

50% rule should be changed.

Bogdan Yakovenko\*  
Warsaw University

## 1 Introduction

Last two years we have had a new rule implemented at IOI competition. Unfortunately, it brought lots of problems with it, and participants, as well as other IOI-related people, weren't happy with this rule. This document attempts to analyze the 50% rule, and to introduce a new one that could replace it.

## 2 Principles

50% rule claims that *correct-but-inefficient* program should obtain at least 50% of the full score. In addition, it is stated that the other 50% of the score is designed to recognize the efficientness of participants' solutions, so that a program that performs better than an inefficient solution and worse than an efficient one should get more than 50% of the score.

The addition of this rule was motivated by the fact that the credit for correct-but-inefficient programs is too small and therefore average contestants get low scores.

## 3 Problems

During last two IOIs I've heard many participants that they were not satisfied with the new rule and thought that it's not fair for them. It's interesting to know why the rule that adds more points for average participants is bad.

---

\*special thanks to Dumitru Ciubatii for corrections and advices

After speaking with participants and making some analysis I've found the problems that caused this:

1. First of all, writing an *inefficient* solution is much simpler than writing an *efficient* one. 'Simpler' means that just a little *time* is needed to *find* and *code* the solution. However, the *time needed for finding* an efficient solution is unpredictable, considerably greater, and depends much on the participant, and the *time needed for coding* it is at least THREE times greater than the one needed for an inefficient solution. Therefore, overall time spent for creating an inefficient solution is more than TWO times less than the time spent for an efficient one.

2. Participants who write *inefficient* solutions have time to think about optimizations and to successively implement them. That's why *correct-but-inefficient* solutions get more than 50% of score.

Essentially, 50% rule was designed to grade *inefficient* solutions that come to mind after understanding problem's model, and therefore getting 50% of score doesn't require any (even obvious) optimizations. So those participants, who add simple and obvious optimizations to their *inefficient* solutions, can get 60-70% of score. If clever optimizations are added, participants can get 80-90% or even 100% of the score.

Therefore, it's clear that an average participant who codes *correct-but-inefficient* solution gets more than 50% of points, which is more than it was designed.

3. Considering the time and skills required to implement efficient and inefficient solutions, the *unfair score distribution* becomes clear - the difference in points is too small.
4. Coding an inefficient solution is faster and simpler than coding an efficient one, thus *bugs probability decrease*. It's clear that having 3 *inefficient* solutions is always better and safer than having only 1 *optimal*. Sometimes 3 *inefficient* solutions ( $70+70+70 = 210$  pts.) can be better than 2 *optimal* ( $100+100 = 200$  pts.). In conclusion, 50% rule encourages participants to code *inefficient* + optimized solutions.
5. Finally, let's consider the problem about confusion. The rule can confuse participants because of unclear test design. If they write *inefficient* solutions, it's only clear that if the solution is correct then their score

will be  $\geq 50\%$ , but there are no possibilities for participant to determine the score more precisely. It means that if two Scientific Committee's members independently make two test case systems, the scores of participants' programs can be significantly different.

## 4 New rule proposal

My proposal is to modify the 50% rule, making the score distribution more fair and more predictable for the participants. The following statements represent some examples of such rule. Of course, it should be developed and discussed and only then implemented to the future IOIs.

1. First, the score percentage for *inefficient* solutions should be reduced from 50% to 30-35%. In addition, test cases should be designed with the consideration that *inefficient* solution could implement all obvious optimizations.
2. Second, to make participants' scores more predictable and to endorse participants to write efficient solutions, a 30-35% *upper rule* should be added. It means that 30-35% of test cases will be designed for *optimal* solutions only. These should be huge test cases on which *inefficient* programs get "time limit exceeded".

The most ambiguous point is why should the 50% score for inefficient solutions be reduced to 30-35%. One explanation is that three years ago, when the scores for inefficient solutions were about 20%, participants and IOI-related people thought that they were too small. Now it's clear that 50% is too much, and thus 30-35% would be a more appropriate percentage. Another explanation is that the score should be divided into three parts: one that tests inefficient solutions, another that tests efficient ones, and the third that tests intermediate solutions (i.e. that are between inefficient and efficient ones).

## 5 Rule advantages

It's clear that reducing score for *inefficient* solutions encourages participants to write *efficient* programs. However they can still receive enough points for

*inefficient* solutions if their skills are not enough for finding optimal ones. Inefficient solutions with optimizations can still get a good score, but not 80-90% as it's with 50% rule. Adding upper rule will help participants to understand what score they will receive for a certain solution. Now it's guaranteed that an *efficient* solution gets at least 30% more score than an *inefficient and optimized* solution.

## 6 Summary

Above were discussed the 50% rule's problems and their possible solutions. I hope that explanations contained in this document are enough for understanding the necessity of modifying the 50% rule to a more appropriate one.