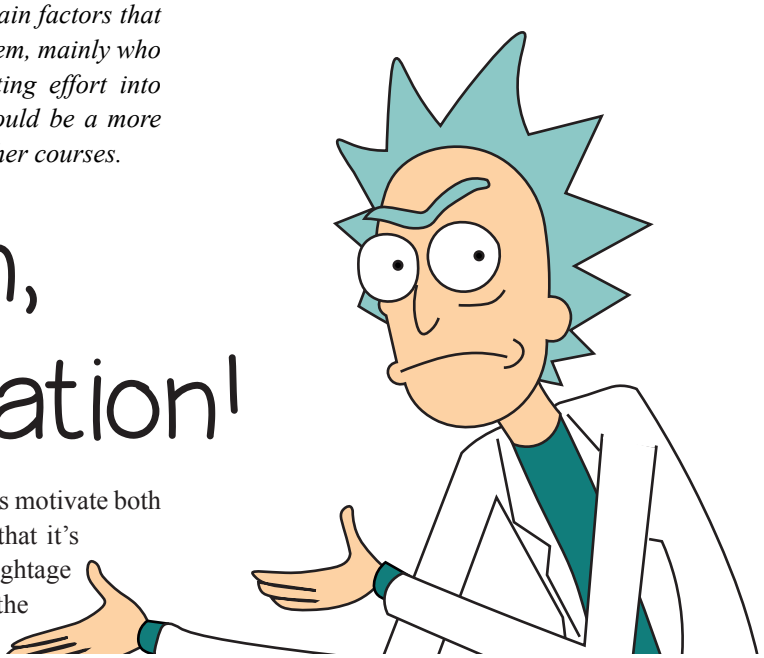


Fig. Review of the public leaderboard with scores: *Al half the students who do minimal CP, and 40% of who do it moderately agree that the leaderboard sh either be anonymous or that it shouldn't exist. Those actively do CP and thus are higher on the leaderb ofcourse crave the satisfaction it provides!*

Education, not evaluation!



One of the fundamental issues that perhaps motivate both the steep curve and plagiarism in OJ is that it's an evaluation metric with significant weightage (20-30%). This often clashes with the learning aspect of the course.

Throughout OJ, a live leaderboard is available to students. The only way it benefits the students in my opinion is that they now know who to ask for solutions. The leaderboard gives birth to unhealthy competition, adding yet another meaningless number that students can use to define one's self-worth.

While it is irrational to hold yourself up to students with 2+ years of prior experience, it is not uncommon for students to feel further discouraged by seeing Informatics olympiad admits (or those who prepared for it) solve questions with ease. Moreover, students who do CP actively interact more frequently with TAs outside academic settings. I myself have cribbed about finding OJ boring in casual conversation with TAs. This leads to a nudge towards making problems interesting for us but unreasonably hard for others. People with in-depth prior knowledge of the course end up causing problems to others unintentionally. A possible fix could be extending policies like RSAO or offering an elective to such students. This would also help mitigate the unhealthy status bestowed upon olympiad admits in UG1.

TA and Jerry

In UG1 CP-circles, multiple conversations arise around OJ and students often express their wish to be TAs in the future. This is accompanied with remarks on how "the opportunity to make life tough for juniors" seems fun. Sadly, these tendencies are harboured deeper than just humour. An indicator of this has been certain E-mails from TAs sent out to UG2k19 during DSA. For upper-classmen who haven't witnessed these gems, I attach them here for your enlightenment. The E-mails are not an isolated incident. After performing poorly in the CPro mid-sem lab exam, students were reprimanded for 6+ hours in labs. Note that this was for many students their first experience with a timed contest.

While the patience TAs show in teaching and doubt-clearing is impressive on many occasions, such actions only reflect hubris. They reinforce the image of assignments and grades being a cat and mouse game, though I do agree it's students who overstep the line first.

Misplaced priorities are demonstrated when little is done to convey the learning aspect of OJ. Some TAs are open to discussing doubts on a one-to-one basis or in tutorials. But, once an OJ is over, no feedback is provided to students formally. Even solutions to the problems aren't released on Moodle/Mail. Fed up, this semester UG2k19 students themselves initiated a forum for post-assignment resources, though it understandably couldn't continue for long. Indeed, it is assumed that once the deadline is past and the evaluation needs are met, OJ has served its purpose. This ideology naturally pervades down to students. Students start seeing OJ as plainly an opportunity to score marks, by hook or by crook.

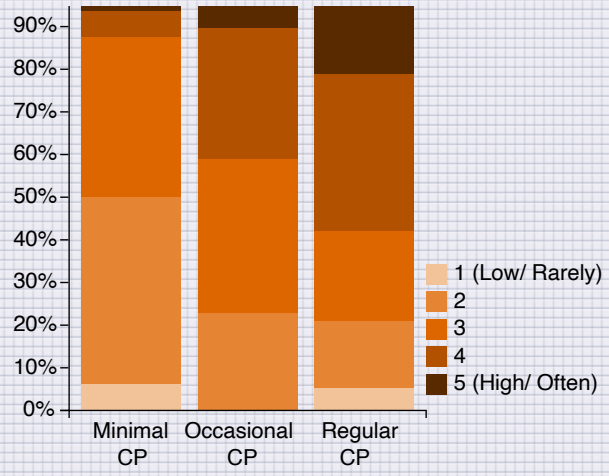
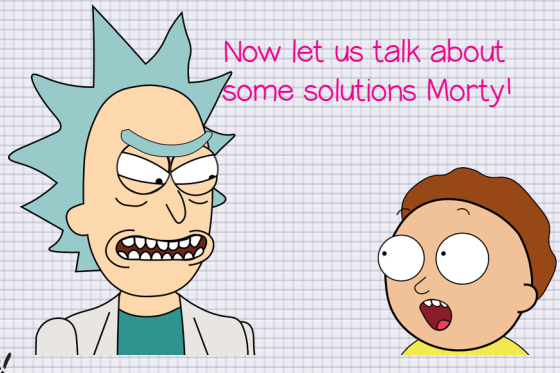


Fig. To what extend did OJ help give students conceptual clarity about the theory taught in lectures: *Almost 90% of students who do minimal CP, and 60% of those who do it occasionally believe that the conceptual clarity conveyed by OJ questions is average or below. The number is as high as 40% even amongst students who actively do CP and thus have a strong grasp on the concepts anyway.*

It is even possible, I believe, for TAs to go through codes of harder questions to provide concise feedback on coding practices and methodology to students. Just like 'Evals' held for assignments in other courses, a student can present his codes to TAs in 5-10 minutes. One might argue that TAs don't have time to provide feedback, but then they do spend entire days looking into plagiarism cases.

All this said, there's one thing OJ does teach us - finding bugs is pointless if you can't fix them.



AUTHOR'S TAKE - SOME POSSIBLE SOLUTIONS

Here are some things that I feel should be implemented, or at least tried.

1. Releasing official editorials with hints, solutions, well-written codes and proofs should be a bare minimum requirement. Additionally, the problems should be kept open for submissions later.

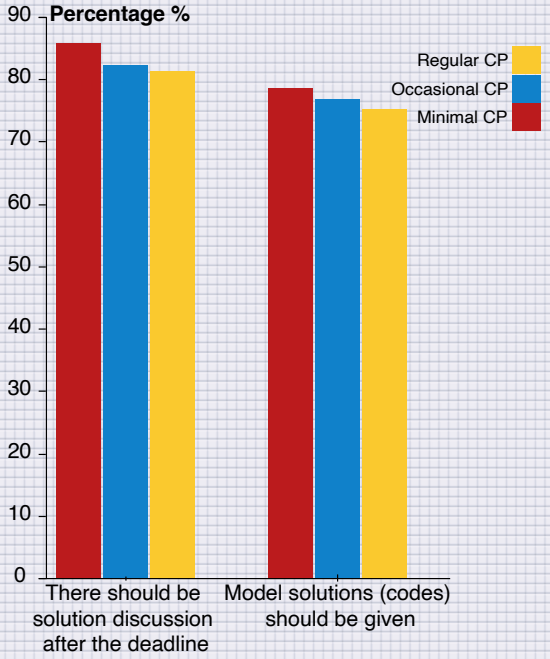


Fig. Opinion on what should be done after an Online Judge assessment: *Almost all students agree there is a need for official model solutions and discussions.*

2. The emphasis of OJ should be learning, not testing. Perhaps the weightage of OJ should be reduced, in favour of holding more lab assessments. Increasing assessments is unpopular with students (ref: the quiz system).

However, it is probably better for CPro/DSA, where back-log cannot be recovered in one night of study. Problem solving and coding can only be improved incrementally with practice. With OJ questions of high enough quality, students will still be incentivized to do them seriously anyway.

This also splits the weightage across assessments. This is necessary as it is natural to panic and mess up a lab-exam, especially for students who don't take part in online contests regularly (which again, should not be a necessity).

3. Not relying solely on OJ. Problem-solving is a key element, but not the only course goal. Students can be given code handouts to improve upon, complete, modify and debug as exercises. These can introduce much longer, harder codes than what the scope of OJ allows. It will also continuously expose them to good coding practices, and they will naturally adopt them. With a similar aim, students should also be introduced to IDE's with static analyzers (that give constant feedback on coding practices) and not be forced to use Vim. This will also help students learn specific caveats of the language. In general, inspiration can be drawn from diverse teaching practices seen across other courses in IIIT. One-OJ-fits-all is an unrealistic expectation.

4. There should be some group projects. There have been talks of getting different students to write codes integrated into a single data structure program. This adds a collaborative aspect while motivating the need for coding practices first-hand. "Challenge problems", which require developing heuristics that maximize an objective function should also be used. These often come up in the real-world. Their non-binary nature incentivizes students to try different approaches and optimizations. Moreover, students cannot use the common 'there are only so many ways you can solve this problem' trope when caught for plagiarism. There is a lot of room for creativity!

5. At least some problems should have some real-world connection students can appreciate afterwards. Meaningful insights from the problem should be communicated too. This will make setters think deeply about what is actually useful for students. Intriguing CP problems are not necessarily ones every CS (even less so Electronics) student gains something from. It might be better to lean towards common (even if less 'beautiful') problems that one is more likely to encounter later in some form.
6. Strictly cracking down on plagiarism from CPro OJ-1 itself. A bad example is set initially

when students are let-off, which makes plagiarism seem a viable shortcut later. Sure, fear is not an ideal solution. But self-awareness is a larger and different issue altogether.

Moreover, insights such as Temporal MOSS can be incorporated. The idea is to make students commit their progress regularly to detect unrealistic jumps in progress. Students putting in effort throughout the assignment period are less likely to feel the need to plagiarize too. Other creative options could include point-decays (as seen at Codeforces) for last day submissions.

7. Efforts should be made towards formalizing the hint process. This could be through increased subtasks as in Olympiad problems. Other options could include a system that gives automated hints or the wrong test-case after 'x' number of wrong submissions. This could be supplemented with a submission delay (slow-mode) of a few minutes to prevent exploitation if necessary. IIIT students are more than capable of making such a separate portal for hints linked to the OJ leaderboard for tracking.

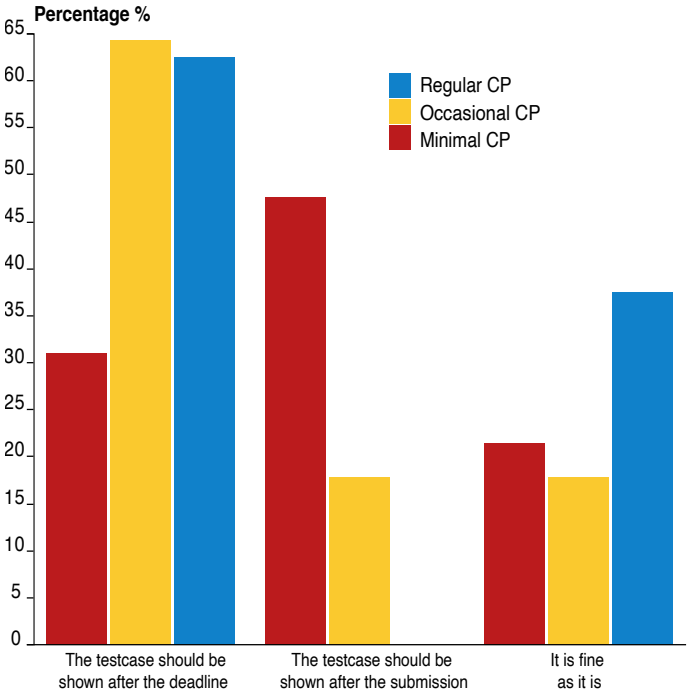


Fig. The hiding of testcases done to improve debugging skills

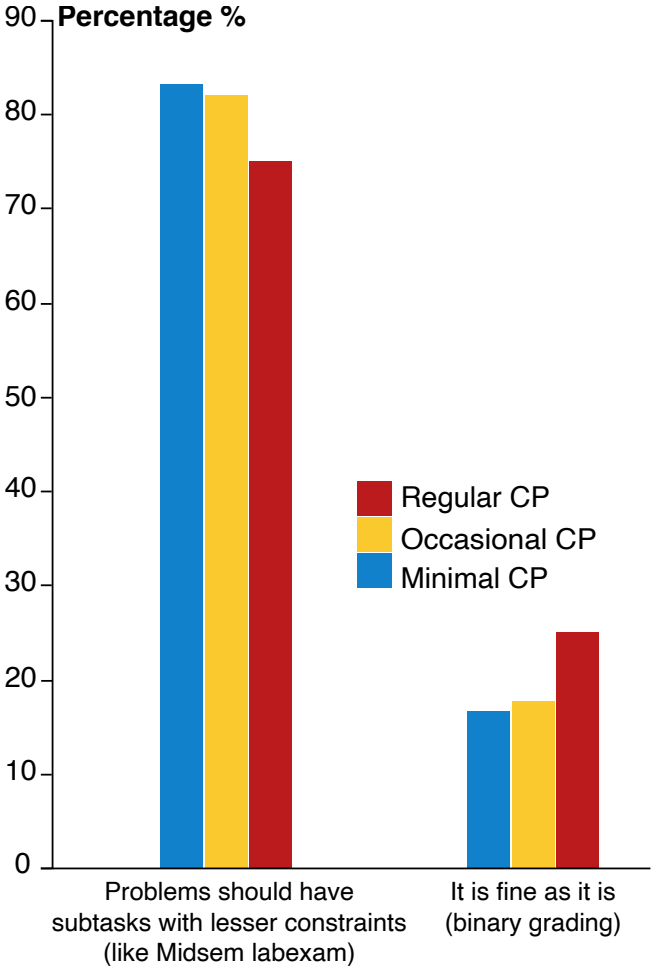


Fig. For or against the current binary system

8. A more diverse (in terms of intellectual interests) set of teaching assistants should be selected. It is true that competitive programmers tend to be more experienced in concepts taught in CPro/DSA. But there are compelling arguments as to why not all TAs should be the most skilled. This helps better optimize the outcomes of the general student body, instead of those who are already inclined towards CP.

   Based on my sources, this is easier said than done as applicants for TAship are mainly students interested in CP. However, it is worth questioning if this is so because of the status quo. If an effort is made to reallocate the emphasis on CP, it is possible more students would volunteer. There should obviously be more professor oversight to ensure course goals are met.

9. The problem statements should be less confusing, fantastical stories and more readable. The problems should be tested rigorously. It is frustrating to waste a day on debugging only to find out the test-cases themselves are wrong.

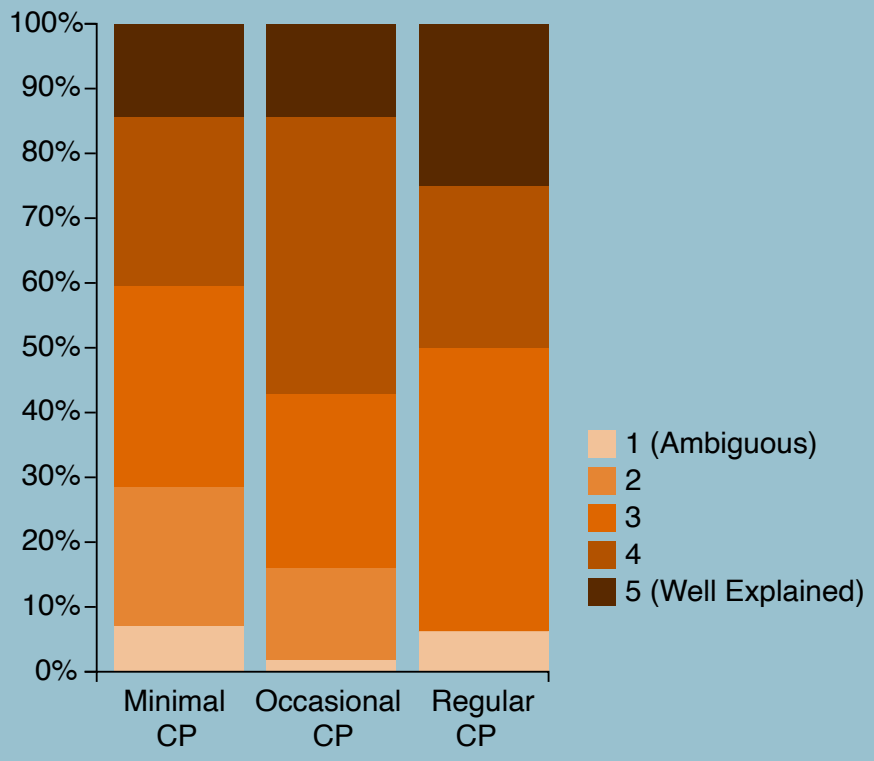


Fig. Clarity of Problem Statements: Around 50% students overall agree that the clarity of problems statements is average (3/5) or below.

10. I realize that the stated reforms require time and effort to enact. However, based on my discussions with past TAs, there are many redundancies in course administration. A lot of re-inventing the wheel happens every year, with practical problems being realized and fixed in the middle of the course. These insights are rarely documented for ready availability to future TAs. Not having to worry about problems tackled before would allow TAs more time to experiment with and incorporate changes.

CONCLUSION

I personally find using CP as a tool to introduce CS a refreshing avenue worth exploring. I am sure IIIT students are all-in to support novel pedagogical ideas like these. In-fact, I feel IIIT's unique programming offerings can be made more widely available. With some refinement, our CPro and DSA course designs hold the potential to be excellent alternate computational problem-solving courses for students (perhaps in high school) to try independently (think OCW). However, the remaining gaps need to first be carefully considered and bridged.

Moreover, I realize this article ends up painting TAs in a bad light. I would like to emphasize again the pains they go through to clear the dumbest of our doubts, handhold us through our inability to follow the clearest of instructions, and haggle with us over issues that clearly start on our side. It is immensely difficult to think about complex, subjective issues like pedagogy during hectic semesters. But with the world on the cusp of a revolution in education, perhaps this is the perfect opportunity to ruminate about the efficacy of learning at IIIT. UG2k20 will join us after a harrowing experience with delayed admissions, hopefully OJ won't make their transition to college as frustrating as ours.

I would like to thank Anurudh Peduri (past DSA TA, ACM-ICPC World Finalist), Athreya C (Lateral Entry student, past TA), Suryansh Srivastava (Student Placement Council, past TA) and more broadly the Theory Reading Group #spam channel (contributions by students who've been TAs, UG2k19 etc.) Discussions with them have offered invaluable insights and shaped my opinions in a more exhaustive and concrete way. ■