Great Ideas in Computing

University of Toronto CSC196
Fall 2022

Week 1: September 12-September 16 (2022)

Course Organization

Course Instructor: Allan Borodin

- Email: bor@cs.toronto.edu
- Instructor course email: csc196-2022-09@cs.toronto.edu

Teaching Assistant : Vignesh Edithal
TA Course Email : csc196-2022-09-ta@cs.toronto.edu
Note: The TA course email may take a couple of days to set up

I strongly encourage questions and discussions in class. In addition, we are also using piazza for questions and discussions.

Using piazza, you can also answer questions posed by others. You can pose or answer questions anonymously or using your name. The benefit of in class and piazza (over email) is that the entire class benefits from the discussion. Please sign up as soon as it is available.

Course organization continued

The class will normally meet M,W,F at 9AM in MY 320.

Usually I will be presenting on Mondays and Wednesdays and the tutorials will be on Fridays. You should not have a conflict with any of these three hour classes.

The next three weeks will be exceptions. Next week the tutorial will be on Wednesday, September 21. with the class on Monday the 19th and Friday the 23rd. The following week there will not be a tutorial and the class will be on Wednesday, September 28 Friday, September 30. Then in the week of October 3, the class will be on Monday, October 3 and Friday, October 7 with the tutorial on Wedsnesday, October 5. Sorry that we have to rearrange the next three weeks but after that we should be able to stay with the standard Monday, Wednesday classes and tutorials on Fridays.

Course web site: http://www.cs.toronto.edu/∼196f22/
My slides will be on the web page. I will also post various documents on the web page.
There are also links to the previous version of CSC196 (fall 2021, fall

COVID considerations

Lets hope we do not have to worry about COVID.

Everyone has their own sense of how much to still be concerned about COVID. I now personally feel less worried about the current status of COVID. Please speak to me if you have any concerns.

While the mask mandate has been paused as of 1 July 2022, the use of medical masks continues to be strongly encouraged in high density indoor settings where physical distancing is not possible. We ask everyone to respect each other's decisions, comfort levels, and health needs. Instructors may strongly encourage and request that students wear masks in their classes and may outline this expectation for their courses, both on the first day of class and in syllabi. However, instructors may not require students to wear masks while the mandate is paused. Please also be aware that some students are not able to wear masks for disability- related reasons. The University will continue to monitor public health conditions to adjust its response as needed, with updates posted at https://www.utoronto.ca/utogether/faqs#masks

Announcements: RSGs and UT AI Club

Recognized Study Groups (RSGs) are voluntary, peer-led study groups of up to eight students enrolled in the same Arts and Science course. Last year, over 3,000 students participated in RSGs for courses spanning all disciplines and class sizes.

[RSG Poster]

Announcements continued

RSGs are led by students enrolled in the course who receive training on group study strategies and academic integrity. Our feedback surveys show that most students led or joined an RSG because their instructor recommended it in-class. Please inform students about RSGs for your course on Quercus, in your course syllabus or in your first class using this material:

UT AI Club A representative of the UT AI Club will be coming at the end of todays class to tell you about the club.

Preliminaries

Getting to know each other

- A little about myself
- Your plans at the University?

What is this course about?

- FAS Calendar about First Year Foundational Seminars
  https://www.artsci.utoronto.ca/future/academic-opportunities/first-year-opportunities/first-year-foundations-seminars
  One might say that all of these first year courses are an "Introduction to Critical Thinking".
- Brief theme of our CSC196 course: Great Ideas in Computing.
  What constitutes a "great idea"?

Preliminaries

Getting to know each other

- A little about myself
- Your plans at the University?

What is this course about?

- FAS Calendar about First Year Foundational Seminars
  https://www.artsci.utoronto.ca/future/academic-opportunities/first-year-opportunities/first-year-foundations-seminars
  One might say that all of these first year courses are an "Introduction to Critical Thinking".
- Brief theme of our CSC196 course: Great Ideas in Computing.
  What constitutes a "great idea"?

  Impact, surprise, elegance. Of course, it is easier to agree on great ideas in retrospect rather than as ideas are being introduced. And of course ideas rarely occur in a "vacuum"; usually there are similar ideas known and often the timing of when an idea becomes viable is very critical. It is also the case that credit for an idea is not always completely fair to all those involved.

More thoughts on what constitutes a great idea

- “It is a breakthrough”; meaning it accomplishes something that was not possible before.
- “It allows for new possibilities”
- “It provides an *optimal* solultion to a problem’
- “It is the first thing I would want to describe to someone not in CS”’

Great ideas, good decisions, good plans

- “At the time, it was not considered a good idea” Can we call an idea a great idea if it never gets adopted?

Great ideas, good decisions, good plans

- "At the time, it was not considered a good idea" Can we call an idea a great idea if it never gets adopted?
- How should we characterize a good decision or a good plan?

Great ideas, good decisions, good plans

- "At the time, it was not considered a good idea" Can we call an idea a great idea if it never gets adopted?
- How should we characterize a good decision or a good plan? A point of view: one probably should not judge the quality of a decision by outcomes. The quality of a good decision may simply be whether or not, given all the information available at the time the decision was made, the decision was a good or the best decision one could make.
- Decisions can be irrevocable or can be modified. Plans usually do evolve over time.

Great ideas, good decisions, good plans

- "At the time, it was not considered a good idea" Can we call an idea a great idea if it never gets adopted?
- How should we characterize a good decision or a good plan?
  A point of view: one probably should not judge the quality of a decision by outcomes. The quality of a good decision may simply be whether or not, given all the information available at the time the decision was made, the decision was a good or the best decision one could make.
- Decisions can be irrevocable or can be modified. Plans usually do evolve over time.
- It is important to understand that in all ideas, decisions, plans, there is always some degree of **uncertainty** as to how the "world" will unfold.
- In research and development (and more generally in life) how long should we stick with our intuition or when do we see that our beliefs and plans are inconsistent with facts or "common wisdom"?

Great ideas may have negative consequences

There can be unintended, undesireable consequences for ideas we may come to accept as great ideas? Is it then still a great idea?

- If a technology becomes a standard, it can also become a barrier to innovation. For example, we will soon encounter our first great idea "The von Neumann" model. Some have argued that it has impeded progress on alternative computational architectures.
- Social networks allow for the rapid spread of information and mis-information. It is claimed that social media deliberately fails to act on hate speech while social media companies argue they are actively removing hate speech above and beyond reported cases.
- Moreover, some claim that these companies do not want to eliminate hate speech and that they often target users which in turn reinforces divisions in society. (See the links to article by Starbird et al and a text file with some links to the issue of hate speech and bias on social media platforms.)
- For controversial and ethical issues, we may sometimes use a debate format where students argue for a point of view and not their view.

Predictions and the nature of research

Niels Bohr, Mark Twain, Yogi Berra: "Prediciitng the future is hard because it hasn't happened yet" and "it's tough to make predictions, especially about the future" Who knows who said it first.

- Predictions about computing and what and can and can't be done within some predicted time frame are often wrong. I posted a link to the 1955 Dartmouth summer project on artificial intelligence AI (where the term seems to have first appeared). Turing's 1950 article provided what is now called the Turing Test as to "whether or not it is possible for machinery to show intelligent behaviour". The Dartmouth project suggests that the indicated challenges for AI could be done over the summer.
- I posted a link to an article by John Backus about the view (that he proved wrong) that source level languages could never be nearly as efficient as machine code.
- There is also an article giving what one individual calls the "7 Worst Tech Predictions of All Time." But note that the quote attributed to T.J. Watson may never have happened.

Grading scheme, syllabus and possible topics

- The grading scheme will be based on 4 assignments (15% each), two quizzes (10% each), and class participation (20%). Students are expected to attend all classes regularly and participate actively. There will be no final exam.
- Assignments will be submitted on Markus
  https://markus.teach.cs.toronto.edu/2022-09
  Once you login you select the course CSC196
  Tentative dates appear on the next slide and I also posted tentative dates in the syllabus.
- It is difficult but not impossible to fail this course. Mainly do NOT plagarize. If you have any questons about plagarism, ask me.
- The syllabus (listed on the course web page) contains other organizational information.
- There is also an ambitious list of possible topics on the web page.
  What have we missed?
  When I ask a question in red, thats a strong invitation for YOU to join the conversation but don't wait for an invitation to speak up.

Relevant dates

- A0 September 22. This is part of the participation grade
- A1 October 7
- Q1 October 21
- A2 October 28
- *Note:* I think November 7 is the last day to drop a Fall (F) course. But if you think you are going to withdraw please do so as soon as possible so as to make room for someone on the wait list.
- Reading week is November 8-12.
- A3 November 18
- Q2 November 25
- A4 December 2
- Fall classes end December 7. We are allowed a possible makeup date on December 8 to compensate for the missed class on Thanksgiving.

End of Monday, September 13 class

We ended Monday starting to mention a list of possible topics. I added one more suggestion (cyber security) by a student. After quickly listing possible topics and the list of our guest presentations, we are ready to begin our first topic.

Possible Topics

- What is a great idea?
- What responsibilties do computer professionals have for the impact (and possible misuse) of the technology?
- What is a computer? The von Neumann architecture. Digital vs analogue. What were the alternatives? What else is possible (parallel, quantum)?
- The genius of Alan Turing; A mathematical definition of computable function. Interpreters. Non computable functions.
- How did computers and computing become a commodity? The amazing advances in software and algorithms came along with advances in hardware (cost, speed, memory size, physical size, power) and communication (cost, capacity and speed) . Demand for "killer applications" such as word processors, email, search engines, navigation systems, games).
- The internet; packet routing. TCP/IP.

More possible topics

- Fortran, the first commericial source level language (with an efficient compiler) First compiler often attributed to Grace Hopper. John Backus vs the prevailing view that compiled code would be too slow compared to machine code. The longevity of code vs the brevity of any particular machine; the cost of a machine vs the cost of software developers.
- How search engines work and what they do well and what (if anythng) they dont do well.
- The semantic web.
- A local great idea: NP completeness. What is and what is not *efficiently* computable.
- Complexity based cryptography; public key cryptography; digital signatures. Captchas.
- Another local great idea: deep neural networks and the success of machine learning (ML). Computers vs human thought and learning.
- HCI (human computer interaction). The mouse. Menus, click, paste and drag. Visualization.

And some more possible topics

- Information theory: the genius of Claude Shannon. Error correcting codes; compression.
- Social networks and the spread of information (and mis-information, conspiracies, etc). Targetting information to different communities. (What is a social network community?)
- Open Source. Wikipedia. Blogs
- Relational data bases. a
- Linear programming; dynamic programming and combinatorial optimization. How far can one go with conceptually simple algorithms? Dijkstra's algorithm and navigation systems.
- Distributed System primitives : mutual exclusion, consensus
- Differential privacy; extracting useful statistical information without sacrificing individual information.
- Algorithmic mechanism design; automated auctions. Algorithmic social choice; voting.

Great ideas are not isolated

To appreicate what we (possibly) cannot do efficiently, we should know what we can do efficiently by known methods such as linear programming.

Many computational problems now have improved solutions using machine learning techniques (ML). For example, search engines now use ML to improve the responses that you (personally) receive to a query.

Of course we will only be able to discuss a small subset of these ideas and yet

Great ideas are not isolated

To appreicate what we (possibly) cannot do efficiently, we should know what we can do efficiently by known methods such as linear programming.

Many computational problems now have improved solutions using machine learning techniques (ML). For example, search engines now use ML to improve the responses that you (personally) receive to a query.

Of course we will only be able to discuss a small subset of these ideas and yet

What great ideas have we might mised?

Great ideas are not isolated

To appreicate what we (possibly) cannot do efficiently, we should know what we can do efficiently by known methods such as linear programming.

Many computational problems now have improved solutions using machine learning techniques (ML). For example, search engines now use ML to improve the responses that you (personally) receive to a query.

Of course we will only be able to discuss a small subset of these ideas and yet

What great ideas have we might mised?

In Assignment A0, I asked you to rank your top 3 choices for a topic to be discussed. In addtion to any of the topics mentioned, feel free to add any topic not mentioned as part of your top three.

Given that different individuals have different preferences, how do we form a consensus? Forming consensus and making "fair" decisions are central topics in computational social choice.

Slides and guest speakers

I will have preliminary slides available before the week begins and then repost sometime at the end of the week (correcting typos, adding comments from the discussions, etc.)

NOTE: My slides will only be an outline of our discussions. **And we will often "wander" (i.e., take tangents) as we discuss ideas.**

I have lined up some guest speakers to lead discussions on some recent "great ideas". Here is the tentative schedule for these guest presentations:

- Rahul Krishnan (ML and medical diagnosis) October 7
- David Lindell (computational vision) October 12
- Daniel Wigdor (HCI) TBA (zoom class; I will post link)

Our first great idea

Our first topic/great idea for discussion: the von Neumann architecture

- By the mid 40s the first computers were being built. (I am not going to talk say about Babbage and the recent implementation of Babbages 1850 machine.)
- The earliest computers had fixed programs and were not general purpose machines.
- The stored program computer. The conceptual idea is associated with von Neumann who first described the model in a paper "First Draft of a Report on the EDVAC" dated June 30, 1945.
- It is clear that the stored program idea is present in Turing's 1936 paper which we will discuss later. But the Turing model was not meant to be a model of an actual computer.

The von Neumann architecture and digitilization

- The basic organization consists of 4 units: memory, I/O, ALU, control. (Some would say that a "bus" to cary signals between units is a fifth component but most say 4 units.) The memory is organized into a list or array of "words", each with its own address.
- Each word w_i (with say address i) is some fixed length string of bits bits ; i.e., $w = b_{n-1}b_{n-2}, \ldots, b_0$ which as an integer represents $\sum_{j=0}^{n-1} b_j 2^j$. In order to represent negative numbers, we can reserve the high order bit as a sign bit. Thus $b_{n-1}b_{n-2}\ldots b_0$ respresents $(-1)^{b_{n-1}} \sum_{j=0}^{n-2} b_j 2^j$. That is, $b_{n-1} = 1$ specifies a negative number. (The choice of 2 is not essential mathematically.). It is also possible to think of a word as a string of "bytes" in which each byte can also addressable but we will ignore that. Why not unary, decimal, or some other representation?

The von Neumann architecture and digitilization

- The basic organization consists of 4 units: memory, I/O, ALU, control. (Some would say that a "bus" to cary signals between units is a fifth component but most say 4 units.) The memory is organized into a list or array of "words", each with its own address.
- Each word w_i (with say address i) is some fixed length string of bits bits ; i.e., $w = b_{n-1}b_{n-2},\ldots,b_0$ which as an integer represents $\sum_{j=0}^{n-1} b_j 2^j$. In order to represent negative numbers, we can reserve the high order bit as a sign bit. Thus $b_{n-1}b_{n-2}\ldots b_0$ respresents $(-1)^{b_{n-1}} \sum_{j=0}^{n-2} b_j 2^j$. That is, $b_{n-1} = 1$ specifies a negative number. (The choice of 2 is not essential mathematically.). It is also possible to think of a word as a string of "bytes" in which each byte can also addressable but we will ignore that. Why not unary, decimal, or some other representation?
- But before we go any further we need to step back and talk about *digitalization and encoding*. That is, what are we storing in these words?

The von Neumann architecture and digitilization

- The basic organization consists of 4 units: memory, I/O, ALU, control. (Some would say that a "bus" to cary signals between units is a fifth component but most say 4 units.) The memory is organized into a list or array of "words", each with its own address.
- Each word w_i (with say address i) is some fixed length string of bits bits ; i.e., $w = b_{n-1}b_{n-2}, \ldots, b_0$ which as an integer represents $\sum_{j=0}^{n-1} b_j 2^j$. In order to represent negative numbers, we can reserve the high order bit as a sign bit. Thus $b_{n-1}b_{n-2}\ldots b_0$ respresents $(-1)^{b_{n-1}} \sum_{j=0}^{n-2} b_j 2^j$. That is, $b_{n-1} = 1$ specifies a negative number. (The choice of 2 is not essential mathematically.). It is also possible to think of a word as a string of "bytes" in which each byte can also addressable but we will ignore that. Why not unary, decimal, or some other representation?
- But before we go any further we need to step back and talk about *digitalization and encoding*. That is, what are we storing in these words? Data and instructions (as part of a program)

Digital computing vs the continuous real world

- The concept of digitalization is one of the most profound concepts that underlies science today.

Digital computing vs the continuous real world

- The concept of digitalization is one of the most profound concepts that underlies science today.
- In the "real world", space and time are continuous A typical computing application might wish to study the trajectory or movement of an object (person, ball, missle, etc). An object is moving through continuous 3 dimensional space in continuous time according to some (hopefully) known laws of motion.
- In contrast, digital computers use discrete finitely represented data and instructions.

Digital computing vs the continuous real world

- The concept of digitalization is one of the most profound concepts that underlies science today.
- In the "real world", space and time are continuous A typical computing application might wish to study the trajectory or movement of an object (person, ball, missle, etc). An object is moving through continuous 3 dimensional space in continuous time according to some (hopefully) known laws of motion.
- In contrast, digital computers use discrete finitely represented data and instructions.

  However, a real value x can be approximated by a rational number. Every integer has a finite representation and every rational can be represented by a pair of integers and hence and hence has a finite reperesentation.
- Furthermore, we will see how we can *approximate* any real number (within some reasonably large range) using *floating point representation*.

Digital computing vs the real world continued

- Furthermore there are some stochastic aspects (such as wind and air polution in the trajectory example) in real world processes.
- In contrast, digital computers are deterministic; that is, each “state” of the computer is precisely determined from the previous state.

Digital computing vs the real world continued

- Furthermore there are some stochastic aspects (such as wind and air polution in the trajectory example) in real world processes.
- In contrast, digital computers are deterministic; that is, each "state" of the computer is precisely determined from the previous state.

  However, we can simulate stochastic events by drawing random variables from some (pseudo) random distribution.
- Stochastic processes can then be simulated by a discrete deterministic machine. Of course, the conclusions to be made from simulations relative to a specific outcome in the real world will depend on how well we have modelled the system and how unpredictable is the system.
- The early debate: relative benefits of digital computation vs that of analogue computation. (In the past, thermostats and other control devices were essentially simple analogue computers.) Why did digitlilization win the debate?