Puzzling Problems in Computer Engineering

Behrooz Parhami, *University of California, Santa Barbara*

University faculty have designed an engaging, puzzle-based freshman seminar intended to motivate and retain computer engineering students.

Researchers have written a great deal about the shortage of a skilled information technology workforce and the concomitant need to attract more students to computer science and engineering programs.[1] Explanations of this shortage range from the perceived impact of the dot-com demise to company consolidations and off-shoring. An anticipated wave of retirements might worsen this already dire situation in the US.[2]

Attracting students to computer science and engineering programs, while necessary and helpful, tackles only one aspect of the problem. Recruitment efforts must be augmented with strategies for retaining and motivating students after they have enrolled in computer engineering educational programs. Unfortunately, explicit attention to student retention and motivation has been lacking in the curricula recommendations by the IEEE Computer Society and the ACM.[3,4]

Because of the need to master foundational and basic-science notions before dealing with their real-world applications, college students in engineering and technology usually do not come in full contact with their chosen discipline until their third year of studies. Many of these students excel in mathematics, physics, and other such subjects but are averse to studying them in isolation; after all, they could have chosen a basic science major if they were so inclined. Consequently, engineering freshmen simply see their early college experience in the same vein as their high school experience—their coursework does not provide a context that relates mathematics, physics, or even programming courses to practical problems in their field of study.

To address this issue, the University of California, Santa Barbara (UCSB), Computer Engineering Program faculty have created a freshman seminar titled "Ten Puzzling Problems in Computer Engineering" (www.ece.ucsb.edu/~parhami/ece_001.htm).[5] This seminar, required for computer engineering majors, introduces students to some of the most challenging problems computer engineers encounter in their daily professional endeavors.

A NEW KIND OF FRESHMAN SEMINAR

Ten Puzzling Problems in Computer Engineering, a one-unit pass/fail course, augments a host of existing and proposed informal gateway courses that aim to

- help with the transition from high school to college;
- create excitement by allowing the students to work in a small setting with a professor and fellow students on topics of special interest;
- introduce challenges of college-level learning, resources available to students, and key study skills; and
- allow students to sample milestones, achievements, and societal impacts of a particular field, serving in effect as an overview of the discipline.

Published by the IEEE Computer Society

The catalog description of the freshman seminar reads: "Gaining familiarity with, and motivation to study, the field of computer engineering, through puzzle-like problems that represent a range of challenges facing computer engineers in their daily problem-solving efforts and at the frontiers of research."

During the first 15-20 minutes of each weekly one-hour class session, the instructor introduces the students to a puzzle and invites them to participate in formulating solutions by unveiling simple forms of the puzzle, followed by more elaborate versions with appropriate hints. The instructor then devotes the middle segment of each session to explaining background, historical context, and variations on the puzzle; why it is deemed interesting; and general solution methods. In the final third of each class period, the instructor demonstrates how the puzzle and its solution strategies relate to everyday technical challenges in computer engineering.

PUZZLES AS PEDAGOGICAL TOOLS

The use of puzzles as pedagogical tools to enhance learning is not new. Mathematicians have been using puzzles in teaching for a long time.[6] Puzzles also serve as teaching tools in other subjects: algorithms,[7] operations research,[8] and biology,[9] to name a few. However, in all such instances, puzzles are used to keep students interested and engaged in standard courses, rather than as a basis for a motivating gateway course.

Two striking examples capture how puzzles can play a role in computer science and engineering education. Lauren Aaronson demonstrates the relationship between Sudoku puzzles and the NP-complete class of computational problems that arise in important application domains such as scheduling, circuit testing, and gene sequencing.[10] Brian Hayes[11] deals with mechanisms for rearranging train cars in railroad yards, showing how these mechanisms relate to stacks, queues, and other data structures and associated algorithms. These two sources are the basis for two of the 10 lectures in the course: task scheduling and sorting networks.

COURSE STRUCTURE AND CONTENT

Faculty did not conceive this freshman seminar as a forum for imparting knowledge of specific facts or methods (things on which the students can later be tested), nor did they intend it to be a vehicle for honing problem-solving skills.[12] The primary purpose of the course is to motivate and excite.

To this end, faculty decided to forego homework and exams and to assign pass/fail grades based on attendance. The instructor allows one absence (in 10 class sessions) with no consequence to students, while two or three absences require instructor consent for a pass grade, with approval granted based on active class participation or a special oral exam. The intention is to encourage full class participation, while also allowing for some flexibility to accommodate personal circumstances.

Table 1. Discussion topics and the associated puzzles.

Lecture no. and title	Lead puzzle
1. Easy, hard, impossible!	Collatz's conjecture
2. Placement and routing	Houses and utilities
3. Satisfiability	Making change
4. Cryptography	Secret messages
5. Byzantine generals	Liars and truth tellers
6. Binary search	Counterfeit coin
7. Task scheduling	Sudoku
8. String matching	Word search
9. Sorting networks	Rearranging trains
10. Malfunction diagnosis	Logical reasoning

The starting point of the seminar is Collatz's conjecture, as yet neither proved nor disproved. The conjecture states that the sequence of integers $x_1, x_2, x_3, \ldots$—where x_1 is an arbitrary integer and x_{i+1} derives from x_i by halving it for even x_i, or by tripling it and adding 1 for odd x_i—always leads to the repeating pattern 4, 2, 1 in a finite number of steps.

Contrasting Collatz sequences with other sequences that are quite easily analyzed shows that appearances can be deceiving and that the brevity or simplicity of a problem's statement does not always foretell an easy solution. The rest of the course lectures cover both hardware and software topics, in accordance with the modern meaning of "computer engineering," which entails more than mere hardware design.

Table 1 lists the lecture titles and associated lead puzzles. The faculty are developing additional topics to replace some of the current modules or to establish a pool of topics from which the instructor can pick 10 lectures for any given course offering. This would provide added flexibility for instructors, allowing them to tailor the course material to the specific focal points of particular programs.

Course handouts are simple and uncluttered. Each two-sided single-sheet handout contains partial images of four of the lecture slides. The students use the handout as a worksheet during the lecture and can take it away with them to work on the more challenging puzzles outside of class. The handouts and complete course presentations (in PowerPoint and PDF formats) are available at the course's website (www.ece.ucsb.edu/~parhami/ece_001.htm).

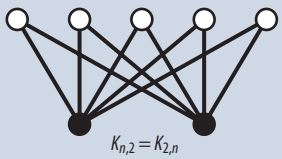

Figure 1. The houses-and-utilities puzzle. A complete bipartite graph models this version with five houses and two utilities, or two houses and five utilities.

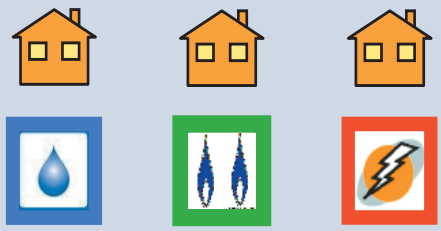

Figure 2. The classic version of the houses-and-utilities puzzle. A few minutes of experimentation reveals that this seemingly simple problem has no solution.

Clue: Z stands for E

```
"CEBA YUC YXSENM PDZ
SERSESYZ, YXZ QESOZDMZ PEJ
XQKPE MYQGSJSYA, PEJ S'K
ECY MQDZ PLCQY YXZ RCDKZD."
          PBLZDY ZSEMYZSE
```

Figure 3. Students are challenged to decipher a quotation like this one from a famous scientist.

Sample lecture: From houses and utilities to multilayer wiring

Imagine two utility facilities and five houses in a flat neighborhood. Can each utility connect to every house in a manner that the 10 lines drawn (trenches dug) do not intersect?

It takes the students only a few minutes to discover that a solution does exist, not only with five houses but with any number of houses that must connect to two utilities. As Figure 1 shows, they also readily determine that the puzzle is wholly symmetrical with regard to the number of houses and utilities.

Next, as Figure 2 shows, the instructor presents the classic form of the puzzle with three houses and three utilities. A few minutes of experimentation convinces the students that this seemingly simple problem has no solution, although they might not be able to explain why; they just know that they always get stuck when trying to draw the last line.

The lecture continues with a simple and elegant proof, by contradiction, for the nonplanarity of $K_{3,3}$. Euler's formula $v - e + f = 2$, which relates the number of vertices, edges, and faces in any planar graph, implies that if $K_{3,3}$ (with $v = 6$ and $e = 9$) was planar, it would have five faces. However, each face in a bipartite graph needs at least four edges, thus implying that there must be at least $20/2 = 10$ edges, after accounting for the sharing of one edge by two neighboring faces.

In the last third of the lecture, the instructor likens utility links to circuits, offering the observation that circuits deposited or "printed" on a surface require nonintersecting connections. If the six-vertex graph $K_{3,3}$ is nonplanar, then there is little hope of arranging the extremely dense electronic circuits in a pattern with nonintersecting lines. This leads to a discussion of multilayer printed circuits (as in a two-sided printed-circuit board). Students then view several drawings and photographs of multiple metal layers in microchips and printed-circuit boards.

Sample lecture: Ancient ciphers and modern cryptography

The simplest codes for secret communication—substitution ciphers—provide a good source of engaging puzzles. Students participate in decoding a few simple substitution ciphers, such as a quotation from a famous scientist like the one shown in Figure 3. They then view other encoding schemes, including key-based ciphers like the one shown in Figure 4, before being told of their weaknesses and vulnerabilities.

Though hand-decoding of such ciphers could take hours or prove impossible in some cases, programmed solution methods, using a combination of exhaustive search and statistical analyses based on the frequencies of letters, letter pairs, triplets, and so on, make decoding a rather easy task. This limits the practical value of substitution and simple key-based ciphers for secure communication.

The instructor presents a brief history of secret communication schemes and associated devices including pictures of a few interesting cipher machines and their principles of operation. Examples range from the ancient cylindrical device composed of rotating wheels shown in Figure 5 to the German Enigma, a sophisticated code that a team of British mathematicians led by Alan Turing deciphered.

The last third of the lecture focuses on modern key-based ciphers and their encoding and decoding algorithms, including those of the data encryption standard. Such ciphers are less vulnerable to statistical attacks, especially if the key is fairly long. However, the need for regular change in keys (for the same reason as password changes) creates the burden of key interchange among communicating parties and introduces the possibility of a breach of security during exchange. This has led to public-

key cryptography. The lecture concludes by showing how public-key cryptography works and how it enables the use of electronic signatures for authentication.

During its first two offerings in spring 2007 and 2008, the freshman seminar had 41 and 36 formally enrolled students, respectively, with five to 10 additional students attending each class session. The students were encouraged to participate in class discussions and to communicate with the instructor outside the classroom during open office hours and via e-mail. Classroom participation was quite good, much better than typically expected of freshmen.

Written student comments at the end of each of the two quarters were overwhelmingly positive. Many students indicated that they enjoyed the course, particularly because it entailed no homework pressure or exam preparation. Overall, the puzzle-based freshman seminar was successful, and the Computer Engineering Program is planning its continuation.

Faculty will conduct a formal evaluation along with its outcomes and curricular impacts after we offer the course a few more times and early enrollees pass through upper-division courses in their major. However, we are quite satisfied with the side effects of the course that are already in evidence: increased student face time with faculty and higher levels of advice-seeking about academic problems and career opportunities.

Ten Puzzling Problems in Computer Engineering is but one tool in the quest to raise student morale and interest. The UCSB computer engineering faculty will actively pursue other methods to supplement and strengthen this unique tool. Key among these additional methods are small-scale design seminars and undergraduate research projects.

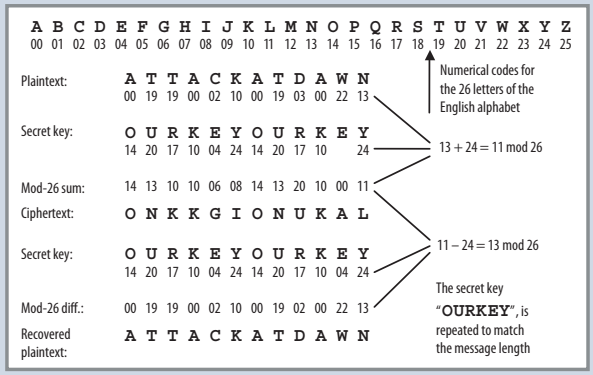

Figure 4. Key-based ciphers are harder to crack than substitution ciphers.

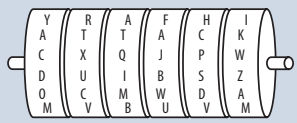

Figure 5. Ancient cipher device. The encrypter rotates the wheels until the desired message appears on a particular row. Letters from a different row form the enciphered message.

References

1. A.Y. Akbulut and C.A. Looney, "Inspiring Students to Pursue Computing Degrees," *Comm. ACM*, vol. 50, no. 10, 2007, pp. 67-71.
2. National Association of State Chief Information Officers, "State IT Workforce: Here Today, Gone Tomorrow?" A National Survey of the States, NASCIO, Oct. 2007.
3. IEEE CS/ACM Joint Task Force on Computing Curricula, *Computing Curricula 2001: Computer Science*, IEEE CS Press, Apr. 2002.
4. IEEE CS/ACM Joint Task Force on Computing Curricula, *Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, IEEE CS Press, July 2006.
5. B. Parhami, "A Puzzle-Based Seminar for Computer Engineering Freshmen," *Computer Science Education*, vol. 18, no. 4, 2008, pp. 261-277.
6. J. Parker, "The Use of Puzzles in Teaching Mathematics," *Mathematics Teacher*, Apr. 1955, pp. 218-227.
7. A. Levitin and M.A. Papalaskari, "Using Puzzles in Teaching Algorithms," *Proc. ACM SIGCSE Conf. Computer Science Education*, ACM Press, 2002, pp. 292-296.
8. H. Müller-Merbach, "The Role of Puzzles in Teaching Combinatorial Programming," *Combinatorial Programming: Methods and Applications*, B. Roy, ed., Springer, 1975, pp. 379-386.
9. S. Franklin, M. Peat, and A. Lewis, "Non-Traditional Interventions to Stimulate Discussion: The Use of Games and Puzzles," *J. Biological Education*, vol. 37, no. 2, 2003, pp. 79-84.
10. L. Aaronson, "Sudoku Science: A Popular Puzzle Helps Researchers Dig into Deep Math," *IEEE Spectrum*, Feb. 2006, pp. 16-17.
11. B. Hayes, "Trains of Thought: Computing with Locomotives and Box Cars Takes a One-Track Mind," *Am. Scientist*, vol. 95, no. 2, 2007, pp. 108-113.
12. B. Averbach and O. Chein, *Problem Solving Through Recreational Mathematics*, Dover, 2000.

Behrooz Parhami *is a professor of electrical and computer engineering at the University of California, Santa Barbara, where he is engaged in teaching and research on computer arithmetic, parallel processing, and dependable computing. He is a Fellow of both the IEEE and the British Computer Society and a professional member of the ACM. Contact him at parhami@ece.ucsb.edu.*