Published: March 20, 2023 | Pages: 184-186

THE IMPORTANCE OF AN AUTOMATED CHECKING SYSTEM AND ITS APPLICATION IN EDUCATIONAL INSTITUTIONS

Abdukarimov Sirojiddin Sayfiddin Ugli

Chief Specialist Of The Ministry Of Preschool And School Education Of The Republic Of Uzbekistan, Tashkent, Uzbekistan

ABSTRACT: This article explores the significance and application of automated checking systems in the realm of educational institutions, particularly focusing on their utility in programming competitions and academic assessments. With the increasing prevalence of software solutions, there is a pressing need for efficient and fair testing methods. The paper discusses the current landscape of automated testing, highlighting the lack of literature on automated task testing systems despite their successful implementation in various countries.

KEYWORDS: Automated testing, educational institutions, programming competitions, software solutions, testing methodologies, system automation, competition equality, problem uniqueness, technical complexity, online challenges, real-time competitions, educational innovation.

INTRODUCTION

At present, automated testing of software solutions is widely discussed. Currently, various methods of automated testing and software products based on their application have been developed and are successfully used abroad. However, the majority of such products are designed for competitions with specific regulations and are practically not subject to modification to support competitions with significantly different rules of conduct.

The object of research is the automation of the process of conducting programming competitions and determining the correctness of software solutions, as well as using such a system for the continuous assessment of the knowledge of pupils and students.

Firstly, it ensures the absence of human involvement in the checking process. Secondly, it allows for the competition to be conducted on an equal footing for all participants.

At present, there are practically no articles dedicated to automated task testing systems, although several such systems have been created and are successfully operating in different countries. However, only the authors can work with most of them, not to mention expanding their capabilities. At the same time, an increasing number of various programming competitions are being held, at most of which participants submit the source code of their programs as a result of their work. Subsequently, these programs need to be evaluated in some way.

In the case of small-scale competitions (where participants submit several dozen solutions), manual testing methods are often used. In this scenario, a jury member independently compiles each submitted solution, runs it on a sequence of tests from a pre-determined (but not necessarily

Published: March 20, 2023 | Pages: 184-186

known to participants!) set, and finally evaluates the result produced by the program on each of them.

This approach is associated with significant time expenditure and manual labor. In particular, as experience shows, incorrectly written programs by participants often interfere with the normal operation of the checker's computer, as a result of which the latter is forced to perform frequent reboots, further increasing the time costs. Clearly, this approach is impractical under more intense loads.

Let's now consider how the automation of the competition process could be applied not to competitions, but, for example, to course projects, or for selecting the best pupils and students over a semester by conducting a small competition based on the materials studied throughout the semester.

But before we start creating such a system, we first need to understand how similar systems work. Let's take a look at an example of the automation of the process of conducting programming competitions.

A serious step is the automation of running participants' solutions on tests. Typically, a software complex is created for this purpose, which limits the actions that a participant's solution can perform (such a restricted environment is usually called a "sandbox"). In particular, the time and amount of memory used by the tested program are strictly limited. This accelerates the testing process several times over and, most importantly, allows for a much more accurate assessment of the time needed to test a single participant's solution, enabling the use of such a testing system in real-time competitions. Such automation makes it possible to reasonably check hundreds, rather than tens, of solutions.

Testing systems of this type are successfully used even in competitions at the level of the World Finals of the ACM International Collegiate Programming Contest. However, since the verification of the correctness of the result issued by the participant's solution on a test is performed manually, this method also has significant drawbacks. One of them is the almost complete impossibility of checking problems with ambiguous solutions, while such types of problems are often encountered in practice. Frequently, to ensure the uniqueness of the solution, certain restrictions are introduced into the problem, for example, the obtained solution must be the first in lexicographical order. These restrictions usually do not affect the algorithmic complexity of the problem, but often significantly increase its technical complexity, which, obviously, is a negative factor.

Such drawbacks can also arise when applying this system to a different approach, in which case, in most instances, it will be necessary to slightly modify the form of the answer, but without changing the main question of the problem, to facilitate its verification.

Therefore, to eliminate errors, or more precisely, to ensure that all participants are on equal footing, as well as to create testing systems designed for round-the-clock use, complete automation of the testing process is necessary. In particular, the role of an administrator of such a system is reduced to adding new problems available for testing and, possibly, expanding the available computational resources. Meanwhile, automatic testing systems can be applied in all

Published: March 20, 2023 | Pages: 184-186

types of competitions, including round-the-clock real-time competitions. They can test thousands of solutions per hour, which is essential for conducting online competitions.

Furthermore, full automation of the testing system will allow for the testing of a significantly greater number of participants' programs within the allocated time. During real online competitions, the system sometimes has to test dozens of programs per minute, which is clearly infeasible with either manual or partially automated testing.

Let's consider examples of existing automated testing systems that are successfully used for conducting various competitions. Such systems fall into two almost non-overlapping classes—systems for conducting online competitions and systems for conducting on-site competitions. The significant difference between the former and the latter is that the online systems operate around the clock, virtually without administrator intervention, while the systems for conducting on-site competitions are used only for a brief period during the competition itself and are constantly monitored by an administrator. Let's look at the main representatives of the class of systems for conducting online competitions and problem archives with automated checking.

Systems for conducting online competitions require constant support and replenishment of the set of problems available for testing. Therefore, automated testing systems of this class are not widely distributed.

Here are a few examples of applying similar systems:

A database of tests on various subjects, where n tests are randomly selected and checked for correctness against answers already stored in the database;

A platform for Olympiad participants with the option of virtual participation. One can choose by the year of the Olympiad and by difficulty level;

Visual mathematical tasks to develop thinking and imagination for elementary grades;

For conducting various competitions within the university and/or school to identify the most talented.

Of course, implementing all of the above examples will have its challenges, but the resulting outcome will be worthwhile.

REFERENCES

1. Корнеев Г.А. Автоматизированная система тестирования программ. // Материалы VIII международной конференции "Современные технологии обучения <<СТО-2002>>". 24 апреля 2002 года. -- Том 2, с.327-329. -- СПб.: СПбГЭТУ, 2002.
2. S.S. Abdukarimov, «Dasturiy yechim to‘g‘riligini avtomatik testlovchi tizimlar yordamida darslarni interfaol tashkil etish(C++ da dasturlash fani misolida)» Bitiruv malakaviy ishi, 2017.
3. de Boer, R. H., & de Campos, C. P. (2019). A retrospective overview of international collegiate programming contest data. Data in Brief, 25 doi:10.1016/j.dib.2019.104382
4. Г.А. Корнеев., Р.А. Елизаров. Автоматическое тестирование решений на соревнованиях по программированиюОпубликовано в журнале «Телекоммуникации и информатизация» образования. 2003, №1 .– с. 61–73.