Historical Projects in Computer Science Education

Inna V Pivkina*, Janet Barnett***, Guram Bezhanishvili**, Hing Leung*, Jerry Lodder**, David Pengelley**, Desh Ranjan*

*Computer Science Department, New Mexico State University**

*Mathematical Science Department , New Mexico State University***

*Mathematics and Physics Department, Colorado State University - Pueblo****

To become a tester please visit
http://www.cs.nmsu.edu/historical-projects

Abstract:

Several classroom projects for use in beginning to advanced undergraduate courses in discrete mathematics, algorithm design, automata, graph theory, and logic have been developed. Designed to capture the spark of discovery and motivate the subsequent lines of inquiry, the projects are rooted in primary historical sources, close to the initial solution of problems that would eventually find resolution in modern concepts such as induction, recursion, or algorithm. Classroom testing of the projects shows improved student attitude and improved performance after completing a course with historical projects. The materials are available at http://www.cs.nmsu.edu/historical-projects

Why historical sources?

•Historical point of view provides context and motivation

•Rediscovery of the conceptual roots CS shares with discrete math

•Connect CS to a longer history

•Develop skills of moving from verbal descriptions to precise formulations

•Seeing how problems which used to be hard can be solved easily today, stimulates interest and appreciation for science

Student reactions:

Students who completed a course with projects performed better in subsequent courses. They scores better than the mean GPA in 64% (62%) of the cases for Math (CS).

Quotes from students:
On the benefits of learning from historical sources:

"*context provides a hook to hang the information on*"
"*helps to understand where other mathematic concepts come from*"
"*learn by example*"
"*I think it makes it a little more interesting*"
"*understand how to reinvent certain ideas when necessary*"
"*to know how easy it is now*"

About the use of historical projects:

"it really expands one's horizons and you come away more knowledgeable and more informed"

One of the projects:

Counting Triangulations of a Polygon

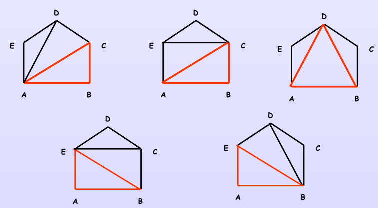
A brief history of the problem:

In a 1751 letter Leonard Euler talks about the problem of counting the number of triangulations of a convex polygon. He provides a "guessed" method for computing the number of triangulations but does not provide a proof…
The problem was posed as an open challenge to the mathematicians by Joseph Liouville in the late 1830's.
He received many responses, including a solution by Belgian mathematician Catalan which was correct but not so elegant.
The most elegant solution was in a paper by Gabriel Lamé in 1838.

Primary Historical source:

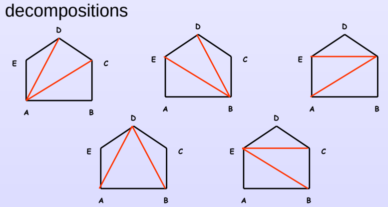
Excerpt from a letter of Monsieur Lamé to Monsieur Liouville on the question: Given a convex polygon, in how many ways one can partition it into triangles by means of diagonals?

I. Side AB serves as the base of a triangle



$$P_{n+1} = P_n + P_3P_{n-1} + P_4P_{n-2} + P_5P_{n-3} + \ldots + P_{n-3}P_5 + P_{n-2}P_4 + P_{n-1}P_3 + P_n$$

II. (n-3) diagonals from a vertex define a group of decompositions



$$P_n = n\,(P_3P_{n-1} + P_4P_{n-2} + \ldots + P_{n-2}P_4 + P_{n-1}P_3) \,/\, (2n - 6)$$

Formula III. $P_{n+1} = (4n-6)\,P_n \,/\, n$

Sample tasks:

Draw a triangulation of a hexagon
How many different possible diagonals does an n sided polygon have?
Prove by mathematical induction: Any triangulation of an n sided polygon has n-2 triangles and n-3 diagonals
Explain in your own words how to derive the formula
Show all the algebraic steps to derive Lame's formula III using formulas I and II
Write a simple recursive function in Java to calculate P_n using the formula
Restrict the total time your program uses to 10 min. What is the largest value of i for which Pi was computed?
Print out a table of values of i and the time required in seconds to compute Pi
Graph and analyze the tables
Write a program to compute P_i for i=3, 4, 5, … using a) Brute force approach b) Dynamic programming approach

List of Projects

Available:

In the Words of Archimedes
In the Words of Archimedes II
In the Words of Pascal
Are All Infinities Created Equal?
An Introduction to Turing Machines
Turing Machines, Induction and Recursion
Turing Machines and Binary Addition
The Universal Computing Machine
The Decision Problem
Binary Arithmetic: From Leibniz to von Neumann
Arithmetic Backwards from von Neumann to the Chinese Abacus
Treatise on the Arithmetical Triangle
Counting Triangulations of a Polygon (Math Version)
Counting Triangulations of a Polygon (CS Version)
Two-Way Deterministic Finite Automata
Church's Thesis
Euler Circuits and the Königsberg Bridge Problem
Topological Connections from Graph Theory
Hamiltonian Circuits and Icosian Game

Coming soon:

Summation of Numerical Powers
Summation of Powers, Bernoulli Numbers, and the Euler-Maclaurin Summation Formula
Logic and Truth Tables
Boolean Algebra and Discrete Structures
Euclid's GCD Algorithm : Recursion vs. Iteration
Induction and Recursive Thought
A History of Sorting: The Emergence of Quicksort
History of Coding and Huffman Codes
Networks and Spanning Trees
Program Correctness
Arthur Cayley and Group Theory
Regular Languages and Finite Automata
Gödel's Completeness Theorem
Peano Arithmetic
Gödel's Incompleteness Theorems

Supported by grant DUE-0717752 from the Division of Undergraduate Education of the National Science Foundation, 2008-2011. Thank you NSF!