CS 111 presentation on TCS

Mark C. Wilson

2007-10-05

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへで

What is (and isn't) Theoretical Computer Science?

• The abstract study of computational processes, independent of particular hardware (running on an idealized model of a computer).

- The abstract study of computational processes, independent of particular hardware (running on an idealized model of a computer).
- "[Theoretical] computer science is no more about computers than astronomy is about telescopes" (E. W. Dijkstra).

- The abstract study of computational processes, independent of particular hardware (running on an idealized model of a computer).
- "[Theoretical] computer science is no more about computers than astronomy is about telescopes" (E. W. Dijkstra).
- It has a strong mathematical flavour. Since it is not experimental, correctness of an idea is established by mathematical proof.

- The abstract study of computational processes, independent of particular hardware (running on an idealized model of a computer).
- "[Theoretical] computer science is no more about computers than astronomy is about telescopes" (E. W. Dijkstra).
- It has a strong mathematical flavour. Since it is not experimental, correctness of an idea is established by mathematical proof.
- Includes: algorithm analysis and design; computational geometry, information theory, cryptography, quantum computing, study of randomness, computational algebra, bioinformatics, formal languages.

- The abstract study of computational processes, independent of particular hardware (running on an idealized model of a computer).
- "[Theoretical] computer science is no more about computers than astronomy is about telescopes" (E. W. Dijkstra).
- It has a strong mathematical flavour. Since it is not experimental, correctness of an idea is established by mathematical proof.
- Includes: algorithm analysis and design; computational geometry, information theory, cryptography, quantum computing, study of randomness, computational algebra, bioinformatics, formal languages.
- Has many applications to, and is inspired by, concrete problems in science, engineering, etc.

Some big ideas in TCS

- algorithm: systematic solution procedure for computational problem;
- uncomputability: not every problem can be solved by computation;
- recursion (self-reference): a program can use its own output as input;
- universality: anything a supercomputer can compute, an ancient laptop can too;
- tractability: some problems are intrinsically harder than others, no matter how clever we are;
- one-way functions: some functions are easy to compute but their inverse is very hard;
- zero-knowledge proofs (ZKP): Alice can convince Bob that she knows a secret without revealing it.

Algorithms

- A systematic procedure for solving a computational problem.
- Any programming task requires algorithmic thinking. Examples:
 - How does TeX decide how to insert spaces?
 - Sort a big database.
 - Find the shortest path between two points in a large network.
 - Find all matches for *eju* in an English dictionary.
 - Find the shortest tour for a travelling sales rep, visiting all towns.
 - Decompose a large integer into prime factors.
- Finding some algorithm is often easy, but finding an efficient one is often hard. See COMPSCI220, 320,

E 990

・ロン ・雪と ・雨と

Cave story for ZKP



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Some recent applications of CS Theory

• Engineering: Fast Fourier Transform revolutionized signal processing.

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

- Engineering: Fast Fourier Transform revolutionized signal processing.
- Biology: algorithmic approach vastly speeded up work on the Human Genome Project.

- Engineering: Fast Fourier Transform revolutionized signal processing.
- Biology: algorithmic approach vastly speeded up work on the Human Genome Project.
- Programming languages: fast efficient algorithms for searching, sorting, string matching, ... now implemented.

- Engineering: Fast Fourier Transform revolutionized signal processing.
- Biology: algorithmic approach vastly speeded up work on the Human Genome Project.
- Programming languages: fast efficient algorithms for searching, sorting, string matching, ... now implemented.
- Computer engineering: chip layout and parallel machine architecture improved by graph algorithms.

- Engineering: Fast Fourier Transform revolutionized signal processing.
- Biology: algorithmic approach vastly speeded up work on the Human Genome Project.
- Programming languages: fast efficient algorithms for searching, sorting, string matching, ... now implemented.
- Computer engineering: chip layout and parallel machine architecture improved by graph algorithms.
- E-commerce: public key cryptography enables you to buy things on the web; ZKP allows secure elections, auctions, etc.

- Engineering: Fast Fourier Transform revolutionized signal processing.
- Biology: algorithmic approach vastly speeded up work on the Human Genome Project.
- Programming languages: fast efficient algorithms for searching, sorting, string matching, ... now implemented.
- Computer engineering: chip layout and parallel machine architecture improved by graph algorithms.
- E-commerce: public key cryptography enables you to buy things on the web; ZKP allows secure elections, auctions, etc.
- Internet: Google's success is largely due to its PageRank algorithm; routing protocols have been improved by graph algorithms.

Trendy example — bioinformatics

- Has become a central part of biological sciences with many applications in ecology, evolutionary biology, diseases, molecular and cell biology.
- Involves using mathematics, statistics and computer science tools to analyse rapidly growing databases of biological data.
- New algorithms are the biggest requirement right now, because we are being buried under an avalanche of data.



Alignments picture

	1 1	10	29	30	40	50	60	79	80	90	100	110	120	130	140
Human Xenopus	TACCTOCTT	CATCOTOCO	ACTACCATA	COTTOTOT	CARAGATTRA CARAGATTRA	CCATCCATO	TOTANOTACO	CACCCCCCCT CACCCCCCCCT	ACACTGAAAC ACACTGAAAC	OCCANTOCO	CATTAAATCI CATTAAATCI	NOT INTGOT T	COTTOSTOS	OTCOCTCCTC OTCCATC	TOTT
Similarity						V	V.								
Human	ACTTOGATA	50	160	0TANTACAT	GOCOACCCC		200		ATTATCAGE		240	250	260	270	280
Xenopus	ACTTGGATA	ACTOTOGT	ATTCTACAG	CTANTACAT	GECGACGAGE	OCTOACCCCC	AC	CONTRECTOR	ATTTATCACA	CANANCOAN	CCCG				CCCC
Similarity	2	20	300	310	320	330	340	350	360	370	380	390	400	410	420
Human	COCCUTTO	TCACTOTA	ATAACCTCC	COCCATCO	CACCCCCCC	TOCCOCON	CONCERNITO	CANCOTOTOO	COTATCAACT	TCONTOCTN	TOCCOTOC	TOPROCATOOT	ACCACCCOT	SACOGGAN	CACO
Similarity		TOACTOTA.	ATRACCICC.	occourteo	CACOL-CCCC	Henceuceun	CONTRONTIO		COLVICANCE.	TEGNICOTA		ST NOCK LOO I	ACCACOUCT	NACOCOURA.	CAUC
11	4	30	440	450	460	470	480	490	500	510	520	530	540	550	560
Xenopus	OTTCCATTO	CCACACO	ACCOTCACA	AACCCCTAC	CACATCCAAC	CAACCEACEA	COCOCCEARA	TTACCCACTO	CCACCCCCC	ACCTACT CA	CAAAAA	CANTACACCA	TOTTTCCAC	GCCCTCTAAT	TCCA
Similarity									L.						
Human	ATCACTCCA	CTTTAAAT	OTTTAACCA	GATCOATT	GOAGGGCAAG	TOTOGTOCCA	GENOCOCCO	TAATTCCAGO	TCCANTAGO	ATATTAAAG	TOCTOCACT	AAAAAGOTO	TACTICAT	OTTOGOAGO	secee
Xenopus	NTGAGTACA	CTTTANATO	OTTTANCOA	GGATCTATT	GGAGGGCAAG	TOTOGTOCCA	GONOCCOCCO	TANTTCOAGO	TCCANTAGOO	ATATTANAS	TOCTOCACT	TAAAAAGOTO	TACTICAT	DTTOGONTCO	ACCT
Similarity	7	10	720	730	740	750	760	770	780	790	800	810	820	830	840
Human	0000000000	COCCARCO COCCARCO	CÁCCCACCO	COCOTOCOCO	00000-1000	TOT COOCCOCC	COCTOCATOO	TOTTACCTON	OTOTOCOCO O	COCCOCANO	OTTACTT	SAAAAATTA	AOTOTICAN	ACCACCCCCC	JACCO
Similarity			le la	V V			U U		V				V		ŨŬ
Human	8	50	860	870	880	890	900	910	920	930	940	950	960	970	980
Xenopus	OCOTOONTA	CTTCACCT	CONTANT	OANTAGONO	TOCCOTTOTA	TT TOT TOOT	TTTCCCAACT	COOCCATON	TANGAGOGA	0000000000	ATTOOTATT	TOCCOUTAG	GOTCANATT	OTTOCACCO	ADDOC
Similarity		0.0	1.000	1.010	1,030	1.010	1.040	1.010	1.040	1.070	1.080	1 000	1,100	1,110	1,110
Human	AGACGGACO	AGAGEGAN	CATTOCO	AAGAATOTT	TTOATTANTO	NACANCORA	OTCOCACOTT	CONTONCONT	CAGATACCOT	OTAOTTOCO	CONTANACO	TOCCACCO	CONTOCOCO	GOCOTTATTO	COONT
Xenopus Similarity	ACACCAACC	ANACCCAN	CONTINCCO	AAGAATOTT	TTENTTANTE	NACANCOAAA	CTCCCACCTT	CCAACACGAT	CAGATACCOT	CTACITCCG	ACCATAAACCI	TOCCCACTA	CONTECCO	SOCOTINITO	CCAT
Similarity	1,	130	1,140	1,150	1,160	1,170	1,180	1,190	1,200	1,210	1,220	1,230	1,240	1,250	1,260
Human Xenopus	GACCOGCCG	AGGAGOTTO	COCCARACO	AAAGTOTTT	GGGTTCCGGG GGGTTCCGGG	DOGACTATOS	TTGCARAGET TTGCARAGET	GAAACTTAAA GAAACTTAAA	GOANTT GACO	AAGGGCACC		SAGCOTOCOO SAGCOTOCOO	TAATT TGA	CTCAACACCC	CAAA
Similarity															
Human	COTCACCO	270 OCCCCCAC	1,280	1,290	1,300	1,310	1,320 COT 000TOOT	1,330 00100AT 000	1,340	1,350	1,360	1,370	1,380	1,390 0ACTCT00CA	1,400 NEGOT
Xenopus	COTCACCO	OCCCCCACI	CCCANACCA	TOACAGAT	TONTAGOTOT	TOTOGATTO	TOTOCOTCOT	COTCONTOCO	COTTOT TAGT	COTCOACCO	VITTOTOTOO	TANTICCON	NACOAACON	ACTOCTCCA	ALCOL
Similarity	1,	410	1,420	1,430	1,440	1,450	1,460	1,470	1,480	1,490	1,500	1,510	1,520	1,530	1,540
Human	AACTACTTA	COCACCO	CACCOT	COCOTOCOO	CARCTTOTTA	ACCCACAR	TOCOTTOAC	CEACECCACA	TCACCAATA	CACOTOTOT	ATCCCCTTA	CATOTOCOCO	CONCORD	SCOTACACTO	ACTO
Similarity	ANGLAGIN	COCONCECC					Toocorrieno			Calconet of	ALCONCILLA.	ALCICCOU.			
Human	1,	550	1,560	1,570	1,580	1,590	1,600	1,610	1,620	1,630	1,640	1,650	1,660	1,670	1,680
Xenopus	CATCACCOT	OTOTOTACO	CECCCCCCA	CACCECCCO	CTARCCCCCT	AACCCCCTT	COTCATAGOG	ATCCCCCCATT	CANTENTE	CCATCAACG	ACCANTTOCC	ACTARCTOCO	CTCATAACC	TCCCCTTCA	TAAC
Similarity	ν.	693	1 700	1.710	1.720	1 210	1.740	1.750	1.760	1.770	1 700	1 790	1 800	1 810	1,820
Human	TCCCTOCCO	TTOTACM	ACCOCCCO	CONTACTAC	CONTIGONTO	TTAGTON	OCCOTCOONT	coccecce			COORCEOT	ACAACACCO	CONNETTON	CTATCTACAS	GANO
Similarity	TCCCTOCCC	TTTOTACA	ACCOCCCO	COCTACTAC	CONTTOONTO	TTTAGTON	OTCOTCOONT	00000000000	00001000-0	ACCOCCCTO	COONCOCO	SAGANGACOA	CANACTTON	DINTOTAGAG	GNNG
	1,	830	1,840	1,850	1,860	1,869									
Human	TAAAAOTCO	TAACAAGO	TICCOTAGO	TCAACOTOO	CONNOONTON	TA									
Similarity															

Tree of life picture



◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 _ のへぐ

The biggest open problem in Computer Science

- There is a class P of problems that can all be solved in "polynomial time" (roughly speaking, quickly). There is another class NP of problems for which a solution can be verified in polynomial time provided you can guess it.
- Obviously *P* is contained in *NP*, but "obviously" *NP* must be much bigger. No one has been able to prove this.
- The Clay Mathematics Institute has a 1 million USD prize.
- If P = NP then most encryption methods will fail and most mathematics will be done by machine. If $P \neq NP$ then many important practical computational problems will never be solved quickly, and randomization does not really help.
- Some problems thought to be in NP but not P: travelling salesman, subset sum, boolean satisfiability, graph colouring.
- It is known that if a quantum computer could be built, it could solve the above problems in polynomial time.

TCS group at UoA

- 6 permanent staff, active in research, with research interests from fundamental to trendy.
- 2 postdoctoral research fellows
- 10(?) PhD and Masters students
- The CDMTCS (http://www.cs.auckland.ac.nz/CDMTCS/) provides a framework for seminars, conferences, student prizes, etc.
- In 2008 there will be a major programme Algorithms: New Directions and Applications that will involve everyone from staff to the general public. Watch out for it!

CS theory staff

- Cris Calude. Professor. Originally from Romania. Research interests: unconventional models of computation.
- Michael Dinneen. Senior Lecturer. Originally from USA. Research interests: graph algorithms.
- Alexei Drummond. Senior Lecturer. Research interests: bioinformatics.
- Bakhadyr (Bakh) Khoussainov. Professor. Originally from Uzbekistan. Research interests: automata, logic.
- Andre Nies. Senior Lecturer. Originally from Germany. Research interests: computability theory (logic), randomness.
- Mark Wilson. Senior Lecturer. Research interests: analysis of algorithms, discrete mathematics.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

CS Theory undergraduate courses

• COMPSCI 105 (principles of computer science)

- COMPSCI 105 (principles of computer science)
- COMPSCI 220 (algorithms and data structures), COMPSCI 225 (discrete mathematics)

- COMPSCI 105 (principles of computer science)
- COMPSCI 220 (algorithms and data structures), COMPSCI 225 (discrete mathematics)
- COMPSCI 320 (algorithm design), COMPSCI 350 (computability), COMPSCI 369 (bioinformatics)

- COMPSCI 105 (principles of computer science)
- COMPSCI 220 (algorithms and data structures), COMPSCI 225 (discrete mathematics)
- COMPSCI 320 (algorithm design), COMPSCI 350 (computability), COMPSCI 369 (bioinformatics)
- COMPSCI 720 (advanced algorithms), COMPSCI 750 (complexity theory), and more.

- COMPSCI 105 (principles of computer science)
- COMPSCI 220 (algorithms and data structures), COMPSCI 225 (discrete mathematics)
- COMPSCI 320 (algorithm design), COMPSCI 350 (computability), COMPSCI 369 (bioinformatics)
- COMPSCI 720 (advanced algorithms), COMPSCI 750 (complexity theory), and more.
- It is important to take a good amount of mathematics also.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Some benefits of CS Theory training

• General skills that remain relevant as technology changes.

▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

- General skills that remain relevant as technology changes.
- Ability to think abstractly and extract the essence of a messy real-world problem, then solve it.

- General skills that remain relevant as technology changes.
- Ability to think abstractly and extract the essence of a messy real-world problem, then solve it.
- Win programming competitions! it is impossible to win without knowing the material in CS320. The UoA team were 12th in the world last year.

- General skills that remain relevant as technology changes.
- Ability to think abstractly and extract the essence of a messy real-world problem, then solve it.
- Win programming competitions! it is impossible to win without knowing the material in CS320. The UoA team were 12th in the world last year.
- "Part of Microsoft's interview process is answering tough questions like the ones we put on our CS320 exams."

- General skills that remain relevant as technology changes.
- Ability to think abstractly and extract the essence of a messy real-world problem, then solve it.
- Win programming competitions! it is impossible to win without knowing the material in CS320. The UoA team were 12th in the world last year.
- "Part of Microsoft's interview process is answering tough questions like the ones we put on our CS320 exams."
- "The software company that I am involved with has 7 fulltime software developers and they are basically all theory guys – i.e. know the hard stuff ... We basically only hire people that know how to think. Who cares whether or not they have done some specific applied paper. They are going to have to constantly learn on the job anyway ..."

Some of our recent graduates

- Liu Xiong, developer for Microsoft (Redmond, Washington, USA).
- Tiki Wong, works for Microsoft (Redmond, Washington, USA).
- Nodira Khoussainova, PhD student at University of Washington (Seattle, USA). Had research internship at Microsoft Research.
- Jiamou Liu, PhD student with Bakh Khoussainov, spends half the year at Cornell University (USA).
- Owen Auger, PhD student in UoA Engineering Science Dept, working on electricity network modelling.
- Sasha Rubin, postdoctoral fellow supported by NZ government fellowship, visiting University of Wisconsin (USA) and Aachen (Germany).

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

The future: bright

• TCS provides the theoretical foundation for scientific computing.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

- TCS provides the theoretical foundation for scientific computing.
- The older sciences have used mathematical formulae to describe phenomena.

- TCS provides the theoretical foundation for scientific computing.
- The older sciences have used mathematical formulae to describe phenomena.
- The newer sciences (neurophysiology, modern biology, realistic economics, ...) have found that formulae are less useful than algorithms.

- TCS provides the theoretical foundation for scientific computing.
- The older sciences have used mathematical formulae to describe phenomena.
- The newer sciences (neurophysiology, modern biology, realistic economics, ...) have found that formulae are less useful than algorithms.
- "The Algorithm's coming-of-age as the new language of science promises to be the most disruptive scientific development since quantum mechanics." (Bernard Chazelle)

- TCS provides the theoretical foundation for scientific computing.
- The older sciences have used mathematical formulae to describe phenomena.
- The newer sciences (neurophysiology, modern biology, realistic economics, ...) have found that formulae are less useful than algorithms.
- "The Algorithm's coming-of-age as the new language of science promises to be the most disruptive scientific development since quantum mechanics." (Bernard Chazelle)
- There are many exciting research challenges ahead!

References

- UoA CS Theory group, http://www.cs.auckland.ac.nz/research/groups/theory/
- The Algorithm: Idiom of Modern Science, by Bernard Chazelle, http://www.cs.princeton.edu/~chazelle/.
- Is The Thrill Gone? by Sanjeev Arora and Bernard Chazelle, www.cs.princeton.edu/ chazelle/pubs/cacm05.pdf.
- Theory Matters Wiki, http://theorymatters.org/pmwiki/pmwiki.php