CLIS

The 2018 ICPC World Finals

icpc.foundation

acm International Collegiate Programming Contest
icpc 2018 World Finals
hosted by Peking University and CYSC in Beijing

CLIS Schedule

Tuesday, April 17, 2018, 20:00-22:00

Marco Polo - Rome Ballroom

20:00-20:25 Contest API - An ICPC standard for contest systems & tools, Jaap Eldering

20:30-20:55 Next generation of VIVA: vanb's Input Verification Assistant, David Van Brackle

21:00-21:25 The North American Online Practice Contests Borja Sotomayor

21:30-21:55 The Book Series "Collegiate Programming Contests and Education", Yonghui Wu

Wednesday, April 18, 2018, 20:00-22:00

Marco Polo - Rome Ballroom

20:00-20:25 Programming Training League: A System Organizing Programming Training Cross Region, Yonghui Wu

20:30-20:55 Asura: An Environment for Assessment of Programming Challenges using Gamification José Paulo Leal

21:00-21:25 Competitive Programming and Sustainability Anup Kalbalia

21:30-21:55 Gamification in Computer Science Jeffrey Paone

Jaap Eldering has been involved in the ACM ICPC since 2001. First as competitor, and then organizer and jury member at NWERC. As one of the developers of the DOMjudge contest control system, he has been involved in the systems group at the World Finals since 2012. He currently works at Booking.com in The Netherlands.

Contest API - An ICPC standard for contest systems & tools

An ICPC standard for contest systems & tools We shall present the ICPC Contest API, a REST interface for communication between contest control systems (CCS), live scoreboards, results presentations, Contest Data Servers (CDS) as provided for example by the ICPC Tools, and anything new you can come up with…

The Contest API specification was developed as part of a year-long collaborative effort by a group of people representing all CCSs and other software tools (such as the CDS) used at the World Finals. It merges various data exchange formats used at the World Finals in the past into a single standard that is useful in a general ICPC context. This year at the World Finals it will be used for communication between primary and shadow CCSs, the CDS, and ICPC Analytics and ICPC Live.

We shall give a quick overview of the API, and describe how it improves previous specifications. Then we continue with a live demonstration of some of the systems and ICPC tools that already support this new API. The Contest API is currently supported by the following systems:

PC^2
Kattis
DOMjudge
ICPC Tools (CDS, live scoreboard and resolver, balloon printer, …)
AutoAnalyst tools (work in progress)

Mr. Van Brackle has been involved in ACM/ICPC programming contests for over 35 years, as a competitor, coach and judge. As a competitor, he competed in Finals twice, his team placing 2nd in 1987. As a coach, he has led 10 Finals teams, 5 each for two different schools: UCF and Georgia Tech. He has been the Chief Judge of the ICPC Southeast USA Region since 2008, and of the University of Chicago's North American Invitational Programming Contest (NAIPC) ever since its inception in 2012. He is an active participant in the North American RCD's group, the Southeast USA Regional steering committee, the NAIPC steering committee, and the North American Championship steering committee.

Next generation of VIVA: vanb's Input Verification Assistant

Each year, at every ICPC contest across the globe, the contest's Chief Judge is tasked with constructing a problem set, and with it, the Judge's Data that will be used to test contestants' submissions. Ensuring the accuracy of this data can be a challenge. Many of the judges write solving programs, to confirm the validity of the judge input and output, but even still, there are issues that can go undetected. Extra spaces, blank lines, values outside of specified ranges, and so on, can creep into the judge input. The judges are mainly former competitors who write very robust programs, which are largely immune to troubles caused by such formatting and constraint errors. However, these kinds of issues can cause problems with competitor's code during the contest, causing, at best, extra effort and delays by the judges during the contest, and at worst, incorrect judge responses.

Many regions develop programs in a standard programming language to verify the correctness of their judge data. Indeed, the Kattis contest control system requires an input format validator for

every problem. But, while these programs reduce the risk of data errors, they can be subject to the same sorts of oversights as judges' solving programs.

VIVA is a program to help verify that judge input conforms to the standards of a given contest, and that the data falls within the constraints specified in the problem statement. It reads a pattern specified in its own simple pattern specification language. It then reads input files, and determines if (and where) they violate the specified constraints.

VIVA was developed in 2010, and was introduced in a presentation at the CLIS in Orlando in 2011. It has undergone a great deal of improvement since that initial version was released, including documentation, bug fixes, enhanced capabilities, and new capabilities. This presentation will briefly introduce VIVA for those who are unfamiliar. It will then describe, in detail, the improvements that have been made to VIVA since the initial version was introduced in 2011. It will also describe ideas for further improvements to VIVA, and describe how any interested party can access VIVA, and provide feedback.

Borja Sotomayor is a Senior Lecturer in Computer Science at the University of Chicago. He is also the coach of the UChicago teams, the Director of Systems for the USA Mid-Central Regional Contest, and the Director of the North American Invitational Programming Contest.

The North American Online Practice Contests

There are many ways in which ICPC teams can prepare for a contest, including doing practice contests that replicate, as much as possible, the experience of being in a real contest. While seasoned ICPC coaches typically have the in-house expertise to run a contest management server and hold practices for their teams, the same is not necessarily true for schools and coaches that are new to ICPC, who may find the experience of setting up a contest to be daunting. This, in turn, can lead to teams not getting enough preparation before a regional, discouraging these newer teams, and ultimately affecting retention in ICPC. To address this issue, as well as to facilitate the work of the more experienced coaches, the North American regions have been running a series of online practice contests since 2015 (http://naipc.uchicago.edu/practices/). These practice contests are run on Kattis on weekends, with several practice contests scheduled before the North American regionals, and another series of practice contests scheduled before the North American Invitational Programming Contest (in March/April) and the World Finals. In this presentation, we will describe how the practice contests are or- ganized and run, and will share insights and lessons learned from the three years the practices have been running.

Yonghui Wu, Ph.D, Associate Professor, Fudan University. He acted as the coach of Fudan University Programming Contest teams from 2001 to 2011. Under his guidance Fudan University qualified for ACM-ICPC World Finals every year and won three medals (2 bronze medals and 1 silver medal) in ACM-ICPC World Finals. Since 2012, he has published a series of books for programming contest and education in simplified and traditional Chinese and English. Since 2013, he has given lectures in Oman, Taiwan, HongKong, Macau, Malaysia, Bangladesh, Mainland China and the United States for programming contest training. He is the chair of ACM-ICPC Asia Programming Contest first Training Committee now.

The Book Series "Collegiate Programming Contests and Education"

Since 2012, a book series "Collegiate Programming Contests and Education" has been published. This series has been published in simplified and traditional Chinese and English: the former by respective publishers of mainland China and Taiwan, and the latter, the first book's translation, by CRC Press. The goal for the book series is polishing students' programming skill solving problems in a systematic way. A large number of programming contests' problems from all over the world can be gotten, analyzed and solved. These problems can be used not only for programming contest training, but also for education. The programming knowledge system can be summarized as a famous formula: "Algorithms + Data Structures = Programs". Strategies solving problems are strategies for data modeling and algorithm design. Three books constitute the book series:

[1] Data Structure Practice: for Collegiate Programming Contests and Education;
[2] Algorithm Design Practice: for Collegiate Programming Contests and Education;
[3] Programming Strategies Solving Problems: for Collegiate Programming Contests and Education.

The outlines for the book series are based on the outlines of data structure and algorithms. Programming contest problems and their analyses and solutions are used as experiments. For each chapter, there is a section "Problems" to let students solve programming contests' problems, and hints for these problems are also showed. The book series can be used not only for systematic programming contest training, but also for polishing computer students' programming skill better, using programming contests' problems. Now we improve the book series, and set up International Joint Lab for Programming Technologies (IJLPT) based on the book series.

Programming Training League: A System Organizing Programming Training Cross Region

Yonghui Wu1, Jingshan Yu2, Xuefeng Jiang3, Sheng-Lung Peng4 1(School of Computer Science, Fudan University, yhwu@fudan.edu.cn) 2(Ningxia Institute of Science and Technology, yujingshan@hrbeu.edu.cn) 3(School of Computer Science, Northwestern Polytechnical University, jxf@nwpu.edu.cn) 4(Library and Information Center, National Dong Hwa University, slpeng@mail.ndhu.edu.tw)

Since 2012, a book series for programming contests and education based on programming contest problems has been published in simplified and traditional Chinese and English. And lectures for programming training have been given in Taiwan, HongKong, Macau, Mainland China, Oman, Bangladesh, Malaysia, America, and so on. Based on these works, ACM-ICPC Asia Training League, which involved more than twenty "211 universities" (denoting the top one hundred universities of Mainland China) of Mainland China and several universities in HongKong, Macau, and Taiwan, has been organized.

Goals for ACM-ICPC Asia Training League are as follows.

[1] Educational and experimental: polish students' programming skills to help them better solve programming problems.

[2] For programming contests: help students train more systematically.

[3] Educational: MOOCs (online courses) to set up based on the series.

The modes of operation for ACM-ICPC Asia Training League are offline courses, online courses, and online mock programming contests, with the help of test data, solutions and analysis. The current works for ACM-ICPC Asia Training League are organizing systematic programming training and establishing International Joint Lab for Programming Technologies (IJLPT). In the future, we'll improve and popularize our work to the whole world through IJLPT.

José Paulo Leal is assistant professor at the Computer Science department of the Faculty of Sciences of the University OF Porto. He is affiliated with the Center for Research in Advanced Computing Systems (CRACS), an R&D unit of INESCTEC Research Laboratory, where he is an effective member. His main research interests are technology enhanced learning, web adaptability, and semantic web.

Asura: An Environment for Assessment of Programming Challenges using Gamification

José Paulo Leal
CRACS & INESC-Porto LA,
Faculty of Sciences, University of Porto,
Porto, Portugal
zp@dcc.fc.up.pt

José Carlos Paiva
CRACS & INESC-Porto LA,
Faculty of Sciences, University of Porto,
Porto, Portugal
up201200272@fc.up.pt

Loss of motivation is one of the most outstanding problems in education. When students are not motivated, they tend to care less about educational activities and to stop striving to complete them. In courses that require lots of practice, especially in programming courses, this results in an unsustainable lack of practice which is accompanied by recurring failures in assessments and later ends up in student dropout.

Several approaches have been proposed to mitigate this problem. One of such proposals, which was coined as gamification, uses game elements and mechanics to engage students. The most common gamification methods simply add some game elements such as leaderboards, badges, or levels to an existing learning

environment. However, various experiments linked these methods to serious educational issues such as decreasing intrinsic motivation, blocking the development of self-determination and independent thinking, and encouraging reward-driven behaviors. More effective methods typically rely on different game aspects, such as graphical feedback, game-thinking, collaboration, competition, and in-game challenges, to bring the student into the game world. In respect to programming learning, there is the possibility of having the student to code the Software Agent (SA) that controls the player and, thus, bring the game completely into learning. This opportunity has already been explored in several games, such as CodeRally - a Java game-based rally competition presented at the 2003 ACM International Collegiate Programming Contest (ICPC) World Finals - and Robocode - a Java-based virtual robot game. Both of these games demonstrated a great potential to catch students and non-students attention. Nevertheless, building these challenges involves a complex process which, most times, educators are not willing to perform.

This paper presents Asura, an environment for assessment of programming challenges using several gamification methods. Firstly, the student must solve a programming challenge in the form of a game (i.e., he/she must code an SA). After understanding and solving the proposed challenge, the student is encouraged to produce that beats the opponents (SAs from other students). To complete this task, the student benefits from the game-like graphical feedback on the SA behavior during the game. Once the challenge ends, educators are able to organize tournaments, similar to those found on traditional games and sports, among SAs submitted by the students. A tournament produces a movie which contains a reference to each match's movie, organized by stages and rounds, as well as partial and complete rankings of each phase. The movie of the tournament is presented in an interactive GUI in which the viewer can control what games he wants to see. Furthermore, one of the key goals of Asura is to enable teachers to build games with a similar complexity to that of creating an ICPC-like problem. These games can be very simple, such as a number

guessing game, or more complex than CodeRally and Robocode.

An experiment with Asura will be conducted in the laboratory classes of an undergraduate Object-Oriented Programming (OOP) course at Escola Superior de Media Artes e Design (ESMAD) - a school of the Polytechnic Institute of Porto. This course aims to introduce Javascript and some basic OOP concepts integrated into its ECMAScript 6 version, to students with little to no programming background. This experiment will evaluate both the effectiveness of Asura in motivating students to practice programming and compare the difficulty of building this kind of challenges with that of creating programming exercises in traditional automated assessment.

Anup Kalbalia,
Sr. Tech Lead, Directi

Leads www.codechef.com.
Have an experience of over 13 years of building software products in the finance, internet and education domain.

Education:
Bachelor of Engineering in Information Technology, WBUT Master of Science in Information Systems, BITS Pilani

Competitive Programming and Sustainability

Competitive Programming or the Sports of programming is picking up. There is plenty of traction in this field with new competitions and tournaments coming up, more students getting involved in it, more organizations being aware of it and getting into it, and new training material and platforms being created. However, there are also some adverse signals to be seen like the number of slots and prizes going down in some of the established onsite regionals, and some big companies turning away in terms of sponsorships.

In this world, where technology is getting more pervasive and the need for people with a deep understanding of algorithms and data structures is growing, the role of competitive programming is getting larger and more decisive. The sport has become instrumental for organisations to find the right talent and it is also helping educationists make an important subject interesting and easy to teach. While there are clear advantages of competitive programming, a question lingers in the minds of the players: Can competitive programming be made a true sport? Can it evolve beyond just being a tool for finding and showcasing talent for hiring purposes, and become a sustainable sport on its own? Sustainability is a very important aspect for any sport and this talk aims to make an attempt in finding answers to some such questions, through the experience of last 9 years in which CodeChef has been trying to promote this as a sport.

About CodeChef - CodeChef, a non profit initiative of Directi, is a global programming community that fosters learning and friendly competition, built on top of the world's largest competitive programming platform. It organises three featured contests every month and gives away prizes and goodies to the winners as

encouragement. Apart from this, the platform is open to the entire programming community to host their own contests free of any charge. Major institutions and organisations across the globe use our platform to host their contests. On an average, 30+ external contests are hosted on our platform every month. We also partner with colleges and groups to create local CodeChef Chapters, meet-ups, orientation sessions and programming workshops.

CodeChef has also been hosting the online and onsite ICPC regionals for India over the last few years.

Dr. Jeffrey Paone is a Teaching Associate Professor and Assistant Department Head of Computer Science at Colorado School of Mines. His research interests include biometrics, computer vision, computer graphics, and augmented reality. He regularly teaches CS1, mobile development, web development, and computer graphics which all employ a variety of active learning techniques, including gamification. These techniques have led to an increase in student engagement and continually have received positive student feedback each semester.

Gamification in Computer Science

Gamification is the application of gaming elements applied to a non-gaming context. For educators, gamification is the application into a classroom setting and a given course. The gaming elements must be included in course design to fully immerse a student in the world of the course. Care must be taken to properly frame each course component in the proper context of the world. To provide a competitive nature and encourage exploration deeper into the world of the course, badges are awarded to students for completing known and hidden tasks. Students are then ranked on a leaderboard for their progress throughout the world. It is important to carefully balance the competitive nature of the gamification with the creativity and exploration that the world provides. The leaderboard and competition must not be a discouragement for students that are not at the top of the rankings and it is critical to build in mechanisms to allow for a student to catch up later in the semester.

Dr. Paone has successfully applied gamification to his Computer Graphics course at Colorado School of Mines for three years. Each year, the gamification framework has been tweaked slightly to provide improvements on the prior iteration. Additionally, a history has been established to expand the world and increase the ability for competition. He will discuss and present strategies to apply

gamification to a course, lessons learned from gamification, and share student feedback provided after students had completed the course. These strategies can then be applied to additional courses and programs to enhance student learning.