19th Knuth Prize: Call for Nominations

The Donald E. Knuth Prize for outstanding contributions to the foundations of computer science is awarded for major research accomplishments and contributions to the foundations of computer science over an extended period of time. The Prize is awarded annually by the ACM Special Interest Group on Algorithms and Computing Theory (SIGACT) and the IEEE Technical Committee on the Mathematical Foundations of Computing (TCMF).

Nomination Procedure. Anyone in the Theoretical Computer Science community may nominate a candidate. To do so, please send nominations to **knuth.prize.2019@gmail.com** by **February 15, 2019**. The nomination should state the nominee's name, summarize his or her contributions in one or two pages, provide a CV for the nominee or a pointer to the nominees webpage, and give telephone and email contact information for the nominator. Any supporting letters from other members of the community (up to a limit of 5) should be included in the package that the nominator sends to the Committee chair. Supporting letters should contain substantial information not in the nomination. Others may endorse the nomination simply by adding their names to the nomination letter. If you have nominated a candidate in past years, you can re-nominate the candidate by sending a message to that effect to the above address. (You may revise the nominating materials if you so desire).

Criteria for Selection. The winner will be selected by a Prize Committee consisting of six people appointed by the SIGACT and TCMF Chairs, see below for the composition of the committee. All nominations will be considered by the Committee, including those submitted in previous years, but nomination is not a requirement to receive the Prize. Note that the Knuth prize is awarded to a single individual each year. Nominations of groups of researchers will not be considered.

In selecting the Knuth-Prize winner, the Committee will pay particular attention to a *sustained record* of high-impact, seminal contributions to the foundations of computer science. The selection may also be based partly on educational accomplishments and contributions such as fundamental textbooks and high-quality students. The award is not given for service to the theoretical computer science community, but service might be included in the citation for

a winner if appropriate. The current prize committee consists of Avrim Blum (TTIC, Chair), Alan Frieze (CMU), Shafi Goldwasser (UC Berkeley), Noam Nisan (Hebrew U.), Ronitt Rubinfeld (MIT and Tel Aviv U.), and Andy Yao (Tsinghua U.).

About the Award. The first Knuth Prize was presented at the 1996 ACM Symposium on Theory of Computing (STOC). Prize presentations now alternate between STOC in odd years and the IEEE Symposium on Foundations of Computer Science (FOCS) in even years. The winner is invited to give a lecture at the conference.

The previous Knuth Prize Laureates are: Andrew C.-C. Yao (1996), Leslie G. Valiant (1997), László Lovász (1999), Jeffery Ullman (2000), Christos Papadimitriou (2002), Miklos Ajtai (2003), Mihalis Yannakakis (2005), Nancy Lynch (2007), Volker Strassen (2008), David Johnson (2009), Ravindran Kannan (2011), Leonid Levin (2012), Gary Miller (2013), Richard Lipton (2014), László Babai (2015), Noam Nisan (2016), Oded Goldreich (2017), and Johan Håstad (2018).

About Donald E. Knuth. The prize is named in honor and recognition of the extraordinary accomplishments of Prof. Donald Knuth, Emeritus at Stanford University. Prof. Knuth is best known for his ongoing multivolume series, *The Art of Computer Programming*, which has played a critical role in establishing and defining Computer Science as a rigorous, intellectual discipline. Prof. Knuth has also made fundamental contributions to the subfields of analysis of algorithms, compilers, string matching, term-rewriting systems, literate programming, and typography. His TeX and MF systems are widely accepted as standards for electronic typesetting. Prof. Knuth's work is distinguished by its integration of theoretical analyses and practical, real-world concerns. In his work, theory and practice are not separate components of Computer Science but rather inexorably linked branches of the same whole.