Massachusetts Institute of Technology
School of Engineering Faculty Personnel Record

Date: November 16, 2023
Full Name: Charles E. Leiserson
Department: Electrical Engineering and Computer Science

1. Date of Birth

November 10, 1953

2. Citizenship

U.S.A.

3. Education

School	Degree	Date
Yale University	B. S. (*cum laude*)	May 1975
Carnegie-Mellon University	Ph.D.	Dec. 1981

4. Title of Thesis for Most Advanced Degree

Area-Efficient VLSI Computation

5. Principal Fields of Interest

Caching
Compilers and runtime systems
Computer chess
Computer-aided design
Data structures
Design and analysis of algorithms
Digital hardware and computing machinery
Distance education and interaction
Fast artificial intelligence
Leadership skills for engineering and science faculty
Multicore computing
Parallel algorithms, architectures, and languages
Parallel and distributed computing
Parallel computer architecture
Scalable computing systems
Software performance engineering
Supercomputing
Theoretical computer science

6. Non-MIT Experience

Position	Date
Founder, Chairman of the Board, and Chief Technology Officer, Cilk Arts, Burlington, Massachusetts	2006 – 2009
Director of System Architecture, Akamai Technologies, Cambridge, Massachusetts	1999 – 2001
Shaw Visiting Professor, National University of Singapore, Republic of Singapore	1995 – 1996
Network Architect for Connection Machine Model CM-5 Supercomputer, Thinking Machines	1989 – 1990
Programmer, Computervision Corporation, Bedford, Massachusetts	1975 – 1976
Programmer Yale University Department of Computer Science, New Haven, Connecticut	1974
Programmer Assistant, International Computing Centre, United Nations, Geneva, Switzerland	1973

7. History of MIT Appointments

Rank	Date
Edwin Sibley Webster Professor in EECS	2014 – present
Margaret MacVicar Faculty Fellow	2007 – present
Professor of Computer Science and Engineering	1992 – present
Associate Director of CSAIL	2017 – 2020
Chief Operating Officer of CSAIL	2017 – 2020
Associate Professor of Computer Science and Engineering (with tenure)	1988 – 1992
Richard B. Adler Scholar	1991
Associate Professor of Computer Science and Engineering	1984 – 1988
Assistant Professor of Computer Science and Engineering	1981 – 1984

8. Consulting Record

Firm	Role	Date
MIT Lincoln Laboratory	Technical Consultant	2015 – present
Algorand	Technical Consultant	2019 – 2020
Radix Trading	Leadership Consultant	2020
Purdue University	Leadership Workshop	2014 – 2019
Indiana University of Pennsylvania	Leadership Workshop	2017
Masdar Institute, Abu Dhabi	Leadership Workshop	2016
Wellcome Trust/DBT India Alliance	Leadership Workshop	2016
Intel (SSG)	Technical Consultant	2009 – 2013
University of California, Berkeley	Leadership Workshop	2012
Harvard University	Leadership Workshop	2011
Cilk Arts	Founder and CTO	2006 – 2009
RealNetworks	Expert Witness	2003 – 2007
Carnegie Mellon University	Leadership Workshop	2006
National University of Singapore	Leadership Workshop	2005
Etisalat University, UAE	International Advisory Panel	2003 – 2005
National University of Singapore	Adjunct Professor	1996 – 2005
Akamai Technologies	Director of System Architecture	1999 – 2001
EMC^2	Expert Witness	1998 – 1999
Pratt & Whitney	Expert Witness	1997 – 1999
NKK	Technical Consultant	1997
Max Planck Institute für Informatik	Fachbeirat (Visiting Committee)	1992 – 1997
National University of Singapore	Shaw Visiting Professor	1995 – 1996
Thinking Machines	Corporate Fellow	1985 – 1994
Ecole Normale Superiéure de Lyon	Visiting Professor	1993
Wolfsort	Technical Consultant	1991
W. W. Oliver Company	Expert Witness	1987
Harris (GCSD)	Technical Consultant	1986
AT&T Bell Laboratories	Technical Consultant	1983 – 1986
Analog Devices	Technical Consultant	1985 – 1985
Cognition	Technical Consultant	1984 – 1985
Harris (GASD)	Technical Consultant	1983 – 1985
MIT Lincoln Laboratory	Technical Consultant	1982 – 1984
POS Corporation	Technical Consultant	1974 – 1975

9. Department and Institute Committees, Other Assigned Duties

Activity	Date
Faculty Director, DAF-MIT AI Accelerator	2020 – present
Graduate Counselor (EECS)	1996 – present
Group Leader, CSAIL Supertech Research Group	1993 – present
Associate Director of CSAIL	2017 – 2020
Chief Operating Officer of CSAIL	2017 – 2020
CSAIL Cabinet Member	2016 – 2020
CSAIL Pretty Committee (Chair 2015 – 2017)	2015 – 2019
CSAIL Branding Committee	2017 – 2018
UPOP Workshop Engineering Co-Chair	2001 – 2014
OpenCourseWare Underwriting Committee	2010 – 2012
Head of Singapore-MIT Alliance program in Computer Science	2001 – 2005
MIT Commencement Committee	1996 – 2001
EECS Faculty Search Committee	1998 – 1999
EECS Client Building Committee	1997 – 1999
Singapore Engineering Education Assessment Committee	1997 – 1998
MIT Student Workshop (Chair)	1997
Undergraduate Advisor (EECS)	1990 – 1995
MIT/LCS Student Workshop (Chair)	1990 – 1994
MIT Supercomputing Technologies Student Workshop (Chair)	1993
EECS Professional Education Policy Committee	1992 – 1993
VI-A Liaison for IBM Research	1991 – 1993
MIT VLSI & Parallel Systems Student Workshop (Chair)	1992
MIT VLSI & Parallel Systems Student Workshop (Chair)	1991
Area II Committee (EECS)	1981 – 1988
Graduate Counselor (EECS)	1981 – 1988
LCS Executive Committee	1984 – 1987
Graduate Admissions Committee (EECS)	1981 – 1985
LCS Executive Committee	1981 – 1982

10. Government Committees, Professional Service, etc.

Activity	Date
Advisory Board for ACM *Transactions on Parallel Computing*	2020 – present
Executive Director for OpenCilk	2019 – present
ACM SPAA Steering Committee	1989 – present
Associate Editor for ACM *Transactions on Parallel Computing*	2012 – 2020
IEEE Computer Society Taylor Booth Award Committee	2018
Workshop on Mission Statements for EECS Postdocs	2017
"Do It Right, or Do It Fast?" Workshop for StartMIT	2017
IEEE Computer Society Fellows Evaluation Committee	2016
Workshop on Mission Statements for Rising Stars in EECS	2015
Workshop on Mission Statements for Rising Stars in EECS	2013
Workshop on Mission Statements for Rising Stars in EECS	2012
Led the effort to create ACM *Transactions on Parallel Computing*	2012
Presentation to the Computer Science and Telecommunications Board at the National Academies	2007
Organizer for Dagstuhl Seminar 04301 in Cache-Oblivious and Cache-Aware Algorithms	2004
Program Committee for ACM SIGPLAN PPoPP	2003
Advisory Board for *Journal of Parallel and Distributed Computing*	1999
Supercomputing'99 Steering Committee	1999
General Chair for ACM SPAA	1994 – 1997
Guest Editor for *Journal of Computer and Systems Science*	1996
Program Chair for ACM SPAA	1994
Organizing Committee for DIMACS Workshop on Models, Architectures, and Technologies for Parallel Computation	1993
Associate Editor for Springer-Verlag Texts and Monographs in Computer Science	1986 – 1993
Program Committee for Supercomputing'91	1991
Joint DARPA/NSF and ESPRIT Exploratory Workshop on Information Science and Technology	1990
Associate Editor for *Applied Mathematics Letters*	1987 – 1990
Program Committee for IEEE FOCS	1989
Program Committee for ACM SPAA	1989
DARPA/ISTO TeraOps Working Group	1987 – 1989
Associate Editor for *Journal of Parallel and Distributed Computing*	1986 – 1988
ACM Turing Award Committee (Chair 1986)	1983 – 1987
Program Committee for IEEE FOCS	1986
MIT VLSI Conference, Program Committee	1986
NSF *Ad Hoc* Committee on Supercomputing Software	1985
Journal of VLSI and Computer Systems, Editor	1983 – 1985
MIT VLSI Conference, Program Committee	1984
MIT VLSI Conference, Program Committee	1982

11. Awards Received

Award	Date
ACM *Symposium on Parallelism in Algorithms and Architectures* Parallel Computing Award	2024
ACM *Symposium on Parallelism in Algorithms and Architectures* Test of Time Award for "The network architecture of the Connection Machine CM-5"	2023
IEEE *Foundations of Computer Science* (*FOCS*) Test of Time Award for "Cache-oblivious algorithms (1999)"	2019
ACM SIGCOMM Networking Systems Award for the Akamai Content Delivery Network	2018
Burgess and Elizabeth Jamieson Award for Excellence in Teaching from the MIT EECS Department	2017
ACM PPoPP Best Paper Award for "Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation"	2017
National Academy of Engineering	2016
IEEE Fellow	2016
SIAM Fellow	2015
MIT EECS Edwin Sibley Webster Professor	2014
ACM/IEEE Computer Society Ken Kennedy High-Performance Computing Award	2014
IEEE Computer Society Taylor L. Booth Education Award	2014
ACM Paris Kanellakis Theory and Practice Award	2013
AAAS Fellow	2013
ACM SPAA Best Paper Award for "Memory-mapping support for reducer hyperobjects"	2012
ACM SPAA Best Paper Award for "Reducers and other Cilk++ hyperobjects"	2009
ACM SIGPLAN Most Influential 1998 PLDI Paper Award for "The implementation of the Cilk-5 multithreaded language"	2008
Margaret MacVicar Faculty Fellow, MIT	2007
ACM Fellow	2006
IEEE Micro Top Picks for "Unbounded transactional memory"	2006
2nd Place in Dutch Open Computer Chess Championship for Cilkchess	1998
1st Prize in the *International Conference on Functional Programming*'s ICFP Programming Contest	1998
IEEE Computer Society Distinguished Visitor for the Asia-Pacific Region	1996 – 1998
2nd Place in Dutch Open Computer Chess Championship for Cilkchess	1997
Recognition of Service Award by ACM for service as Conference General Chair for SPAA'97	1997
1st Place in Dutch Open Computer Chess Championship for Cilkchess	1996
Recognition of Service Award by ACM for service as Conference General Chair for SPAA'96	1996
2nd Place in ICCA 8th Computer Chess World Championship for ⋆Socrates 2.0	1995
Recognition of Service Award by ACM for service as Conference General Chair for SPAA'95	1995
3rd Place in ACM International Computer Chess Championship for ⋆Socrates	1994
3rd Place in ACM International Computer Chess Championship for StarTech	1993
Richard B. Adler Scholar (6.035 and 6.918), MIT EECS Department	1991

Award	**Date**
Association of American Publishers Best 1990 Professional and Scholarly Book in Computer Science and Data Processing for *Introduction to Algorithms*	1990
IEEE ICPP Most Original Paper Award for "Communication-efficient parallel algorithms for distributed random-access machines"	1986
IEEE ICPP Best Presentation Award for "A hyperconcentrator switch for routing bit-serial messages"	1986
IEEE ICPP Best Presentation Award for "Fat-trees: universal networks for hardware-efficient supercomputing"	1985
NSF Presidential Young Investigator Award	1985
ACM Doctoral Dissertation Award for *Area-Efficient VLSI Computation*	1982
Fannie and John Hertz Foundation Doctoral Thesis Award for *Area-Efficient VLSI Computation*	1982
Fannie and John Hertz Foundation Fellowship	1977
Yale University Benjamin F. Barge Prize in Mathematics	1972

12. Organization Membership

AAAS (Fellow)
ACM (Fellow)
IEEE (Fellow)
SIAM (Fellow)
National Academy of Engineering

13. Patents and Patent Applications Pending

1. James L. Banal, Mark Bathe, Joseph Berleant, Charles E. Leiserson, and Tao Benjamin Schardl, "Sequence-controlled polymer storage," *United States Patent application* 63/208,973, filed June 9, 2021.

2. Jie Chen, Aldo Pareja, Giacomo Domeniconi, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Timothy Francis Samson Kaler, Tao Benjamin Schardl, and Charles E Leiserson, "Evolving graph convolutional networks for dynamic graphs," *United States Patent application* 6/790682, filed February 13, 2020.

3. Charles E. Leiserson, Kunal Agrawal, Wen-Jing Hsu, and Yuxiong He, "Computing the processor desires of jobs in an adaptively parallel scheduling environment," *United States Patent* 8,510,741, filed March 28, 2007, issued August 13, 2013.

4. Steven C. Miller, Martin M. Deneroff, Curt F. Schimmel, Larry Rudolph, Charles E. Leiserson, Bradley C. Kuszmaul, and Krste Asanovic, "System and method for performing memory operations in a computing system," *United States Patent* 8,321,634, filed April 11, 2011, issued November 27, 2012.

5. Steven C. Miller, Martin M. Deneroff, Curt F. Schimmel, Larry Rudolph, Charles E. Leiserson, Bradley C. Kuszmaul, and Krste Asanovic, "System and method for performing memory operations in a computing system," *United States Patent* 7,925,839, filed July 7, 2008, issued April 12, 2011.

6. Steven C. Miller, Martin M. Deneroff, Curt F. Schimmel, Larry Rudolph, Charles E. Leiserson, Bradley C. Kuszmaul, and Krste Asanovic, "System and method for performing memory operations in a computing system," *United States Patent* 7,398,359, filed April 30, 2004, issued July 8, 2008.

7. Timothy N. Weller and Charles E. Leiserson, "Content delivery network service provider (CDNSP)-managed content delivery network (CDN) for network service provider (NSP)," *United States Patent* 7,376,727, filed December 11, 2006, issued May 20, 2008.

8. Timothy N. Weller and Charles E. Leiserson, "Content delivery network service provider (CDNSP)-managed content delivery network (CDN) for network service provider (NSP)," *United States Patent* 7,149,797, filed April 2, 2002, issued December 12, 2006.

9. Bradley C. Kuszmaul, Charles E. Leiserson, Shaw-Wen Yang, Carl R. Feynman, W. Daniel Hillis, and David C. Douglas, "Digital computer for determining a combined tag value from tag values selectively incremented and decremented reflecting the number of messages transmitted and not received," *United States Patent* 5,680,550, filed February 13, 1995, issued October 21, 1997.

10. W. Daniel Hillis, David C. Douglas, Charles E. Leiserson, Bradley C. Kuszmaul, Mahesh N. Ganmukhi, Jeffrey V. Hill, and Monica C. Wong-Chan, "Parallel computer system with physically separate tree networks for data and control messages," *United States Patent* 5,590,283, filed January 27, 1995, issued December 31, 1996.

11. David C. Douglas, Charles E. Leiserson, Bradley C. Kuszmaul, Shaw-Wen Yang, Daniel W. Hillis, David Wells, Carl R. Feynman, Bruce J. Walker, and Brewster Kahle, "Router for parallel computer including arrangement for redirecting messages," *United States Patent* 5,530,809, filed January 14, 1994, issued June 25, 1996.

12. Bradley C. Kuszmaul, Charles E. Leiserson, Shaw-Wen Yang, Carl R. Feynman, W. Daniel Hillis, David Wells, and Cynthia J. Spiller, "Parallel computer system including arrangement for quickly draining messages from message router," *United States Patent* 5,390,298, filed January 14, 1994, issued February 14, 1995.

13. Charles E. Leiserson, Robert C. Zak, Jr., W. Daniel, Hillis, Bradley C. Kuszmaul, and Jeffrey V. Hill, "Parallel computer system including request distribution network for distributing processing requests to selected sets of processors in parallel," *United States Patent* 5,388,214, filed January 14, 1994, issued February 7, 1995.

14. Robert C. Zak, Charles E. Leiserson, Bradley C. Kuszmaul, Shaw-Wen Yang, W. Daniel Hillis, David C. Douglas, and David Potter, "Parallel computer system including arrangement for transferring messages from a source processor to selected ones of a plurality of destination processors and combining responses," *United States Patent* 5,265,207, filed April 8, 1993, issued November 23, 1993.

15. David C. Douglas, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Charles E. Leiserson, David S. Wells, Monica C. Wong, Shaw-Wen Yang, and Robert C. Zak, Jr., "Parallel computer system," *United States Patent* 5,333,268, filed September 16, 1992, issued July 26, 1994.

16. Thomas H. Cormen and Charles E. Leiserson, "Message merging device," *United States Patent* 4,922,246, filed November 25, 1986, issued May 1, 1990.

17. H. T. Kung and Charles E. Leiserson, "Systolic array apparatuses for matrix computations," *United States Patent* 4,493,048, filed May 16, 1983, issued January 8, 1985.

14. Teaching Experience of Charles E. Leiserson

Term	Subject	Title	Role
ST23	6.506	*Algorithm Engineering*	Lectures, in charge
FT22	6.106	*Performance Engineering of Software Systems*	Lectures, in charge
SS22	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST22	6.009	*Fundamentals of Programming*	Recitation (2 sections)
FT21	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
ST21	6.S083	*Introduction to Computational Thinking*	Lectures
FT20	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
FT19	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS19	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
FT18	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS18	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
FT17	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
FT17	6.871	*Performance Engineering of Software Systems*	Lectures, in charge
SS17	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST17	6.S898	*Adv. Performance Engineering for Multicore Applications*	Lectures, in charge
FT16	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
FT16	6.871	*Performance Engineering of Software Systems*	Lectures, in charge
SS16	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
SS15	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST15	6.886	*Advanced Performance Engineering of Software Systems*	Lectures, in charge
FT14	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS14	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST14	6.172	*Performance Engineering of Software Systems*	Development
FT13	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS13	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST13	6.UAT	*Preparation for Undergraduate Advanced Project*	Recitation (3 sections)
FT12	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS12	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST12	6.172	*Performance Engineering of Software Systems*	Development
FT11	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS11	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST11	6.046	*Design and Analysis of Algorithms*	Lectures, in charge
FT10	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS10	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
ST10	6.884	*Concepts in Multicore Programming*	Lectures, in charge
FT09	6.172	*Performance Engineering of Software Systems*	Lectures, in charge
SS09	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge
SS09	6.02s	*Concepts in Multicore Programming*	Lectures, in charge
FT08	6.197	*Performance Engineering of Software Systems*	Lectures, in charge
SS07	PI.61s	*Leadership Skills for Engineering and Science Faculty*	Facilitator, in charge

Term	Subject	Title	Role
ST06		*Leadership Skills for Engineering Faculty*	Lectures, in charge
FT05	6.046	*Introduction to Algorithms*	Lectures, in charge
ST05	6.046	*Introduction to Algorithms*	Lectures, in charge
FT04	6.046	*Introduction to Algorithms*	Lectures, in charge
ST04	6.895	*Theory of Parallel Hardware*	Lectures, in charge
FT03	6.895	*Theory of Parallel Systems*	Lectures, in charge
ST03	6.042	*Mathematics for Computer Science*	Lectures, in charge
FT02	6.042	*Mathematics for Computer Science*	Lectures, development
ST02	6.033	*Computer System Engineering*	Recitation (2 sections)
FT01	6.046	*Introduction to Algorithms*	Lectures, in charge
FT99	6.046	*Introduction to Algorithms*	Lectures, in charge
ST99	6.046	*Introduction to Algorithms*	Lectures, in charge
ST98	6.046	*Introduction to Algorithms*	Lectures, in charge
FT98	6.972	*The Structure of Engineering Revolutions*	Development, in charge
ST97	6.892	*Theory of Parallel Systems*	Lectures, in charge
FT96	6.046	*Introduction to Algorithms*	Lectures, in charge
ST96	CS413	*Introduction to Parallel Systems* (National University of Singapore)	Lectures, in charge
ST95	6.046	*Introduction to Algorithms*	Lectures, in charge
FT94	6.042	*Mathematics for Computer Science*	Lectures, in charge
ST94	6.042	*Mathematics for Computer Science*	Lectures, in charge
FT93	6.042	*Mathematics for Computer Science*	Development, in charge
SS93	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
ST93	6.851	*Theory of Algorithms*	Lectures, in charge
FT92	6.004	*Computation Structures*	Recitation (1 section)
ST92	6.851	*Theory of Algorithms*	Lectures, in charge
SS91	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
ST91	6.046	*Introduction to Algorithms*	Lectures, in charge
FT90	6.046	*Introduction to Algorithms*	Lectures, in charge
SS90	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
FT89	6.046	*Introduction to Algorithms*	Lectures, in charge
SS89	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
ST89	6.004	*Computation Structures*	Recitation (2 sections)
SS88	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
FT87	6.848	*Introduction to VLSI and Parallel Computation*	Lectures
FT87	6.046	*Introduction to Algorithms*	Lectures, in charge
SS87	6.84s	*Parallel Algorithms and Architectures*	Lectures, in charge
ST87	6.849	*Advanced VLSI and Parallel Computation*	Lectures, in charge
FT86	6.848	*Introduction to VLSI and Parallel Computation*	Lectures
FT86	6.046	*Introduction to Algorithms*	Lectures, in charge
ST86	6.891	*Theory of Computing Machinery*	Lectures, in charge
FT85	6.046	*Introduction to Algorithms*	Lectures, in charge

Term	Subject	Title	Role
FT84	6.001	*Structure and Interpretation of Computer Programs*	Recitation (2 sections)
ST84	6.895	*VLSI Algorithms*	Lectures, in charge
FT83	6.045	*Computability, Automata, and Formal Languages*	Lectures, in charge
ST83	6.045	*Computability, Automata, and Formal Languages*	Lectures, in charge
FT82	6.033	*Computer System Engineering*	Recitation (2 sections)
ST82	6.002	*Circuits and Electronics*	Recitation (2 sections)
FT81	6.032	*Computation Structures*	Recitation (2 sections)
ST81	6.001	*Structure and Interpretation of Computer Programs*	Recitation (2 sections)

Publications of Charles E. Leiserson

15. Books

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, fourth edition, The MIT Press, 2022. (The first three editions sold over one million copies worldwide.)

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, third edition, The MIT Press, 2009.

3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, second edition, The MIT Press and McGraw-Hill, 2001.

4. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Introduction to Algorithms*, The MIT Press and McGraw-Hill, 1990. (Selected by the Association of American Publishers as the Best 1990 Professional and Scholarly Book in Computer Science and Data Processing.)

5. Charles E. Leiserson, editor, *Advanced Research in VLSI*, The MIT Press, Cambridge, Massachusetts, 1986.

6. Charles E. Leiserson, *Area-Efficient VLSI Computation*, ACM Doctoral Dissertation Award Series, The MIT Press, Cambridge, Massachusetts, 1983. (Won the ACM award for best Ph.D. thesis in computer science for the 1981–1982 academic year, as well as the John and Fannie Hertz Foundation award for best Ph.D. thesis in 1981.)

16. Papers in Refereed Journals

1. Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez, and Tao B. Schardl, "There's plenty of room at the Top: What will drive growth in computer performance after Moore's Law ends?" *Science*, Vol. 368, No. 6495, 2020.

2. Tao B. Schardl, William S. Moses, and Charles E. Leiserson, "Tapir: Embedding recursive fork-join parallelism into LLVM's intermediate representation," *ACM Transactions on Parallel Computing*, Vol. 6, No. 4, December 2019, Article 15. Available at `\https://doi-org.libproxy.mit.edu/10.1145/3365655`.

3. Rezaul Chowdhury, Pramod Ganapathi, Stephen Tschudi, Jesmin Jahan Tithi, Charles Bachmeier, Charles E. Leiserson, Armando Solar-Lezama, and Bradley C. Kuszmaul, and Yuan Tang, "Autogen: Automatic discovery of efficient recursive divide-and-conquer algorithms for solving dynamic programming problems," ACM Transactions on Parallel Computing, Vol. 4, No. 1, October 2017, pp. 4:1–4:30. An early version appeared as "AUTOGEN: automatic discovery of cache-oblivious parallel recursive algorithms for solving dynamic programs" in *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, March 2016, pp. 10:1–10:12.

4. Ekanathan Palamadai Natarajan, Maryam Mehri Dehnavi, and Charles E. Leiserson, "Auto-tuning divide-and-conquer stencil computations, *Concurrency and Computation: Practice and*

Experience, Vol. 29, No. 17, 2017. Available at `http://onlinelibrary.wiley.com/doi/10.1002/cpe.4127/epdf`.

5. Tim Kaler, William C. Hasenplaugh, Tao B. Schardl, and Charles E. Leiserson, "Executing dynamic data-graph computations deterministically using chromatic scheduling," *ACM Transactions on Parallel Computing*, Vol. 3, No. 1, July 2016, pp. 2:1–2:31. An early version appeared in *ACM Symposium on Parallelism in Algorithms and Architectures*, June 2014, pp. 154–165.

6. Charles E. Leiserson, "A simple deterministic algorithm for guaranteeing the forward progress of transactions," *Information Systems*, Vol. 57, Issue C, April 2016, pp. 69–74.

7. Charles E. Leiserson, Tao B. Schardl, and Warut Suksompong, "Upper bounds on number of steals in rooted trees," *Theory of Computing Systems*, 2016, Vol. 58, Issue 2, pp. 223–240.

8. I-Ting Angelina Lee, Charles E. Leiserson, Tao B. Schardl, Jim Sukha, and Zhunping Zhang, "On-the-fly pipeline parallelism," *ACM Transactions on Parallel Computing*, Vol. 2, No. 3, September 2015, pp. 17:1–17:42. An early version appeared in *ACM Symposium on Parallelism in Algorithms and Architectures*, July 2013, pp. 140–151.

9. Warut Suksompong, Charles E. Leiserson, and Tao B. Schardl, "On the efficiency of localized work stealing," *Information Processing Letters*, Vol. 116, No. 2, February 2016, pp. 100–106.

10. Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran, "Cache-oblivious algorithms," *ACM Transactions on Algorithms*, Vol. 8, No. 1, January 2012, pp. 4:1–4:22.

11. Charles E. Leiserson, "The Cilk++ concurrency platform," *Journal of Supercomputing*, Vol. 51, No. 3, March 2010, pp. 244–257.

12. Kunal Agrawal, Yuxiong He, Wen Jing Hsu, and Charles E. Leiserson, "Adaptive scheduling with parallelism feedback," *ACM Transactions on Computing Systems*, Vol. 16, No. 3, September 2008, pp. 7:1–7:32. An early version appeared in the *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, March 2006, pp. 100–109.

13. Yuxiong He, Wen Jing Hsu, and Charles E. Leiserson, "Provably efficient online nonclairvoyant adaptive scheduling," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 9, Feb. 2008, pp. 1263–1279. An early version appeared in *2007 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Long Beach, CA, March 2007, pp. 1–10.

14. John S. Danaher, I-Ting Angelina Lee, and Charles E. Leiserson, "Programming with exceptions in JCilk," *Science of Computer Programming*, Vol. 63, No. 2, December 2006, pp. 147–171.

15. C. Scott Ananian, Krste Asanovic, Bradley C. Kuszmaul, Charles E. Leiserson, Sean Lie, "Unbounded transactional memory," *IEEE Micro*, Vol. 26, No. 1, January 2006, pp. 59–69. (Won the *IEEE Micro* "Top Picks" Award for the most industry relevant and significant papers of the year in computer architecture.) An early version appeared in *11th International Symposium on High-Performance Computer Architecture*, San Francisco, CA, February, 2005, pp. 316–327.

16. Don Dailey and Charles E. Leiserson, "Using Cilk to write multiprocessor chess programs," *Advances in Computer Games*, H.J. van den Herik and B. Monien, eds., Vol. 9, University of Maastricht, 2001, pp. 25–52.

17. Guy E. Blelloch, Charles E. Leiserson, Bruce M. Maggs, C. Gregory Plaxton, Stephen J. Smith, and Marco Zagha, "An experimental analysis of parallel sorting algorithms," *Theory of Computing Systems*, Vol. 31, No. 2, 1998, pp. 135–167. An early version appears under the title, "A comparison of sorting algorithms for the Connection Machine CM-2," in the *3rd Annual ACM Symposium on Parallel Algorithms and Architectures*, Hilton Head, South Carolina, July 1991, pp. 3–16.

18. Charles E. Leiserson and Keith H. Randall, "Parallel algorithms for the circuit value update problem," *Theory of Computing Systems*, Vol. 30, 1997, pp. 583–597. An early version appears in the *7th Annual ACM Symposium on Parallel Algorithms and Architectures*, July 1995, pp. 13–20.

19. Alexander T. Ishii, Charles E. Leiserson, and Marios Papaefthymiou, "Optimizing two-phase, level-clocked circuitry," *Journal of the ACM*, Vol. 44, No. 1, January 1997, pp. 148–199. An early version appears in the *Brown/MIT Conference on Advanced Research in VLSI and Parallel Systems*, Providence, Rhode Island, March 1992, pp. 245–264.

20. Robert D. Blumofe, Christopher F. Joerg, Bradley C. Kuszmaul, Charles E. Leiserson, Keith H. Randall, and Yuli Zhou, "Cilk: An efficient multithreaded runtime system," *Journal of Parallel and Distributed Computing*, Vol. 37, No. 1, August 1996, pp. 55–69. An early version appears in the *5th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, July 1995, pp. 207–216.

21. Robert D. Blumofe and Charles E. Leiserson, "Scheduling multithreaded computations by work stealing," *Journal of the ACM*, Vol. 46, No. 5, September 1999, pp. 720–748. An early version appears in the *35th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, November 1994, pp. 356–368.

22. Robert D. Blumofe and Charles E. Leiserson, "Space-efficient scheduling of multithreaded computations," *SIAM Journal on Computing*, Vol. 7, No. 1, February 1998, pp. 202–229. An early version appears in *Proceedings of the 25th Symposium on Theory of Computing*, ACM, San Diego, California, May, 1993, pp. 362–371.

23. Charles E. Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong, Shaw-Wen Yang, and Robert Zak, "The network architecture of the Connection Machine CM-5," *Journal of Parallel and Distributed Computing*, Vol. 33, No. 2, March 15, 1996, pp. 145–158. An early version appears in the *4th Annual ACM Symposium on Parallel Algorithms and Architectures*, San Diego, California, July 1992, pp. 272–285. (Won the 2023 Test of Time award from the conference.)

24. Charles E. Leiserson and James B. Saxe, "Retiming synchronous circuitry," *Algorithmica*, Vol. 6, 1991, pp. 5–35.

25. Thomas H. Cormen and Charles E. Leiserson, "A hyperconcentrator switch for routing bit-serial messages," *Journal of Parallel and Distributed Computing*, Vol. 10, 1990, pp. 193–204. An early version appears in *1986 International Conference on Parallel Processing*, St. Charles, Illinois, August 1986, pp. 721–728. (Received Best Presentation Award at the conference.)

26. Joe Kilian, Shlomo Kipnis, and Charles E. Leiserson, "The organization of permutation architectures with bussed interconnections," *IEEE Transactions on Computers*, Vol. 39, No. 11, November 1990, pp. 1346–1358. An early version appears in *28th Annual Symposium on Foundations of Computer Science*, Syracuse, New York, IEEE Computer Society, October 1987, pp. 305–315.

27. Ronald I. Greenberg and Charles E. Leiserson, "A compact layout for the three-dimensional tree of meshes," *Applied Mathematics Letters*, Vol. 1, No. 2, 1988, pp. 171–176.

28. Charles E. Leiserson and Bruce M. Maggs, "Communication-efficient parallel algorithms for distributed random-access machines," *Algorithmica*, Vol. 3, 1988, pp. 53–77. An early version appears as "Communication-efficient parallel graph algorithms," in *1986 International Conference on Parallel Processing*, St. Charles, Illinois, August 1986, pp. 861–868. (Received Most Original Paper Award at the conference.)

29. Charles E. Leiserson and James B. Saxe, "A mixed-integer linear programming problem which is efficiently solvable," *Journal of Algorithms*, Vol. 9, 1988, pp. 114–128. An early version appears in *21st Annual Allerton Conference on Communication, Control, and Computing*, Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign, Allerton House, Monticello, Illinois, October 1983, pp. 204–213.

30. Ronald I. Greenberg and Charles E. Leiserson, "Randomized routing on fat-trees," *Advances in Computing Research*, Vol. 5, JAI Press, Inc., 1989, pp. 345–374. An early version appears in *26th Annual Symposium on Foundations of Computer Science*, Portland, Oregon, IEEE Computer Society, October 1985, pp. 241–249.

31. Charles E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, Vol. C-34, No. 10, October 1985, pp. 892–901. An early version appears in the *1985 International Conference on Parallel Processing*, St. Charles, Illinois, IEEE Computer Society Press, August 1985, pp. 393–402. (Received Best Presentation Award at the conference.)

32. Tom Leighton and Charles E. Leiserson, "Wafer-scale integration of systolic arrays," *IEEE Transactions on Computers*, Vol. C-34, No. 5, May 1985, pp. 448–461. An early version appears in the *23rd Annual Symposium on Foundations of Computer Science*, Chicago, Illinois, IEEE Computer Society, November 1982, pp. 297–311.

33. Benny Chor, Charles E. Leiserson, Ronald L. Rivest, and James Shearer, "An application of number theory to the organization of raster-graphics memory," *JACM*, Vol. 33, No. 1, January

1986, pp. 86–104. An early version by the first three authors appears in the *23rd Annual Symposium on Foundations of Computer Science,* Chicago, Illinois, IEEE Computer Society, November 1982, pp. 92–99.

34. Sandeep N. Bhatt and Charles E. Leiserson, "How to assemble tree machines," *Advances in Computing Research,* Vol. 2, 1984, pp. 95–114. An early version appears in the *14th Annual ACM Symposium on Theory of Computing,* San Francisco, California, ACM Special Interest Group for Automata and Computability Theory, May 1982, pp. 77–84.

35. Charles E. Leiserson and Ron Y. Pinter, "Optimal placement for river routing," *SIAM Journal on Computing,* Vol. 12, No. 3, August 1983, pp. 447–462. An early version appears in *CMU Conference on VLSI Systems and Computations,* Pittsburgh, Pennsylvania, October 1981, pp. 126–142.

36. Charles E. Leiserson and James B. Saxe, "Optimizing synchronous systems," *Journal of VLSI and Computer Systems,* Vol. 1, No. 1, Spring 1983, pp. 41–67. An early version appears in the *22nd Annual Symposium on Foundations of Computer Science,* IEEE Computer Society, November 1981, pp. 23–36.

17. Papers in Proceedings of Refereed Conferences

(*Other than early versions of those above.*)

1. Rocío Carratalá-Sáez, Arturo González-Escribano, Alexandros-Stavros Iliopoulos, Charles E. Leiserson, Charlotte Park, Isabel Rosa, Tao B. Schardl, Yuri Torres, David P. Bunde, "Peachy Parallel Assignments (EduHPC 2022)," *Workshop on Education for High-Performance Computing* (*EduHPC*), November 2022.

2. Tim Kaler, Nickolas Stathas, Anne Ouyang, Alexandros-Stavros Iliopoulos, Tao Schardl, Charles E. Leiserson, and Jie Chen, "Accelerating Training and Inference of Graph Neural Networks with Fast Sampling and Pipelining," *Machine Learning and Systems* (*MLSys*), August 2022.

3. Helen Xu, Sean Fraser, and Charles E. Leiserson, "Multidimensional Included and Excluded Sums," *SIAM Conference on Applied and Computational Discrete Algorithms* (*ACDA21*), July 2021.

4. Tim Kaler, Tao B. Schardl, Brian Xie, Charles E. Leiserson, Jie Chen, Aldo Pareja, and Georgios Kollias, "PARAD: A work-efficient parallel algorithm for reverse-mode automatic differentiation," *SIAM Symposium on Algorithmic Principles of Computer Systems* (*APoCS*), January 2021, pp. 144–158.

5. William Kuszmaul and Charles E. Leiserson, "Floors and Ceilings in Divide-and-Conquer Recurrences," *Symposium on Simplicity in Algorithms* (*SOSA*), January 2021.

6. Sean Fraser, Helen Xu, and Charles E. Leiserson, "Work-efficient parallel algorithms for accurate floating-point prefix sums," *IEEE High Performance Extreme Computing Conference* (*HPEC*), September 2020, pp. 1–7.

7. Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," *AAAI Conference on Artificial Intelligence*, 34(04), April 2020, pp. 5363–5370.

8. Tao B. Schardl, Tyler Denniston, Damon Doucet, Bradley C. Kuszmaul, I-Ting Angelina Lee, and Charles E. Leiserson, "The CSI framework for compiler-inserted program instrumentation," *ACM International Conference on Measurement and Modeling of Computer Systems* (*SIGMETRICS*), Vol. 1, No. 2, December 2017, pp. 43:1–43:25.

9. Tao B. Schardl, William S. Moses, and Charles E. Leiserson, "Tapir: Embedding fork-join parallelism into LLVM's intermediate representation," *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP*), February 2017, pp. 249–265. (Won Best Paper Award.)

10. Tao B. Schardl, Bradley C. Kuszmaul, I-Ting Angelina Lee, William M. Leiserson, and Charles E. Leiserson, "The Cilkprof scalability profiler," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2015, pp. 89–100.

11. William C. Hasenplaugh, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson, "Ordering heuristics for parallel graph coloring," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2014, pp. 166–177.

12. Tim Mattson, David Bader, Jon Berry, Aydın Buluç, Jack Dongarra, Christos Faloutsos, John Gilbert, Joseph Gonzalez, Bruce Hendrickson, Jeremy Kepner, Charles Leiserson, Andrew Lumsdaine, David Padua, Stephen Poole, Steve Reinhardt, Mike Stonebraker, Steve Wallach, and Andrew Yoo, "Standards for graph algorithm primitives," *IEEE High Performance Extreme Computing Conference* (*HPEC*), September 2013.

13. Kunal Agrawal, Jeremy T. Fineman, Jordan Krage, Charles E. Leiserson, and Sivan Toledo, "Cache-conscious scheduling of streaming applications," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2012, pp. 236–245.

14. I-Ting Angelina Lee, Aamir Shafi, and Charles E. Leiserson, "Memory-mapping support for reducer hyperobjects," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2012, pp. 287–297. (Won Best Paper Award.)

15. Charles E. Leiserson, Tao B. Schardl, and Jim Sukha, "Deterministic parallel random-number generation for dynamic-multithreading platforms," *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP*), January 2012, pp. 193–204.

16. Yuan Tang, Rezaul Chowdhury, Bradley C. Kuszmaul, Chi-Keung Luk, and Charles E. Leiserson, "The Pochoir stencil compiler," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2011, pp. 117–128.

17. Yuan Tang, Rezaul Chowdhury, Chi-Keung Luk, and Charles E. Leiserson, "Coding stencil computations using the Pochoir stencil-specification language," *USENIX Workshop on*

Hot Topics in Parallelism, May 2011, available at `http://www.usenix.org/events/hotpar11/poster.html`.

18. I-Ting Angelina Lee, Silas Boyd-Wickizer, Zhiyi Huang, and Charles E. Leiserson, "Using memory mapping to support cactus stacks in work-stealing runtime systems," *International Conference on Parallel Architectures and Compilation Techniques* (*PACT*), September 2010, pp. 411–420.

19. Charles E. Leiserson, Liyun Li, Marc Moreno Maza, and Yuzhen Xie, "Efficient evaluation of large polynomials," *International Congress on Mathematical Software*, September 2010, pp. 342–353.

20. Charles E. Leiserson, Liyun Li, Marc Moreno Maza, and Yuzhen Xie, "Parallel computation of the minimal elements of a poset," *International Workshop on Parallel and Symbolic Computation*, July 2010, pp. 53–62.

21. Charles E. Leiserson and Tao B. Schardl, "A work-efficient parallel breadth-first search algorithm (or how to cope with the nondeterminism of reducers)," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2010, pp. 303–314.

22. Yuxiong He, Charles E. Leiserson, and William M. Leiserson, "The Cilkview scalability analyzer," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2010, pp. 145–156.

23. Kunal Agrawal, Charles E. Leiserson, and Jim Sukha, "Executing task graphs using work stealing," *IEEE International Parallel and Distributed Processing Symposium* (*IPDPS*), April 2010, pp. 1–12.

24. Kunal Agrawal, Charles E. Leiserson, and Jim Sukha, "Helper locks for fork-join parallel programming," *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP*), January 2010, pp. 245–256.

25. Matteo Frigo, Pablo Halpern, Charles E. Leiserson, and Stephen Lewin-Berlin, "Reducers and other Cilk++ hyperobjects," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), August 2009, pp. 79–90. (Won Best Paper Award.)

26. Aydın Buluç, Jeremy T. Fineman, Matteo Frigo, John R. Gilbert, and Charles E. Leiserson, "Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), August 2009, pp. 233–244.

27. Charles E. Leiserson, "The Cilk++ concurrency platform," *Design Automation Conference* (*DAC*), July 2009, pp. 522–527.

28. Edya Ladan-Mozes and Charles E. Leiserson, "A consistency architecture for hierarchical shared caches," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2008, pp. 11–22.

29. Kunal Agrawal, Yuxiong He, and Charles E. Leiserson, "Adaptive work stealing with parallelism feedback," *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP*), March 2007, pp. 112-120.

30. Kunal Agrawal, Charles E. Leiserson, and Jim Sukha, "Memory models for open-nested transactions," *ACM SIGPLAN Workshop on Memory Systems Performance and Correctness*, October 2006.

31. Kunal Agrawal, Yuxiong He, and Charles E. Leiserson, "Work stealing with parallelism feedback," *International Conference on Distributed Computing Systems* (*IPDPS*), July 2006.

32. Kunal Agrawal, Yuxiong He, and Charles E. Leiserson, "An empirical evaluation of work stealing with parallelism feedback," *Proceedings of the International Conference on Distributed Computing Systems* (*ICDCS*), July, 2006.

33. Yuxiong He, Wen-Jing Hsu, and Charles E. Leiserson, "Provably efficient two-level adaptive scheduling," *Workshop on Job Scheduling Strategies for Parallel Processing*, June 2006, pp. 1–32.

34. John S. Danaher, I-Ting Angelina Lee, and Charles E. Leiserson, "The JCilk language for multithreaded computing," *OOPSLA Workshop on Synchronization and Concurrency in Object-Oriented Languages*, October 2005.

35. Michael A. Bender, Martin Farach-Colton, Simai He, Bradley C. Kuszmaul, and Charles E. Leiserson, "Adversarial analyses of window backoff strategies for simple multiple-access channels," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), July 2005, pp. 325–332.

36. Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, and Charles E. Leiserson, "On-the-fly maintenance of series-parallel relationships in fork-join multithreaded programs," *ACM Symposium on Parallelism in Algorithms and Architectures* (*SPAA*), June 2004, pp. 133–144.

37. Ching Law and Charles E. Leiserson, "A new competitive analysis of randomized caching," *International Symposium on Algorithms And Computation*, December 2000, pp. 35–46.

38. Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran, "Cache-oblivious algorithms," *IEEE Symposium on Foundations of Computer Science* (*FOCS*), October 17–19, 1999, pp. 285–297.

39. Guang-Ien Cheng, Mingdong Feng, Charles E. Leiserson, Keith H. Randall, and Andrew F. Stark, "Detecting data races in Cilk programs that use locks," *ACM Symposium on Parallel Algorithms and Architectures* (*SPAA*), June 1998, pp. 298–309.

40. Matteo Frigo, Charles E. Leiserson, and Keith Randall, "The implementation of the Cilk-5 multithreaded language," *ACM SIGPLAN Conference on Programming Language Design and Implementation* (*PLDI*), June 1998, pp. 212–223. (Won the 2008 retrospective award for Most Influential Paper.)

41. Mingdong Feng and Charles E. Leiserson, "Efficient detection of determinacy races in Cilk programs," *ACM Symposium on Parallel Algorithms and Architectures* (*SPAA*), June 1997, pp. 1–11.

42. Charles E. Leiserson, "Programming irregular parallel applications in Cilk," *International Symposium on Solving Irregularly Structured Problems in Parallel* (*IRREGULAR*, Paderborn, Germany, June 1997, Springer-Verlag, pp. 61–71.

43. Robert D. Blumofe, Matteo Frigo, Christopher F. Joerg, Charles E. Leiserson, and Keith H. Randall, "An analysis of dag-consistent distributed shared-memory algorithms," *ACM Symposium on Parallel Algorithms and Architectures* (*SPAA*), June 1996, pp. 297–308.

44. Robert D. Blumofe, Matteo Frigo, Christopher F. Joerg, Charles E. Leiserson, and Keith H. Randall, "Dag-consistent distributed shared memory," *10th International Parallel Processing Symposium* (*IPPS*), IEEE Computer Society, April 1996, pp. 132–141.

45. Charles E. Leiserson, Satish Rao, and Sivan Toledo, "Efficient out-of-core algorithms for linear relaxation using blocking covers," *IEEE Symposium on Foundations of Computer Science* (*FOCS*), November 1993, pp. 704–713.

46. Alexander T. Ishii and Charles E. Leiserson, "A timing analysis of level-clocked circuitry," *MIT Conference on Advanced Research in VLSI*, April 1990, pp. 113–130.

47. Charles E. Leiserson and F. Miller Maley, "Algorithms for routing and testing routability of planar VLSI layouts," *ACM Symposium on Theory of Computing* (*STOC*), May 1985, pp. 69–78.

48. Kyle A. Gallivan and Charles E. Leiserson, "High-performance architectures for adaptive filtering," *SPIE Conference on Real Time Signal Processing*, Vol. 495, No. 7, August 1984, pp. 30–38.

49. Charles E. Leiserson, "Systolic and semisystolic design," *IEEE International Conference on Computer Design* (*ICCD*), October 1983, pp. 627–632.

50. Charles E. Leiserson, Flavio M. Rose, and James B. Saxe, "Optimizing synchronous circuitry by retiming," *Caltech Conference on VLSI*, March 1983, pp. 87–116.

51. Charles E. Leiserson, "Area-efficient graph layouts (for VLSI)," *IEEE Symposium on Foundations of Computer Science* (*FOCS*), October 1980, pp. 270–281.

52. Dan Hoey and Charles E. Leiserson, "A layout for the shuffle-exchange network," *International Conference on Parallel Processing* (*ICPP*), August 1980, pp. 329–336.

53. Charles E. Leiserson, "Systolic priority queues," *Caltech Conference on Very Large Scale Integration*, January 1979, pp. 199–214.

54. H. T. Kung and Charles E. Leiserson, "Systolic arrays (for VLSI)," *SIAM Sparse Matrix Proceedings 1978*, 1979, pp. 256–282.

18. Other Major Publications

1. Charles E. Leiserson, Tao B. Schardl, and Neil C. Thompson, "Can Software Performance Engineering Save Us from the End of Moore's Law?" *IEEE Spectrum*, `https://spectrum.ieee.org/software-engineering-can-save-us-from-the-end-of-moores-law`, June 2020.

2. Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," *2nd KDD Workshop on Anomaly Detection in Finance*, August 2019.

3. Charles E. Leiserson and Lily Fu, "How to determine if leadership, management training could improve lab productivity, morale," *Laboratory Equipment*, January 9, 2019, available at `https://bit.ly/2RGB9UW`.

4. Mark Weber, Jie Chen, Toyotaro Suzumura, Aldo Pareja, Tengfei Ma, Hiroki Kanezashi, Tim Kaler, Charles E. Leiserson, and Tao B. Schardl, "Scalable graph learning for anti-money laundering: a first look," *NIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy,* December 2018.

5. Charles E. Leiserson, "Keynote Talk: The resurgence of software performance engineering," *ACM Symposium on Parallelism in Algorithms and Architectures*, July 2018, p. 53.

6. Tao B. Schardl, I-Ting Angelina Lee, and Charles E. Leiserson, "Brief announcement: Open Cilk," *ACM Symposium on Parallelism in Algorithms and Architectures*, July 2018, pp. 351–353.

7. Charles E. Leiserson, "Leadership Training in Academia," *MIT Faculty Newsletter*, Vol. XXIX, No. 4, March/April 2017.

8. Charles E. Leiserson and Chuck McVinney, "Lifelong learning: Science professors need leadership training," *Nature*, Vol. 523, No. 7560, July 16, 2015, pp. 279–281.

9. Arch D. Robison and Charles E. Leiserson, "Cilk Plus," Chapter 13 of *Programming Models for Parallel Computing*, edited by Pavan Balaji, The MIT Press, 2015, pp. 323–352.

10. Jeremy T. Fineman and Charles E. Leiserson, "Race detectors for Cilk and Cilk++ programs, *Encyclopedia of Parallel Computing*, David Padua, editor, Springer US, 2011, pp. 1706–1719.

11. Charles E. Leiserson, "Cilk," *Encyclopedia of Parallel Computing*, David Padua, editor, Springer US, 2011, pp. 273–288.

12. Charles E. Leiserson and Ilya Mirman, "How to survive the multicore software revolution (or at least survive the hype)," *Journal of Advancing Technology*, Vol. 9, Summer 2009, pp. 42–53.

13. Charles E. Leiserson, "The case for a concurrency platform," *Dr. Dobb's Journal*, Vol. 33, November 2008.

14. Erik D. Demaine, Martin L. Demaine, Alan Edelman, Charles E. Leiserson, and Per-Olof Persson, "Building blocks and excluded sums," SIAM News, Vol. 38, No. 1, January/February 2005.

15. Charles E. Leiserson and Aske Plaat, "Programming parallel applications in Cilk," *SIAM News*, Vol. 31, No. 4, May 1998, pp. 6–7.

16. Charles E. Leiserson, "Timekeeper," *SIGACT News*, Vol. 23, No. 4, ACM Press, Fall 1992, pp. 81–82.

17. Charles E. Leiserson, "VLSI theory and parallel supercomputing," Chapter 2 of *Carnegie Mellon University School of Computer Science 25th Anniversary Symposium*, Richard F. Rashid, ed., Addison-Wesley, 1991, pp. 29–44. An early version appeared in *Decennial Caltech Conference on VLSI*, March 1989, The MIT Press, pp. 5–16.

18. Charles E. Leiserson and John G. Lewis, "Orderings for parallel sparse symmetric factorization," Chapter 5 of *Parallel Processing for Scientific Computing*, Garry Rodrigue, ed., SIAM, 1989.

19. Tom Leighton and Charles E. Leiserson, "A survey of algorithms for integrating wafer-scale systolic arrays," in *Wafer-Scale Integration*, G. Saucier and J. Trilhe, eds., North-Holland, 1986, pp. 177–195.

20. Charles E. Leiserson, Jill P. Mesirov, Lena Nekludova, Stephen M. Omohundro, and John Reif, "Solving sparse linear systems via parallel nested dissection on the Connection Machine," *SIAM 1986 National Meeting*, Boston, Mass., July 1986. Also appears as a Thinking Machines Corporation technical memorandum (unnumbered).

21. H. T. Kung and Charles E. Leiserson, "Algorithms for VLSI processor arrays," Chapter 8.3 of *Introduction to VLSI Systems* by Carver A. Mead and Lynn A. Conway, Addison-Wesley, 1978.

19. Internal Memoranda and Progress Reports

(*Other than early versions of those above.*)

1. Charles E. Leiserson and Harald Prokop, "A minicourse on multithreaded programming," unpublished manuscript, available at `http://supertech.lcs.mit.edu/cilk`, 1999.

2. Charles E. Leiserson, Harald Prokop, and Keith H. Randall, "Using de Bruijn sequences to index a 1 in a computer word," unpublished manuscript, available at `http://supertech.lcs.mit.edu/cilk`, 1998.

3. Supercomputing Technologies Group, MIT Laboratory for Computer Science, *Cilk-5.2 (Beta 1) Reference Manual*, unpublished manuscript, available at `http://supertech.lcs.mit.edu/cilk`, 1998. Earlier versions of this manual are also available.

4. Charles E. Leiserson, editor, "Proceedings of the 1994 MIT Student Workshop on Supercomputing Technologies," MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-622, July 1994.

5. Phillip B. Gibbons, Richard M. Karp, Charles E. Leiserson, Gregory M. Papadopoulos, editors, "DIMACS Workshop on Models, Architectures, and Technologies for Parallel Computation," DIMACS Center for Discrete Mathematics and Theoretical Computer Science, Technical Report 93-87, September, 1993.

6. Charles E. Leiserson, editor, "Proceedings of the 1993 MIT Student Workshop on Supercomputing Technologies," MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-575, August 1993.

7. Charles E. Leiserson, editor, "Proceedings of the 1992 MIT Student Workshop on VLSI and Parallel Systems," MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-546, August 1992.

8. Alexander T. Ishii, Charles E. Leiserson, and Marios C. Papaefthymiou, "An algorithm for the tramp steamer problem based on mean-weight cycles," MIT Laboratory for Computer Science Technical Memorandum MIT/LCS/TM-457, November 1991.

9. Charles E. Leiserson, editor, "Proceedings of the 1991 MIT Student Workshop on VLSI and Parallel Systems," MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-513, August 1991.

10. Tom Leighton, Charles E. Leiserson, and Dina Kravets, "Theory of Parallel and VLSI Computation," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 8, May 1990.

11. Tom Leighton and Charles E. Leiserson, "Advanced Parallel and VLSI Computation," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 7, December 1989.

12. Tom Leighton, Charles E. Leiserson, and Eric Schwabe, "Theory of Parallel and VLSI Computation," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 6, March 1989.

13. C. E. Leiserson, F. T. Leighton, and S. A. Plotkin, editors, "Connection Machine Projects," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 4, January 1989.

14. Tom Leighton, Charles E. Leiserson, Bruce Maggs, Serge Plotkin, and Joel Wein, "Advanced Parallel and VLSI Computation," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 2, March 1988.

15. Tom Leighton, Charles E. Leiserson, Bruce Maggs, Serge Plotkin, and Joel Wein, "Theory of Parallel and VLSI Computation," MIT Laboratory for Computer Science Research Seminar Series Memorandum MIT/LCS/RSS 1, March 1988.

16. Charles E. Leiserson and Cynthia A. Phillips, "Parallel contraction of planar graphs," MIT Laboratory for Computer Science Technical Memorandum MIT/LCS/TM-343, October 1987.

17. Charles E. Leiserson and Cynthia A. Phillips, "A space-efficient algorithm for finding the connected components of rectangles in the plane," MIT Laboratory for Computer Science Technical Memorandum MIT/LCS/TM-323, February 1987.

18. Sandeep N. Bhatt and Charles E. Leiserson, "Minimizing the longest edge in a VLSI layout," MIT VLSI Memo No. 82-86, May 1981.

20. Invited Lectures

After Moore's Law

- Sept. 2022, Computation Foundations of Prosperity Conference, Cambridge, MA

Software Performance after Moore's Law

- Apr. 2021, Institute for Data Science Seminar, New Jersey Institute of Technology, webcast

Technical Leadership in Academia

- Apr. 2021, Brazil Tech Trends, webcast

The Resurgence of Software Performance Engineering

- Mar. 2022, Red Hat Performance Conference keynote, webcast
- Oct. 2019, IEEE Computer Society and Greater Boston Chapter of the ACM, Cambridge, MA
- June 2019, Fast Code seminar, MIT CSAIL, Cambridge, MA
- Apr. 2019, University of Maryland, College Park, MD
- Nov. 2018, Duke University, Durham, NC
- July 2018, ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Vienna, Austria
- June 2018, Two Sigma Investments, New York, NY
- Feb. 2017, University of Texas at Austin, TX
- Feb. 2017, Texas A&M University, College Station, TX
- July 2016, Microsoft Research, Redmond, WA
- Sep. 2015, IEEE High Performance Extreme Computing Conference (HPEC) keynote, Waltham, MA

Systems That Learn to Be Fast

- Oct. 2018, Nokia, webcast
- Sept. 2018, SystemsThatLearn@CSAIL, CSAIL Annual Meeting, MIT, Cambridge, MA

Software Performance after Moore's Law

- Oct. 2018, MIT CSAIL/Epoch Foundation Annual Meeting, MIT, Cambridge, MA
- June 2018, CSAIL Alliance Program, MIT, Cambridge, MA

Parallel Algorithms

- Sept. 2018, Lincoln Laboratory Workshop on Modern Algorithms all-day tutorial, Cambridge, MA

Open Cilk

- Feb.2018, Invited Talk, Emu Technologies, New York, NY

A Simple Deterministic Algorithm for Guaranteeing the Forward Progress of Transactions

- Apr. 2016, Symposium on Programming: Logics, Models, Algorithms and Concurrency, University of Texas at Austin, TX
- June 2015, ACM SIGPLAN Workshop on Transactional Computing, Portland, OR

What the $#@! Is Parallelism? (And Why Should Anyone Care?)

- Sept. 2015, K2I Ken Kennedy Award Lecture, Rice University, Houston, TX
- Sept. 2015, Computer Science Colloquium, Harvard University, Cambridge, MA
- May 2015, Colloquium, Lincoln Laboratory, Lexington, MA
- Mar. 2015, MICDE Seminar, University of Michigan, Ann Arbor, MI
- Nov. 2014, Ken Kennedy Award keynote, Supercomputing, New Orleans, LA

Cilk Multithreaded-Programming Technology

- July 2012, Industrial Light & Magic, San Francisco, CA

The Pochoir Stencil Compiler

- Mar. 2012, University of Texas at Austin, Austin TX
- Oct. 2011, Applied Mathematics Colloquium, MIT, Cambridge, MA
- Apr. 2011, University of Connecticut, Storrs, CT

Cilk Runtime Support for Deterministic Parallel Random-Number Generation

- Mar. 2011, Workshop on Determinism and Correctness in Parallel Programming, Newport Beach, CA

Using Thread-Local Memory Mapping to Support Cactus Stacks in Work-Stealing Runtime Systems

- Mar. 2010, Intel Corporation, Champaign, IL

Nonnumeric Computing: A Cilk Perspective

- Mar. 2010, Intel Workshop on Parallel Algorithms for Nonnumeric Computing, Intel Corporation, Santa Clara, CA

Reducers and Other Cilk++ Hyperobjects

- Mar. 2010, University of Illinois at Urbana-Champaign, Champaign, IL

Cilk++ and Reducers

- Jan. 2010, Indo-US Workshop on Parallelism and the Future of High Performance Computing, Bangalore, India

Cilk++

- Nov. 2009, IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)

The Cilk++ Concurrency Platform

- July 2009, Design Automation Conference Special Session on Multicore Computing and EDA, San Francisco, CA

The Design and Analysis of Multithreaded Algorithms

- July 2009, SIAM Annual Meeting plenary lecture, Denver, CO

Designing Cilk++ Algorithms

- May 2009, American University in Cairo, Cairo, Egypt

The Cilk++ Runtime System

- May 2009, American University in Cairo, Cairo, Egypt

Multicore Programming in Cilk++

- May 2009, American University in Cairo, Cairo, Egypt

Cilk++: Multicore-Enabling Legacy C++ Code

- Sept. 2008, University of California, Berkeley, CA
- Apr. 2008, Carnegie Mellon University, Pittsburgh, PA

Cache-Oblivious Jeopardy

- Aug. 2007, Center for Massive Data Algorithmics, Aarhus University, Aarhus, Denmark

Multithreaded Programming in Cilk

- Nov. 2007, Workshop on Manycore and Multicore Computing at Supercomputing, Reno, NV
- Apr. 2007, Intel Research Laboratory, Berkeley, CA
- June 2006, International Workshop on OpenMP keynote, Reims, France
- Dec. 2005, Intel Programming Systems Conference keynote, Santa Clara, CA

Leadership and Engineering

- Apr. 2006, 50th Anniversary Celebration of Computer Science at Carnegie Mellon University, Pittsburgh, PA

Unbounded Transactional Memory

- Jan. 2006, Intel Corporation, Cambridge, MA
- Oct. 2005, University of Rochester, Rochester, NY
- Sep. 2005, Workshop on High-Performance Embedded Systems, Lexington, MA
- Apr. 2005, Workshop on Transactional Systems, Chicago, IL

MIT.001 Final Exam

- Sep. 2005, CSAIL Student Workshop, Gloucester, MA

Cache-Oblivious Algorithms

- July 2005, Computational Research in Boston, Cambridge, MA
- June 2004, Seminar on Cache-Oblivious and Cache-Aware Algorithms, Dagstuhl, Germany

Programming Shared-Memory Multiprocessors Using the Cilk Multithreaded Language

- Oct. 2004, Reflections Projections, University of Illinois, Urbana-Champaign, IL
- July 2004, Scandinavian Workshop on Algorithm Theory, Copenhagen, Denmark
- Mar. 2003, George Washington University, Washington, DC
- May 1999, Understanding the New World of Information '99, Taipei, Taiwan

- Apr. 1999, Seminar on High-Level Parallel Programming, Dagstuhl, Germany
- Mar. 1999, NTT Corporation, Atsugi, Japan
- Jan. 1999, Workshop on Parallel Computing for Irregular Applications, Orlando, FL
- Dec. 1998, International Conference on High-Performance Computing, Chennai, India
- Nov. 1998, Rice University, Houston, TX
- Oct. 1998, University of Delaware, Wilmington, DL
- Oct. 1998, Stanford University, Stanford, CA
- Sep. 1998, University of California, Berkeley; Berkeley, CA
- Sep. 1998, Intel Corporation, Beaverton, OR
- Sep. 1998, MIT EECS Department Colloquium, Cambridge, MA

Design and Analysis of Algorithms for Shared-Memory Multiprocessors

- Aug. 1999, Workshop on Algorithms and Data Structures, Vancouver, CA
- May 1999, Workshop on Parallel Algorithms, Atlanta, GA

Using Cilk to Write Multiprocessor Chess Programs

- June 1999, International Conference on Computer Chess, Paderborn, Germany

Algorithmic Multithreaded Programing Using Cilk

- Aug. 1998, International Workshop on Languages and Compilers for Parallel Computing, Chapel Hill, NC
- Jan. 1998, Sun Microsystems, Inc., Palo Alto, CA
- Jan. 1998, Silicon Graphics, Inc., Mountain View, CA
- Jan. 1998, NASA Ames Research Center, Moffett Field, CA
- July 1997, National University of Singapore, Singapore

Debugging Multithreaded Programs

- Nov. 1998, University of Texas at Austin, Austin, TX
- Sep. 1998, Microsoft Research, Redmond, WA

Teaching Parallel Algorithms using the Cilk Multithreaded Programming Language

- July 1998, Academia Sinica, Taipei, Taiwan
- July 1998, National University of Singapore, Singapore
- June 1998, Yale Workshop on Multithreaded Algorithms, New Haven, CT
- June 1997, Forum on Parallel Computing Curricula, Newport, RI

Efficient Detection of Determinacy Races in Cilk Programs

- Jan. 1998, DEC Systems Research Center, Palo Alto, CA
- Nov. 1997, Distinguished Lecture, University of California, Santa Barbara
- June 1997, Max Planck Institut für Informatik, Saarbrücken, Germany
- Jan. 1997, Dartmouth College, Hanover, New Hampshire
- Jan. 1997, National University of Singapore, Singapore

Algorithmic Multithreaded Computing

- Nov. 1996, University of Maryland, College Park, MD
- Nov. 1996, Theory of Computation Seminar, MIT, Cambridge, MA

- Sep. 1996, Courant Institute, NYU, New York, New York
- June 1996, International Conference on Algorithms and Architectures for Parallel Processing keynote, Singapore

Can Multithreaded Programming Save Massively Parallel Computing?

- May 1996, GINTIC Institute of Manufacturing Technology, Singapore
- Apr. 1996, International Parallel Processing Symposium keynote, Honolulu, Hawaii

What Is Theoretical Computer Science?

- Mar. 1996, Science Research Seminar keynote, National University of Singapore, Singapore

Efficient Scheduling of Multithreaded Computations

- Sep. 1995, Workshop on High-Performance Computing Activities in Singapore, National Supercomputing Research Centre, Singapore
- Apr. 1995, Harvard University, Cambridge, Massachusetts
- Apr. 1995, Carnegie Mellon University, Pittsburgh, Pennsylvania
- Mar. 1995, National University of Singapore, Singapore

Space-Efficient Scheduling of Multithreaded Computations

- Sep. 1994, Carleton University, Ottawa, Canada
- Apr. 1994, Columbia University Theory Day, New York, New York
- Dec. 1993, Université de Paris-Sud, Paris, France
- Sep. 1993, DIMACS Workshop on Models, Architectures, and Technologies for Parallel Computation, Rutgers University, New Jersey

The Network Architecture of the Connection Machine CM-5

- Sep. 1993, Max Planck Institut für Informatik, Saarbrücken, Germany
- Sep. 1993, University of Zurich, Zurich, Switzerland
- July 1993, International Workshop on Interconnection Networks, Marseille, France
- Feb. 1993, Stanford University, Stanford, California
- Dec. 1992, University of Massachusetts, Amherst, Massachusetts
- Dec. 1992, Princeton University, Princeton, New Jersey
- Dec. 1992, IEEE Symposium on Parallel and Distributed Processing keynote, Dallas, Texas
- Nov. 1992, International Heinz Nixdorf Symposium on Parallel Architectures and Their Efficient Use, Paderborn, Germany
- Oct. 1992, IEEE Foundations of Computer Science Conference, Pittsburgh, Pennsylvania
- Oct. 1992, Symposium on New Directions in Parallel and Concurrent Computing, New York, New York
- Sep. 1992, DARPA Joint Microsystems/Computer Systems/HPC Software PI Meeting, Daytona, Florida
- Sep. 1992, Thinking Machines Corporation, Cambridge, Massachusetts
- Sep. 1992, Commisariat à l'Energie Atomique, Saclay, France
- Sep. 1992, International Conference on Parallel Processing, Lyon, France
- June 1992, Dartmouth Institute for Advanced Graduate Studies, Hanover, New Hampshire
- May 1992, University of Washington, Seattle, Washington

- Apr. 1992, MIT, Cambridge, Massachusetts
- Apr. 1992, Carnegie Mellon University, Pittsburgh, Pennsylvania
- Nov. 1991, Sandia National Laboratory, Albuquerque, New Mexico
- Nov. 1991, Yale University, New Haven, Connecticut

How to Interconnect One Million Processors (panel session)

- Oct. 1992, Frontiers of Massively Parallel Computation, McLean, Virginia

Special-Purpose vs. General-Purpose Parallel Computing Networks

- Aug. 1992, International Conference on Application-Specific Array Processors keynote, San Francisco, California

VLSI Theory and Parallel Supercomputing

- Apr. 1992, AT&T Bell Laboratories, Holmdel, New Jersey
- Dec. 1990, NEC Research Institute, Princeton, New Jersey
- Oct. 1990, Texas Instruments Corporation, Dallas, Texas
- Sep. 1990, School of Computer Science 25th Anniversary Symposium, Carnegie Mellon University, Pittsburgh, Pennsylvania
- Apr. 1989, Thinking Machines Corporation, Cambridge, Massachusetts
- Mar. 1989, Decennial Caltech Conference on VLSI, Pasadena, California

A Comparison of Sorting Algorithms for the Connection Machine CM-2

- Nov. 1991, Yale University, New Haven, Connecticut
- Oct. 1991, University of Texas, Austin, Texas
- Mar. 1991, Indiana University, Bloomington, Indiana
- Mar. 1991, Purdue University, West Lafayette, Indiana

A Menagerie of Parallel Computing Networks

- June 1991, MIT Technology Day, Cambridge, Massachusetts

Engineering Parallel Algorithms

- May 1991, Workshop on Parallel Algorithms, New Orleans, Louisiana

Highly Reliable Large-Scale Computing

- Nov. 1990, MIT VLSI Research Review, Cambridge, Massachusetts

Very Large Scale Computing

- Oct. 1988, Project MAC 25th Anniversary Symposium, MIT, Cambridge, Massachusetts

New Machine Models for Synchronous Parallel Algorithms

- Dec. 1987, Institute for Mathematics and Its Applications, , , University of Minnesota, Minneapolis, Minnesota

The Relevance of VLSI Theory to Parallel Supercomputing

- Nov. 1987, Siemens-MIT Conference, Munich, West Germany
- June 1986, SIAM Annual Conference, Boston, Massachusetts
- May 1986, Mathematical Sciences Research Institute, Berkeley, California

- Jan. 1986, Microelectronics and Computer Technology Corporation Workshop on Interconnection Networks, Austin, Texas

Communication-Efficient Parallel Graph Algorithms

- Oct. 1986, Graph Theory Day (New York Academy of Sciences), Albany, New York

Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing

- Apr. 1985, Thinking Machines Corporation, Cambridge, Massachusetts
- Apr. 1985, New York University, New York, New York
- Mar. 1985, Cornell University, Ithaca, New York
- Feb. 1985, University of Toronto, Toronto, Canada
- Feb. 1985, University of Minnesota, Minneapolis, Minnesota
- Jan. 1985, Lawrence Livermore National Laboratory, Livermore, California
- Jan. 1985, University of California at Berkeley, Berkeley, California
- Jan. 1985, Stanford University, Stanford, California
- Jan. 1985, Bolt, Baranek, and Newman, Inc., Cambridge, Massachusetts
- Oct. 1984, IBM Research, Yorktown Heights, New York
- June 1984, AT&T Bell Laboratories, Murray Hill, New Jersey
- June 1984, MIT, Cambridge, Massachusetts

Optimization of Digital Circuitry by Retiming

- Apr. 1984, Brown University, Providence, Rhode Island
- Jan. 1983, Bell Laboratories, Murray Hill, New Jersey
- Nov. 1982, University of Rochester, Rochester, New York
- Oct. 1982, Duke University, Durham, North Carolina

Systolic and Semisystolic Design

- Aug. 1983, Princeton University, Princeton, New Jersey
- Aug. 1983, AT&T Bell Laboratories, Murray Hill, New Jersey

Systolic and Semisystolic Systems

- July 1983, Harris Corporation, Melbourne, Florida
- July 1983, MIT, Cambridge, Massachusetts

Wafer-Scale Integration of Systolic Arrays

- July 1983, MIT, Cambridge, Massachusetts
- Mar. 1983, University of California at Berkeley, Berkeley, California
- Nov. 1982, Carnegie-Mellon University, Pittsburgh, Pennsylvania

Digital Circuit Optimization

- May 1982, MIT, Cambridge, Massachusetts

Optimal Placement for River Routing

- Dec. 1981, MIT, Cambridge, Massachusetts

Optimizing Synchronous Systems

- Dec. 1981, MIT Lincoln Laboratory, Lexington, Massachusetts

- Sep. 1981, Harvard University, Cambridge, Massachusetts
- July 1981, Digital Equipment Corporation, Maynard, Massachusetts

Systolic Systems

- May 1981, MIT, Cambridge, Massachusetts

Theses Supervised by Charles E. Leiserson

Degree	Total	Completed	In Progress
S.B.	22	22	0
S.M.	20	20	0
M.Eng.	40	40	0
Engineers	2	2	0
Doctoral Supervisor	31	31	0
Doctoral Reader	28	28	0

21. S.B. Theses and Undergraduate Advanced Projects

1. Parker Tew, *Expressive and Scalable Simulation with Simit and GraphLab*, May 2015.
2. Daewook Kim, *Parallel processing for next generation Ultra-HD video compression with H.265/HEVC*, May 2015.
3. Tim Kaler, *Parallel Iterative Graph Computation*, May 2012.
4. Boon Teik Ooi, *An Ordered-Set Reducer*, May 2012.
5. Tao Benjamin Schardl, *A Work-Efficient Parallel Breadth-First Search Algorithm*, May 2009. (Won the Robert M. Fano UROP Award for Outstanding EECS UROP and the Arnold L. Nylander Advanced Undergraduate Project Award.)
6. John Danaher, *A Real-Time Distributed Score-Keeping System*, May 2004.
7. David Venturini, *Chess Endgame Database Compression*, May 1999.
8. Svetoslav Tzvetkov, *Parallel Memory Allocation*, May 1999.
9. Adrian Soviani, *5×5 Magic Squares in Cilk*, May 1999.
10. Arup R. Guha, *Implementation of Band Cholesky Factorization in Cilk*, May 1997.
11. Tzu-Yi Chen, *Efficient Implementation of Out-of-Core Conjugate Gradient Algorithms*, May 1995. (Honorable Mention in the Computer Research Association's Outstanding Undergraduates Competition.)
12. Daniel Schmidt, *An Empirical Comparison of Four Heap Data Structures*, August 1994.
13. Keith Randall, *TIM: A CAD Tool for Designing Two-Phase Level-Clocked Circuitry*, May 1993.

14. David B. Wilson, *Embedding Weak Hypercubes in Strong Hypercubes*, May 1991. (Winner of the William Martin Prize for best undergraduate computer science thesis at MIT.)
15. Michele L. Monclova, *An Experimental Comparison of Red-Black Trees and Skip Lists*, May 1991.
16. Vu Le Phan, *Massively Parallel Solutions to the Sparse Assignment Problem*, May 1991.
17. Marie J. Sullivan, *Parallel Graph Algorithms on the Connection Machine,* February 1987.
18. Bruce M. Maggs, *Computing Minimum Spanning Trees on a Fat-Tree Architecture,* May 1985.
19. Alexander T. Ishii, *A Comparison of Routing Algorithms for Fat-Tree Supercomputers,* May 1985.
20. David J. Jilk, *Methodology for User-Aided Silicon Compilation,* January 1985.
21. Csaba Peter Gabor, *A Comparison of VLSI Designs for Complex Multiplication,* January 1984.
22. David H. Covert, *Graphic Objects That Draw Themselves,* May 1983.

22. S.M. Theses

1. William Kuszmaul, *Fair Buffer Management: Achieving Optimal Backlog in Multiprocessor Cup Games*, May 2020.
2. I-Ting Angelina Lee, *The JCilk Multithreaded Language*, August 2005.
3. Jeremy Fineman, *Provably Good Race Detection That Runs in Parallel*, August 2005.
4. Sridhar Ramachandran, *An Algorithmic Theory of Caching*, February 2000.
5. Harald Prokop, *Cache-Oblivious Algorithms*, May 1999.
6. Bin Song, *Scheduling Adaptively Parallel Jobs*, January 1998.
7. Matteo Frigo, *The Weakest Reasonable Memory Model*, January 1998.
8. Keith Randall, *Virtual Networks: Implementation and Analysis*, May 1993.
9. Robert D. Blumofe, *Efficient Storage Management of Multithreaded Computations*, September 1992.
10. Daniel Tunkelang, *An Aesthetic Layout Algorithm for Undirected Graphs*, August 1992.
11. Michael Ernst, *Serializing Parallel Programs by Removing Redundant Computation*, August 1992.
12. Marios C. Papaefthymiou, *On Retiming Synchronous Circuitry and Mixed-Integer Optimization*, August 1990.
13. Jeffrey A. Fried, *VLSI Processor Design for Communication Networks*, January 1989.
14. James A. Park, *Notes on Searching in Multidimensional Monotone Arrays*, January 1989.

15. Alexander T. Ishii, *A Digital Model for Level-Clocked Circuitry*, January 1989.
16. Bruce M. Maggs, *Communication-Efficient Parallel Graph Algorithms,* August 1986.
17. Thomas H. Cormen, *Concentrator Switches for Routing Messages in Parallel Computers,* August 1986.
18. F. Miller Maley, *Compaction with Automatic Jog Introduction,* May 1986.
19. Cynthia A. Phillips, *Space-Efficient Algorithms for Computational Geometry,* August 1985.
20. Martin I. Eiger, *Analysis of Algorithms to Compute Wirings Within Test Fixtures,* June 1985.

23. M.Eng. Theses

1. Daniel Edelman, *Characterizing the Energy Requirement of Computer Vision*, June 2023.
2. Obada M. Alkhatib, *Sampling Methods for Fast and Versatile GNN Training*, September 2022.
3. Philip Murzynowski, *Optimizing Graph Neural Network Training on Large Graphs in A Distributed Setting*, September 2022.
4. Wanlin Li, *Contention Bounds for Locking Computations*, May 2022.
5. Elizabeth Y. Zou, *Preliminary Investigation of Productivity Tools for Memory Profiling in Parallel Programs*, May 2022.
6. Sai Sameer Pusapaty, *Combining Task Parallelism and Multithreaded Concurrency*, February 2022.
7. Qi Qi, *Cilk Hyperobjects, and Other Topics on Parallel Computing*, May 2021.
8. Tim Kralj, *Integrating Julia and OpenCilk*, May 2021.
9. Nickolas Stathas, *Towards an Expressive Framework for High-Throughput Graph Neural Network Training on Large Graphs*, May 2021.
10. Helen He, *Performance Engineering of Reactive Molecular Dynamics Simulations*, May 2021.
11. Sean Fraser, *Computing Included and Excluded Sums Using Parallel Prefix*, May 2020.
12. Grace Q. Yin, *Parallel Exception Handling in Cilk*, jointly supervised by Tao B. Schardl, May 2020.
13. Severyn Kozak, *Chasing Zero Variability in Software Performance*, jointly supervised by Tao B. Schardl, May 2020.
14. Matthew S. Hutchinson, *Applying High Performance Computing to Early Fusion Video Action Recognition*, jointly supervised by Vijay Gadepally, May 2020.
15. Brian Wheatman, *Image Alignment and Dynamic Graph Analytics: Two Case Studies of How Managing Data Movement Can Make (Parallel) Code Run Fast*, May 2019.

16. William S. Moses, *How Should Compilers Represent Fork-Join Parallelism?*, May 2017.
17. Tana Wattanawaroon, *Local versus Global Tables in Minimax Game Search*, May 2014.
18. Warut Suksompong, *Bounds on Multithreaded Computations by Work Stealing*, May 2014.
19. Tim Kaler, *Chromatic Scheduling of Data-Graph Computations*, May 2013.
20. Ruben Perez, *Speculative Parallelism in Intel Cilk Plus*, May 2012.
21. Tao Benjamin Schardl, *Design and Analysis of a Nondeterministic Parallel Breadth-First Search Algorithm*, May 2010.
22. Jelani Nelson, *External-Memory Search Trees with Fast Insertions*, May 2006.
23. Tushara Karunaratna, *Parallel Race Detection*, August 2005.
24. Tim Olsen, *LexTix: A Multimedia Lecture-Viewer*, May 2005.
25. John S. Danaher, *The JCilk-1 Runtime System*, May 2005.
26. Jim Sukha, *Memory-Mapped Transactions*, May 2005.
27. Siddhartha Sen, *Dynamic Processor Allocation for Adaptively Parallel Jobs*, August 2004.
28. Kai Huang, *Data-Race Detection in Transactions-Everywhere Parallel Programming*, May 2003.
29. Matthew S. DeBergalis, *A Parallel File I/O API for Cilk*, May 2000.
30. David B. Berman, *Efficient Parallel Binary Decision Diagram Construction Using Cilk*, May 2000.
31. Jeremy Sawicki, *Using Cilk for Parallel Computation in MATLAB*, May 1999.
32. Ching Law, *A New Competitive Analysis of Randomized Caching*, May 1999.
33. Igor B. Lyubashevskiy, *Portable Fault-Tolerant File I/O*, May 1998.
34. Nathaniel A. Kushman, *Identifying and Fixing Processor Performance Monotonicity Problems: A Case Study of the SUN UltraSPARC*, May 1998.
35. Guang-Ien Cheng, *Algorithms for Data-Race Detection in Multithreaded Programs*, May 1998.
36. Andrew F. Stark, *Debugging Multithreaded Programs that Incorporate User-Level Locking*, May 1998. (Winner of the William Martin Prize for best computer science M.Eng. thesis at MIT.)
37. Philip Lisiecki, *Macroscheduling in the Cilk Network of Workstations Environment*, May 1996.
38. Daricha Techopitayakul, *Dynamic Parallel Tables*, May 1995.
39. Howard J. Lu, *Heterogeneous Multithreaded Computing*, May 1995.
40. Robert C. Miller, *A Type-checking Preprocessor for Cilk 2, a Multithreaded C Language*, May 1995.

24. Engineers Theses

1. Flavio Rose, *Models for VLSI Circuits*, March 1982.
2. Matthew Kilgore, *Fast Reducer Hyperobjects*, February 2022.

25. Doctoral Theses, Supervisor

1. William H. Kuszmaul, *Randomized Algorithms that Achieve the Unexpected*, August 2023.
2. William S. Moses, *Compiler Abstractions and Transformations to Reduce Programming Burden*, May 2023.
3. Helen Jiang Xu, *The Locality-First Strategy for Developing Efficient Multicore Algorithms*, February 2022.
4. Timothy F. Kaler, *Programming Technologies for Engineering Quality Multicore Software*, September 2020.
5. Tao B. Schardl, *Performance Engineering of Multicore Software: Developing a Science of Fast Code for the Post-Moore Era*, September 2016.
6. William C. Hasenplaugh, *Parallel Algorithms for Scheduling Data-Graph Computations*, February 2016.
7. I-Ting Angelina Lee, *Memory Abstractions for Parallel Programming*, March 2012.
8. Jim Sukha, *Composable Abstractions for Efficient Dynamic-Threaded Programs*, August 2011.
9. Jeremy T. Fineman, *Provably Good Algorithms for Atomicity, Caching, and Scheduling*, August 2009.
10. Kunal Agrawal, *Scheduling and Synchronization for Multicore Processors*, August 2009.
11. Yuxiong He, *Provably Efficient Adaptive Scheduling of Parallel Jobs on Multiprocessors*, July 2007. (School of Computer Engineering, Nanyang Technological University, and Computer Science Program, Singapore-MIT Alliance. Cosupervised by Wen-Jing Hsu.)
12. Matteo Frigo, *Portable High-Performance Programs*, June 1999.
13. Keith H. Randall, *Cilk: Efficient Multithreaded Computing*, May 1998.
14. Christopher F. Joerg, *Cilk: A System for Parallel Multithreaded Computing*, January 1996.
15. Robert D. Blumofe, *Executing Multithreaded Programs Efficiently*, September 1995.
16. Sivan Toledo, *Quantitative Performance Modeling of Scientific Computations and Creating Locality in Numerical Algorithms*, May 1995.
17. Bradley C. Kuszmaul, *Synchronized MIMD Computation*, May 1994.

18. Marios C. Papaefthymiou, *Timing Optimization of VLSI and Parallel Systems*, August 1993.
19. Thomas H. Cormen, *Virtual Memory for Data Parallel Computing*, September 1992.
20. Alexander T. Ishii, *Timing Verification of Level-Clocked Circuits*, October 1991.
21. James K. Park, *The Monge Array: An Abstraction and Its Applications*, May 1991.
22. Shlomo Kipnis, *Organization of Systems with Bussed Interconnections*, August 1990.
23. Cynthia A. Phillips, *Theoretical and Experimental Analyses of Parallel Combinatorial Algorithms*, October 1989.
24. Ronald I. Greenberg, *Efficient Interconnection Schemes for VLSI and Parallel Computation*, September 1989.
25. Bruce M. Maggs, *Locality in Parallel Computation*, September 1989.
26. Guy E. Blelloch, *Scan Primitives and Parallel Vector Models*, September 1988.
27. Serge A. Plotkin, *Graph-Theoretic Techniques for Parallel, Distributed, and Sequential Computation*, August 1988.
28. F. Miller Maley, *Single-Layer Wire Routing,* August 1987.
29. Andrew V. Goldberg, *Efficient Graph Algorithms for Sequential and Parallel Computers,* February 1987.
30. Sandeep N. Bhatt, *The Complexity of Graph Layout and Channel Routing for VLSI,* January 1984.
31. Ron Y. Pinter, *Routability and Its Impact on Placement Algorithms for Integrated Circuits,* August 1982. (Cosupervised by Ronald L. Rivest.)

26. Doctoral Theses, Reader

1. Siddhartha Visveswara Jayanti, *Simple, Fast, Scalable, and Reliable Multiprocessor Algorithms*, February 2023.
2. Jeffrey Bosboom, *Exhaustive Search and Hardness Proofs for Games*, September 2020.
3. Fredrik Kjølstad, *Compiler Techniques and Language Concepts for Sparse Linear and Tensor Algebra*, June 2019.
4. Julian Shun, *Shared-Memory Parallelism Can Be Simple, Fast, and Scalable*, May 2015. (School of Computer Science, Carnegie Mellon University.)
5. Jonathan Ragan-Kelley, *Decoupling Algorithms from the Organization of Computation for High Performance Image Processing*, June 2014.
6. Harsha Vardhan Simhadri, *Program-Centric Cost Models for Locality and Parallelism*, September 2013. (School of Computer Science, Carnegie Mellon University.)

7. Anne Benoit, *Scheduling Pipelined Applications: Models, Algorithms and Complexity*, July 2009. (Habilitation thesis for Ecole Normale Supériere de Lyon, France.)
8. Sam Larsen, *Compilation Techniques for Short-Vector Instructions*, April 2006.
9. Russell Schwartz, *The Local Rules Dynamics Model for Self-Assembly Simulation*, February 2000.
10. Rajeev Barua, *Maps: A Compiler-Managed Memory System for Software-Exposed Architectures*, January 2000.
11. Parry Husbands, *Interactive Supercomputing*, January 1999.
12. James Alexander Stuart Fiske, *Thread Scheduling Mechanisms for Multiple-Context Parallel Processors*, May 1995.
13. Feng Ming Dong, *Compilation and Run-Time Environment of Parallel Lisp on Distributed Systems*, May 1995. (Department of Information Systems and Computer Science, National University of Singapore.)
14. Yuan Ma, *Fault-Tolerant Sorting Networks*, June 1994.
15. David Williamson, *On the Design of Approximation Algorithms for a Class of Graph Problems*, September 1993.
16. Clifford Stein, *Approximation Algorithms for Multicommodity Flow and Shop Scheduling*, August 1992.
17. Filip Van Aelten, *Automatic Procedures for the Behaviorial Verification of Digital Designs*, May 1991.
18. Joel Wein, *Parallel Computation and Combinatorial Optimization: Scheduling Algorithms for Parallel Machines and Parallel Algorithms for the Assignment Problem*, August 1991.
19. Eric Schwabe, *Theoretical Issues in Parallel Computation: Hypercube-Related Architectures and Dynamic Structures*, May 1991.
20. Bonnie Berger, *Using Randomness to Design Efficient Deterministic Algorithms*, May 1990.
21. Mark R. Newman, *Randomness and Robustness in Hypercube Computation*, September 1989.
22. Alan T. Sherman, *Cryptology and VLSI,* October 1986.
23. Daniel Weise, *Hierarchical, Multilevel Verification of MOS/VLSI Circuits,* August 1986.
24. Thang Nguyen Bui, *Graph Bisection Algorithms,* January 1986.
25. Peter W. Shor, *Random Planar Matching and Bin Packing,* August 1985. (MIT Mathematics Department.)
26. James B. Saxe, *Generalized Transformations on Algorithms: Two Case Studies,* August 1985. (Department of Computer Science, Carnegie-Mellon University.)

27. G. A. Boughton, *Routing Networks for Packet Communication Systems*, August 1984.
28. Aloysius K. Mok, *Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment*, May 1983.

27. Postdoctoral Advisees

1. Yuhao Chen
2. Rezaul Chowdhury
3. Maryam Mehri Dehnavi
4. Mingdong Feng
5. Andrew Goldberg
6. Yuxiong He
7. Alexandros-Stavros Iliopoulos
8. Tim Kaler
9. Shahin Kamali
10. I-Ting Angelina Lee
11. Bruce M. Maggs
12. Aske Plaat
13. Tao B. Schardl
14. Kyle Singer
15. Volker Strumpen
16. Gideon Stupp
17. Yuzhen Xie

28. Professional Biography

Charles E. Leiserson received his B.S. from Yale University in 1975 and his Ph.D. from Carnegie Mellon University in 1981. He joined the faculty of the Massachusetts Institute of Technology in 1981, where he is now the Edwin Sibley Webster Professor in MIT's Electrical Engineering and Computer Science (EECS) Department. He formerly served as Associate Director and Chief Operating Officer of the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), the largest on-campus laboratory at MIT, where he leads the Supertech research group and is a member of the Theory of Computation research group. He currently serves as the MIT Faculty Director of the USAF-MIT AI Accelerator and leads its Fast AI project. He is a Margaret MacVicar Faculty Fellow, the highest recognition at MIT for undergraduate teaching. He is a Fellow of four professional societies—AAAS, ACM, IEEE, and SIAM—and he is a member of the National Academy of Engineering. He has received many Best Paper awards at prestigious conferences, as well as major awards, including the ACM-IEEE Computer Society Ken Kennedy Award, the IEEE Computer Society Taylor L. Booth Education Award, the ACM Paris Kanellakis Theory and Practice Award, and the ACM and Hertz Foundation Doctoral Dissertation Awards. Almost all the contributions he has made represent joint work with his students and other collaborators.

Professor Leiserson's current research centers on ***software performance engineering***—making computer programs run fast (or otherwise use fewer resources) by whatever means available: algorithms, parallel computing, caching, compilation, processor pipelining, bit tricks, vectorization, etc. His current work centers on the award-winning ***Cilk task-parallel programming language and runtime system***, which he began developing in the mid-1990's. Several open-source and commercial compilers have supported Cilk, and Cilk has served as an inspiration for many other task-parallel systems. In 2006 Professor Leiserson founded the MIT spinoff ***Cilk Arts, Inc.***, which was acquired by Intel Corporation in 2008. His research team at MIT began work on the open-source ***OpenCilk*** platform in 2016, capitalizing on their ***Tapir/LLVM compiler***, which won the 2017 Best Paper award at the ACM Principles and Practice of Parallel Programming (PPoPP) conference. The OpenCilk compiler embeds fork-join parallelism directly in the compiler's intermediate representation, enabling the compiler to optimize across parallel control constructs with only minor changes to its existing analyses and code transformations.

Professor Leiserson's research contributions span 45 years. In 1978, he coauthored the first paper on ***systolic architectures***, high-performance hardware algorithms that can be directly implemented as integrated circuits. In 1979, he invented the ***retiming*** method of digital-circuit optimization, as well as efficient optimization algorithms for retiming that are used to optimize clocked circuitry in almost all electronic-design systems today. In the 1980's and early 1990's, he designed and led the implementation of the network architecture for Thinking Machines Corporation's ***Connection Machine Model CM-5***. The CM-5, which was the world's most powerful supercomputer in 1993, incorporated the "universal" ***fat-tree interconnection network*** he conceived at MIT. Fat-trees are now the preferred interconnect strategy for InfiniBand technology. In the 1990's, he developed several Cilk-based ***parallel chess-playing programs***, including ⋆Socrates and Cilkchess, winning numerous prizes in international competition. In the late 1990's, he introduced the notion of ***cache-oblivious algorithms***, which automatically adapt to the memory hierarchies of modern machines without tuning. As Director of System Architecture at Akamai Technologies while on leave from

MIT in 1999–2001, Professor Leiserson led the engineering team that developed Akamai's ***content-distribution network*** comprising (at the time) tens of thousands of servers deployed across the globe.

An engaging educator, Professor Leiserson developed the MIT undergraduate courses on algorithms, discrete mathematics for computer science, and software performance engineering. He is well known as a coauthor of ***Introduction to Algorithms*** (The MIT Press), the leading textbook on computer algorithms, now in its fourth edition, which has sold over one million copies worldwide and is one of the most cited publications in all of computer science. His annual workshop on ***Leadership Skills for Engineering and Science Faculty*** has educated hundreds of faculty at MIT and around the world in the human issues involved in leading technical teams in academia. He was the founding Workshop Chair for the MIT ***Undergraduate Practice Opportunities Program*** (***UPOP***), which teaches sophomore majors in engineering how human-centered leadership skills can leverage their technical skills in professional environments. He was for many years the head of the computer-science program for the Singapore-MIT Alliance, one of the first ***distance-education*** collaborations. Professor Leiserson has graduated over two dozen doctoral students and supervised more than 80 master's and bachelor's students.