Fine-grained Algorithms and Complexity

Virginia Vassilevska Williams

MIT EECS, CSAIL, Cambridge, MA, USA
virgi@mit.edu

──── **Abstract** ────

A central goal of algorithmic research is to determine how fast computational problems can be solved in the worst case. Theorems from complexity theory state that there are problems that, on inputs of size n, can be solved in $t(n)$ time but not in $O(t(n)^{1-\varepsilon})$ time for $\varepsilon > 0$. The main challenge is to determine where in this hierarchy various natural and important problems lie. Throughout the years, many ingenious algorithmic techniques have been developed and applied to obtain blazingly fast algorithms for many problems. Nevertheless, for many other central problems, the best known running times are essentially those of their classical algorithms from the 1950s and 1960s.

Unconditional lower bounds seem very difficult to obtain, and so practically all known time lower bounds are conditional. For years, the main tool for proving hardness of computational problems have been NP-hardness reductions, basing hardness on P neq NP. However, when we care about the exact running time (as opposed to merely polynomial vs non-polynomial), NP-hardness is not applicable, especially if the problem is already solvable in polynomial time. In recent years, a new theory has been developed, based on "fine-grained reductions" that focus on exact running times. Mimicking NP-hardness, the approach is to (1) select a key problem X that is conjectured to require essentially $t(n)$ time for some t, and (2) reduce X in a fine-grained way to many important problems. This approach has led to the discovery of many meaningful relationships between problems, and even sometimes to equivalence classes.

The main key problems used to base hardness on have been: the 3SUM problem, the CNF-SAT problem (based on the Strong Exponential Time Hypothesis (SETH)) and the All Pairs Shortest Paths Problem. Research on SETH-based lower bounds has flourished in particular in recent years showing that the classical algorithms are optimal for problems such as Approximate Diameter, Edit Distance, Frechet Distance, Longest Common Subsequence etc.

In this talk I will give an overview of the current progress in this area of study, and will highlight some exciting new developments and their relationship to database theory.