Developing a Course on Competitive Programming

Waqar Saleem

waqar.saleem@sse.habib.edu.pk

GRANT STREAM: SUMMER TEHQIQ RESEARCH PROGRAM

Abstract

Competitive programming provides a fun setting for algorithmic problem solving and computer programming—participants compete as teams within a fixed time. The skills required for competitive programming align well with the learning objectives of a computer science undergraduate program. This proposal is to involve undergraduate students in exploring the best practices and methodologies for teaching competitive programming to undergraduate liberal arts students. The research entails surveying current literature and courses at other universities in order to study their pedagogy, activities, reference material, and course organization; become familiar with the various online platforms that they use; and identify topic-specific resources from these platforms. Best practices to measure the effectiveness of a course will also be studied. The obtained insight and resources will then be used to develop the lectures, activities, assessments, and grading scheme for a semester-long course on competitive programming that caters to the growing community of interested DSSE students. A set of assessment metrics and methodologies for evaluating the effectiveness of the course will also be developed. All the research and development will be carried out by student researchers, and correspondingly they will be responsible for all of the outcomes. They will prepare a report at the end in order to document their experience and findings which will form the basis of an academic publication once the course has been offered a few times and data about its effectiveness becomes available.

1 Introduction and Problem Statement

Programming competitions have become increasingly popular over the years, with many participants, both professionals and amateurs, competing for recognition and prizes. Competitive programming competitions [39] involve solving algorithmic problems within a specified amount of time, typically a few hours. These competitions are designed to test the participants' problem-solving and programming skills under time pressure. Competitive programming has become a global phenomenon, with contests held regularly in various countries.

Often conducted as a team sport, competitive programming is a valuable tool for improving programming skills, particularly among undergraduate students in computer science programs. Competing requires a high level of knowledge of data structures, algorithms, and problem-solving skills, which are fundamental to computer science education. It provides an engaging and challenging environment for students to develop their programming skills and apply them to real-world problems. Technology companies like Google, Microsoft, Amazon, Apple, and Facebook, as well as local companies like Folio3, securiti.ai, Creative Chaos, and Bazaar, famously test potential employees for exactly these skills [23], often in settings similar to competitive programming. Some of these companies run their own competitions [7, 24, 11, 1, 2] in order to promote and identify talented programmers.

Habib University's emphasis on critical thinking, creativity, and problem-solving has naturally led DSSE students to partake in competitive programming. Specifically, they competed in International Collegiate Programming Contest (ICPC) [28], which is the largest and most prestigious computer programming contest for students at the undergraduate level. Colleges are represented by teams of three students each under the tutelage of a coach who is typically a faculty member. The annual competition proceeds in rounds—national, regional, and a world final. DSSE students have qualified for and appeared in the top 10 of the regional round for three consecutive years [10, 16].

Recognizing the value of competitive programming in a computer science education, several universities offer formal instruction in it. A survey of prominent courses is included in the next section.

This research proposal aims **to develop a Computer Science elective course on competitive programming, and propose metrics to measure its effectiveness**. The course will be offered to students with existing exposure to programming, data structures, algorithms, and programming languages. The course will provide students with the opportunity to enhance their problem-solving skills, collaborate with peers, and participate in programming contests, thereby preparing them for real-world programming challenges. It will also allow students a creative outlet for their studies in abstract computer science topics like data structures and algorithms. Indirectly, it will serve to popularize competitive programming on campus and prepare teams for ICPC. This in turn will make our students more attractive to industry and to graduate schools.

2 Literature Review

Courses on competitive programming have started to appear at some universities [26, 35, 33, 37, 31, 34, 9, 17, 36, 30]. The instructors are usually faculty members who have themselves been participates in ICPC or its school-level counterpart, International Olympiad in Informatics (IOI) [19], or are long-time coaches for these events [14, 21, 38, 25, 8, 4, 15]. Some of the enrolled students [32] similarly have experience in IOI, ICPC, International Mathematics Olympiad (IMO) [18], or many of the online competitive programming platforms [13, 27, 6, 20, 12, 29, 5]. Some instructors and their courses have amassed significant reputation. The instructors are invited by other universities to offer a version of the course [40], or students travel from other countries just to take the course [22].

The pedagogical value of these courses in their respective curricula is to develop the students' algorithmic, programming, and problem-solving skills. But they also serve to prepare motivated students to eventually represent the university at ICPC.

A typical ICPC-style programming competition starts by assigning a common set of problems to all the participating teams. They then have a specified amount of time to solve the problems and submit their solutions real-time over a network connection to an automated checking system. The system returns a result of correct or incorrect and the team can choose to correct their solution and resubmit or to attempt and submit the solution to another problem. Each team is scored based on the number of problems it solves correctly, and the time and number of attempts taken to submit the correct solutions. Teams that solve more problems quicker and in a smaller number of attempts are ranked higher.

The problems present scenarios which can be mapped to one or more algorithmic techniques. There may be more than one way to solve a problem and students are required to pick the solution that is the most efficient. Other teams are simultaneously and independently solving the same problems. The scores are updated real-time as the submissions happen and a live scoreboard with the score of all teams is centrally displayed so as to be visible to all the participating teams. The environment is generally tense and highly competitive.

Collectively, courses on competitive programming achieve multiple outcomes. By teaching algorithmic techniques and concepts, they provide the students the necessary intellectual background. Through curated problems on specific topics, they train the students in those topics. Through individual assessments, they ensure that each student has adequate practice and stands to become a valuable team member.

Frequent team competitions in a course serve to develop team dynamics and expose students to more realistic problems which do not rely on a single technique. They also acclimatize students to the high pressure environment of these contests. Extensive practice ensures that the students are familiar with a rich, diverse, and voluminous set of problems so that they may relate any new problem to one that they have previously solved.

Online platforms to deliver problems and assess solutions are imperative for these courses. These serve the dual role of question banks for practice and assessments, and of automated judges for the students' solutions during assessments and mock contests. Some courses have developed their own platforms while others use existing competitive programming platforms, of which there are several that are freely available online.

The field of algorithms is deep and vast. A semester-long course covers but a small portion. Some universities therefore offer a sequence of courses on competitive programming that build upon the usual introductory programming course [33].

2.1 Proposal

This research proposal aims to develop a course on competitive programming for the Habib community, keeping our constraints in mind. Our students and faculty have minimal exposure to competitive programming. There is a shortage of available faculty to deliver such a course. Our student body hails from diverse backgrounds and many of them lose a good portion of their day commuting to and from campus, and attending to their domestic responsibilities. A significant part of their curriculum consists of courses that are not directly related to computer science. The number of computer labs is not keeping pace with the number of students on campus. The infrastructure—electricity, internet—outside campus cannot be relied on completely.

The main aim of the proposed course is to develop students' skills to solve problems algorithmically and implement their solutions using a high level programming language in order to meet stringent constraints. Through inclusive activities, the course will popularize competitive programming on campus. It will also serve to train students who aspire to represent Habib University at ICPC or other competitive programming events or platforms.

3 Research Questions/Aims

The objectives of this research proposal are:

1. To explore the best practices and methodologies for teaching competitive programming to undergraduate liberal arts students.

2. To design a course on competitive programming that is appropriate for Habib University students.

3. To involve Habib University students in the research process, from review of available and relevant resources to course design.

4. To disseminate the findings of the research to the broader academic community.

4 Methodology

The research will involve several stages. Though they are presented sequentially below, some of them may overlap as per the project's needs and findings.

1. Course/literature review: The student researchers will identify courses on competitive programming through online search and survey them in order to identify the best practices and methodologies for teaching competitive programming to undergraduate liberal arts students. Any identified literature, e.g. from academic journals, conference proceedings, and other online resources will also be consulted.

   The review will yield pedagogies, assessment schemes, topical material, reference books, and applicable platforms.

2. Testing and resource identification: The various online platforms and reference material collected through Step 1 will be assessed for ease of use and adoption by the potential faculty and the students of the proposed course. The student researchers will sign up on the identified platforms and attempt problems and contests on them. They will document their progress and observations about these platforms, culminating in a comparative analysis based on ease of access and use, availability of topical problems for practice, possibility to hold mock contests, and other relevant factors identified through Step 1.

   The researchers will similarly survey the usual textbooks in order to assess them on exposition, coverage, and accessibility. Through my requests, copies of these books are already part of our library's catalog.

   In order to gauge the target level of difficulty and finalize suitable problems for the propose course, the researchers will perform selected assessments of selected courses.

3. Course design: Based on their findings from the first two steps, the student researchers will finalize a list of topics to be covered in the proposed course, accompanying teaching and assessment material, a lecture plan and an assessment plan for the course. Care will be taken to incorporate different teaching approaches, such as online resources, hands-on activities, in-class discussions, and programming contests, to cater to different learning styles and abilities. A few activities will be designed to include on a regular basis, outside students. These may be Habib students that are not enrolled in the course, as well other students who may be able to join remotely or physically as appropriate.

   The researchers will contact the instructors of other courses for guidance and potential guest talks in the course.

4. Effectiveness metrics: The researchers will survey relevant literature in order to identify the tools used to measure the effectiveness of such a course. They will then solicit student feedback through surveys and interviews in order to adapt these tools and techniques to the proposed course.

5. Write-up: The researchers will write a report documenting their work, findings, and experiences. They will be encouraged to populate this report alongside their work. It will serve as a basis for eventual publication at a suitable SoTL venue after some offerings of the course when data on the effectiveness of the course is also available.

5 Outcomes and Deliverables

The expected outcomes of this research proposal are:

1. The development of a course, i.e. its syllabus, on competitive programming specifically designed for students at Habib University.

2. The involvement of undergraduate students in the research process, providing them with valuable research experience.

3. A set of assessment metrics and methodologies for evaluating the effectiveness of the course.

4. The eventual dissemination of the findings of this research to the broader academic community.

6 Contribution to Undergraduate Research

a) GSCP Scholar Role: All the research will be performed by the GSCP scholars, referred to above as “student researchers” under the supervision of the PI. That is, they will perform the literature/course review, test and identify resources, design the course, identify and develop effectiveness metrics, and write the report.

b) Student skillset and Program preference: The ideal scholars for this project are passionate about programming and problem solving. They have an interest in pedagogy and designing courses. This can be identified through grades in CS 101, CS 102, CS 201, and CS 224; past TA experience within DSSE; and any participation in competitive programming events. No additional background is necessary.

c) Student Supervision: The scholars will learn to identify online information relevant to a given topic. They will become familiar with searching and reading academic publications. They will get insight into course design and measurement of course effectiveness. They will learn to work within a timeline on an independent basis and as part of a team. They will learn to catalog and document collected information in a formal manner. Supervision will occur through frequent and regular meetings with the supervisor.

d) Student Deliverables: All the deliverables of this project are the responsibility of the scholars. That is, the syllabus, the effectiveness metrics and techniques, and the eventual write-up.

7 Implementation Plan

Week	Activity	Faculty Research & Supervision	Undergraduate Research	Resources Required
1	Course/Literature review	Introduction of project	Surveying courses and literature	Internet connection
2	Course/Literature review	Progress meeting(s)	Surveying courses and literature	Internet connection
3	Testing	Progress meeting(s)	Testing identified resources	Internet connection, Library
4	Testing	Progress meeting(s)	Testing identified resources	Internet connection, Library
5	Testing	Progress meeting(s)	Testing identified resources	Internet connection, Library
6	Testing	Progress meeting(s)	Testing identified resources	Internet connection, Library
7	Course Design	Progress meeting(s)	Designing course syllabus	Internet connection, SimpleSyllabus
8	Course Design, Effectiveness Metrics	Progress meeting(s)	Designing course syllabus, Researching metrics	Internet connection, SimpleSyllabus
9	Course Design, Effectiveness Metrics	Progress meeting(s)	Adapting & incorporating effectiveness metrics	Internet connection, SimpleSyllabus
10	Write-up	Progress meeting(s)	Documenting the research	Internet connection

8 Ethical Considerations (if any)

None.

9 Total Budget in Pakistani Rupees [PKR]

None required.

10 Acknowledgments

In preparing this proposal, help was taken from ChatGPT [3] in order to refine the language, and for better presentation and formatting.

List of References

[1] 10Pearls. *10Pearls University holds WTQ 2022.* URL: https://10pearls.com/10pearls-university-holds-wtq-2022/ (visited on 03/20/2023).

[2] 10Pearls. *10Pearls University Hosts CodeFest Competition.* URL: https://10pearls.com/10pearls-university-holds-codefest-2017/ (visited on 03/20/2023).

[3] Open AI. *ChatGPT.* URL: https://chat.openai.com/chat (visited on 03/20/2023).

[4] Mattox Beckman. *Mattox Beckman.* URL: https://mattox.netlify.app (visited on 03/20/2023).

[5] *CodeChef.* URL: https://www.codechef.com (visited on 03/20/2023).

[6] *Codeforces.* URL: https://codeforces.com (visited on 03/20/2023).

[7] Google. *Google's Coding Competitions.* URL: https://codingcompetitions.withgoogle.com (visited on 03/20/2023).

[8] Bjarki Ágúst Guomundsson. *Bjarki Ágúst Guomundsson.* URL: https://algo.is (visited on 03/20/2023).

[9] Bjarki Ágúst Guomundsson. *T-414-ÁFLV: A Competitive Programming Course.* URL: https://algo.is/competitive-programming-course/ (visited on 03/20/2023).

[10] Habib Univeristy, Facebook. 2021. URL: https://www.facebook.com/HabibUniversity/posts/4210743022314945 (visited on 03/20/2023).

[11] HackerEarth. *Amazon - Events — Top tech companies of 2023.* URL: https://www.hackerearth.com/companies/amazon/events/ (visited on 03/20/2023).

[12] *HackerEarth.* URL: https://www.hackerearth.com (visited on 03/20/2023).

[13] *HackerRank.* URL: https://www.hackerrank.com (visited on 03/20/2023).

[14] Steven Halim. *World of Seven - Steven Halim's Personal Website.* URL: https://www.comp.nus.edu.sg/~stevenha/ (visited on 03/20/2023).

[15] Greg Hamerly. *Greg Hamerly.* URL: https://cs.baylor.edu/~hamerly/ (visited on 03/20/2023).

[16] ICPC Pakistan, Facebook. 2022. URL: https://www.facebook.com/permalink.php?story_fbid=297527922485884&id=141825621389449 (visited on 03/20/2023).

[17] University of Illinois Urbana-Champaign. *CS 491 Competitive Algorithmic Programming.* URL: https://courses.engr.illinois.edu/cs491cap/fa2019/ (visited on 03/20/2023).

[18] *International Mathematical Olympiad.* URL: https://www.imo-official.org (visited on 03/20/2023).

[19] *International Olympiad in Informatics.* URL: https://ioinformatics.org (visited on 03/20/2023).

[20] *LeetCode.* URL: https://leetcode.com (visited on 03/20/2023).

[21] Ninghui Li. *Ninghui Li's homepage.* URL: https://www.cs.purdue.edu/homes/ninghui/ (visited on 03/20/2023).

[22] Jay Lim. *Getting into Competitive Programming.* URL: https://imjching.com/writings/2019/06/21/getting-into-competitive-programming/ (visited on 03/20/2023).

[23] G.L. McDowell. *Cracking the Coding Interview: 150 Programming Interview Questions and Solutions*. CareerCup, LLC, 2011. ISBN: 9781466208681. URL: https://books.google.com.pk/books?id=anhAXwAACAAJ.

[24] Meta. *Meta Coding Competitions*. URL: https://www.facebook.com/codingcompetitions (visited on 03/20/2023).

[25] Jaehyun Park. *Jaehyun Park*. URL: https://jaehyunp.github.io/profile/ (visited on 03/20/2023).

[26] National Univeristy of Singapore. *CS3233 - Competitive Programming*. URL: https://www.comp.nus.edu.sg/~stevenha/cs3233.html (visited on 03/20/2023).

[27] *The Hitchhiker's Guide to the Programming Contests*. URL: https://comscigate.com/Books/contests/icpc.pdf (visited on 03/20/2023).

[28] *The ICPC International Collegiate Programming Contest*. URL: https://icpc.global (visited on 03/20/2023).

[29] *Topcoder*. URL: https://www.topcoder.com (visited on 03/20/2023).

[30] Baylor University. *CSI 4144: Competitive Learning, Spring 2022*. URL: https://cs.baylor.edu/~hamerly/courses/4144_22s/ (visited on 03/20/2023).

[31] Carnegie Mellon University. *15-295: Competition Programming and Problem Solving*. URL: https://contest.cs.cmu.edu/295/ (visited on 03/20/2023).

[32] Purdue University. *Competitive Programmers Union*. URL: https://boilerlink.purdue.edu/organization/cpu (visited on 03/20/2023).

[33] Purdue University. *Competitive Programming Courses at Purdue Computer Science*. URL: https://www.cs.purdue.edu/homes/ninghui/courses/cpx_index.html (visited on 03/20/2023).

[34] Stanford University. *CS 97SI: Introduction to Programming Contests*. URL: https://web.stanford.edu/class/cs97si/ (visited on 03/20/2023).

[35] Stony Brook University. *CSE 392 - Programming Challenges*. URL: https://www.cs.stonybrook.edu/about-us (visited on 03/20/2023).

[36] Texas A&M University. *Problem Solving Programming Strategies – CSCE430/2021Spring*. URL: https://tamu.kattis.com/courses/CSCE430/2021Spring (visited on 03/20/2023).

[37] UT Austin. *CS104c: Competitive Programming*. URL: https://www.cs.utexas.edu/users/downing/cs104c/ (visited on 03/20/2023).

[38] Etienne Vouga. *Etienne Vouga - UT Austin Graphics - Home*. URL: https://www.cs.utexas.edu/users/evouga/ (visited on 03/20/2023).

[39] Wikipedia contributors. *Competitive programming — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Competitive_programming. [Online; accessed 24-May-2022]. 2022.

[40] YouTube. *COMP300E - Programming Challenges - 2009 HKUST*. URL: https://www.youtube.com/playlist?list=PL07B3F10B48592010 (visited on 03/20/2023).