Project Einstein: Abstracts and Paper Presentation

Journal homepage: http://www.projecteinstein.cf
http://www.dlsu.edu.ph/conferences/project-einstein

Project Einstein
ISSN 2467-785X

An Analysis of Programming Languages and Competitive Programming Problems as Bound by the Limits of Non-Quantum Computation

Jonathan Casano of Ateneo de Naga University, Philippines

Abstract

During his miracle year in 1905, Albert Einstein published two papers entitled “On a Heuristic Viewpoint Concerning the Production and Transformation of Light” and “The Photoelectric Effect”. These two publications later on paved new avenues for research in Physics and related fields. One of the many implications of his papers was the realization of the universe in the quantum level - Quantum Mechanics. In the recent years, Computer Science has been wrapping its head around the idea of quantum computers. Historically, a “bit”, or the computer’s fundamental measure of data, could be described in binary; zero (0) or one (1). With the injection of quantum computing principles, the bit could be said to hold either zero or one, at any given time, depending on the bit’s spin. The quantum bit, or “qubit”, being in two states at one time promises faster performance and lower instruction count in carrying out special kinds of computations. However, as far as Computer Science has reached, it hasn’t been able to manufacture a stable quantum computer. The basic reason being the “Observer Effect” which heralds that quantum computations will change behavior when observed. As a result, seeking to find maximum efficiency using a non-quantum computer is a niche for a good pool of researchers and academics. A quick internet search will reveal a saturated collection of algorithms created to amplify efficiency towards solving a problem that could have been otherwise solved in brute-force using a quantum computer. In light of this, the researcher zeroes in on one of the ways Computer Science is attempting to achieve efficiency closest to that of a quantum computer through an analysis of programming languages (C, C++, Java, Python, Ruby, Perl) and competitive programming problems as bound by the limits of non-quantum computation. This includes a report on the limitations of various programming languages used against specific types of computing problems and their pros and cons as a result of constraints enforced by various online programming competition hosts (www.codechef.com, www.hackerearth.com, www.urionlinejudge.com.br, www.codeforces.com). This paper also delves into the best practices per programming language as ranked by run-time (speed), memory (space) and character count (golf). A side-by-side comparison of selected brute force algorithms and their mathematically refactored counterparts are also presented to demonstrate the limits and strengths of the programming languages identified.

Keywords

Algorithms, competitive programming, applied computing