Programming Contests, Competitive Programming Curricula, and how Everybody Benefits

Arup Guha
Department of Computer Science
University of Central Florida
Orlando, USA
dmarino@ucf.edu

Jerzy W. Jaromczyk
Department of Computer Science
University of Kentucky
Lexington, USA
jurek@g.uky.edu

Christian Yongwhan Lim
Department of Computer Science
Columbia University
New York, USA
yongwhan.lim@columbia.edu

Benjamin Reed
Department of Computer Science
San Jose State University
San Jose, USA
ben.reed@sjsu.edu

Etienne Vouga
Department of Computer Science
University of Texas at Austin
Austin, USA
paul.vouga@austin.utexas.edu

***Abstract*—The panelists, all competitive programming team coaches and college educators, share their experiences in teaching and working with competitive programming students and teams. The panel discusses using the model of the International Collegiate Programming Competition (ICPC) in various academic settings such as typical college courses as well as competitive programming courses. We compare various competitive programming courses and different ways of incorporating components of programming competitions into computing courses. We discuss the payoffs for participating students of varying backgrounds and interests, aptitudes, and motivations, as well as payoffs to industry and academia in terms of developing the best-prepared workforce. We extract best practices and provide practical suggestions for how attendees can effectively leverage existing tools and materials into their own courses and extracurricular activities.**

***Index Terms*—competitive programming, education, problem solving, workforce**

I. Summary

Competitions in various academic disciplines have been around for a long time to engage students and identify, develop, and retain talent. It is natural that educators, and later professional organizations and leading industry players, created and sponsored Olympiads, contests, and competitions in programming at different levels and formats. Several universities have recently added a 'competitive programming" course on algorithmic problem solving to their curriculum. Clearly, these courses prepare students for programming competitions. However, there is another fundamental motivation: topics and activities included in preparations for programming competitions benefit all of the participating students, not only those exceptionally gifted or competitive. Furthermore, relevant courses contribute to student learning outcomes and implicitly benefit future employers with a better-educated and skilled workforce. Following the approach from H. Walker's *"What is the Payoff?"* (see ACM INROADS, 2022, Vol 13, No 1), we look at those benefits as payoffs to everyone on the path from school to industry, to academia, and to society. That perspective also guides the design of materials and pedagogies for courses, practices, or extracurricular activities: by optimizing the desired payoffs, instructors ensure that every student can make capital out of participation. The payoffs for students include:

- Strengthening algorithmic problem-solving;
- Reinforcing program development techniques with testing and disciplined coding that produces correct and efficient programs;
- Strengthening communication, planning, and teamwork skills;
- Gaining white-board interviewing confidence.

For education, the payoff is in:

- Providing a structured, rigorous and inspiring environment for students;
- Helping students to identify their interests;
- Building students' sense of belonging;
- Identifying highly gifted students and exposing them to challenging tasks.

All of those directly translate into payoffs for industry, academia and, consequently for society, with a well-educated and prepared workforce.

II. Panel Structure

We will open with a short description of the context and goals of the panel. Then each panelist will have 10 minutes to share their experiences and formulate suggestions. Following the presentations, the moderator will facilitate audience questions and involve the attendees in a discussion on ranking the payoffs, as described above, with respect to their importance in motivating and inspiring students to greater efforts, and supporting broad student learning outcomes.

III. Panelists

A. Arup Guha

In order for the University of Central Florida to compete with schools across the country in the ICPC, students must work together to solve problems that they would not have been able to solve independently. This focus on teamwork has been translated into helping the general undergraduate Computer Science population excel in competitive programming in teams in the course UCF course "COP 4516C: Problem Solving Techniques and Team Dynamics." Encouraging leadership, collaboration and camaraderie helps teams reach their potential.

Arup Guha has helped coach the UCF programming team, one of the top teams in the United States, for over the past 20 years. He has used his coaching experience to revamp COP 4516 so that a greater number of students benefit from solving algorithmic problems together.

Arup will discuss how to help competitive programming teams reach their potential, regardless of their initial background in programming competitions.

B. Jerzy W. Jaromczyk

Jerzy W. Jaromczyk has been a faculty in the Computer Science Department of the University of Kentucky for more than three decades, an Honor Faculty in the Lewis Honors College, engaged with the Secondary STEM Education Program at the College of Education. He served as a Director of Undergraduate Studies, mentored numerous high schoolers, promoted diverse participation in computer science education, organized and contributed to multiple outreach and enrichment programs ranging from K-12 to STEM teachers at the local and KY state level. He has coached competitive programming teams for twenty-five years. Jerzy will discuss the structure of competitive programming activities at the University of Kentucky and how those activities affected classes, enrichment and outreach programs for diverse groups exposing them to computational thinking. Furthermore, he will show how competitive programming methods advise assignments in CS classes. Specifically, he will describe his *Design your programming assignment yourself* exercise incorporated in the Honors Algorithm course to inspire creativity and deepen understanding of algorithms, their applicability, and nuances of testing. This exercise, liked by students, combines creative technical writing with training in putting to use the Extend/Reflect paradigm and heuristics recommended in George Pólya's *How to Solve it?* book.

C. Christian Yongwhan Lim (Moderator)

Christian Yongwhan Lim is VP of Engineering at Arklex AI, where he designs and builds systems that move fast without breaking. He architected and led the development of the open-source Agent-First Organization framework—175K+ lines of modular Python powering enterprise-grade AI agent orchestration.

He is also an Adjunct at Columbia University, coaching the ICPC programming team and leading global initiatives in algorithmic learning and internship programs. Christian serves as Director of Internships at the ICPC Foundation and Chief Judge of the Mid-Central North America Regional. Previously, he spent nearly six years as a Research Software Engineer at Google and over half a year as a Senior Quantitative Software Engineer at Two Sigma Investments.

Passionate about teaching, Christian develops courses on technical interview preparation and competitive programming, including Technical Interview Prep in C++ and Competitive Programming at Columbia. He has coached the Columbia ICPC team for several years and regularly leads workshops on technical interviews worldwide.

In his session, Christian will share strategies for effective preparation for tech and quantitative interviews, and explore how educators can help students achieve strong learning outcomes and successful academic and industry placements.

D. Benjamin Reed

Benjamin Reed has been a faculty in the Computer Science Department of San Jose State University for five years and recently changed to the Computer Engineering Department. San Jose State University did not have a competitive programming team and had not participated in ICPC for many years. He restarted San Jose State University's team in 2019.

Benjamin will discuss the challenges and motivation for starting a programming team at a university without a history of participating in ICPC. In 2021, he realized that the confidence and the ability to implement algorithms that students gained from competitive programming could benefit all students, leading to the development of two one-unit courses spread across two semesters for all students. He will share the experience of teaching these courses with students building their programming skills and other students who needed to strengthen their skills in preparation for upper-division computer science courses.

E. Etienne Vouga

Etienne Vouga is an associate professor of computer science at the University of Texas at Austin. He has taught Competitive Programming as a one-hour elective each semester for the past six years, and is also the head coach of UT Austin's ICPC teams.

Etienne will discuss factors influencing the curriculum of competitive programming classes, and lessons learned from designing such a class at UT Austin. He will address the pros and cons of targeting beginning students who haven't yet formally studied algorithms and data structures vs. teaching advanced algorithms to upper-division students; the tension between topics covered in theoretical algorithms courses, programming contests, and technical job interviews and how to align the curriculum to maximize student payoffs; and how the large pool of existing competitive programming problems and ongoing programming contests open to the public can be leveraged for student assessment.

IV. Acknowledgement

We would like to thank other significant contributors: Dante Bencivenga, Rezaul Chowdhury, Glenn Downing, Robert

Kramer, James Lembke, Simanta Mitra, and Nouhad Rizk. J. W. Jaromczyk acknowledges partial support of the NSF Noyce grant # 1000100147 (PI: Margaret Schroeder): *Producing STEM Teachers for Rural and Urban Schools in Kentucky*.