Designing Algorithms with Limited Work Space

Tetsuo Asano

Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
t-asano@jaist.ac.jp

Abstract. Recent progress in computer systems has provided programmers with virtually unlimited amount of work storage for their programs. This leads to space-inefficient programs that use too much storage and become too slow if sufficiently large memory is not available. Thus, I believe that space-efficient algorithms or **memory-constrained algorithms** deserve more attention.

Constant-work-space algorithms have been extensively studied under a different name, log-space algorithms. Input data are given on a read-only array of n elements, each having $O(\log n)$ bits, and work space is limited to $O(\log n)$ bits, in other words, a constant number of pointers and counters, each of $O(\log n)$ bits. This memory constraint in the log-space algorithms may be too severe for practical applications. For problems related to an image with n pixels, for example, it is quite reasonable to use $O(\sqrt{n})$ work space, which amounts to a constant number of rows and columns.

I will start my talk with a simple algorithm for detecting a cycle in a graph using only some constant amount of work space (more exactly, $O(\log n)$ bits in total) and then its applications. Then, I will introduce some paradigms for designing such memory-constrained algorithms and their applications to interesting problems including those in computational geometry and computer vision.

M. Ogihara and J. Tarui (Eds.): TAMC 2011, LNCS 6648, p. 1, 2011.