Great Ideas in Computing

University of Toronto CSC196
Fall 2025

Class 20: November 14 (2025)

Announcements and Agenda

Announcements

- I have posted the first two questions for Assignment 4. Assignment 4 is due Wednesday, November 26 at 3PM.
- Our fourth and final guest presenter will be Jonathan Panuelos who will discuss "computational mechanics" which relates to the simulation of physical systems. His presentation is this coming Wednesday, November 19.
- The second and final quiz will be held Friday, November 28 in the usual tutorial room BA 2139.

Today's Agenda

- Finish complexity based cryptography
- Start Social networks

What computational facts do we need to know for public key cryptography?

1. The extended Euclidean algorithm can efficiently compute an s and t such that $sa + tb = gcd(a, b)$
2. a^k mod N can be computed efficiently for any a, k, N.
3. We can efficiently determine if a number p is prime.

In practice, the public key (i.e. the exponent e) is chosen to be reasonably small so that encryption can be made more efficient.

Note that we have been assuming that an adversary EVE (i.e., is just eavesdropping) and not changing messages. That is, EVE just wants to learn the message or something about the message. If EVE could change messages then EVE could pretend to be BOB. So one needs some sort of a public key infrastructure.

If EVE knows that the message M was one a few possibilities, then EVE can try each of the possibilities; that is compute $M^e mod N$ for each possible M to see what message was being sent. So we pad the plain text M with random bits so that it is infeasible to just guess the plain text M.

WARNING: Real world cryptography is sophisticated

Complexity based cryptography requires careful consideration of the definitions and what precise assumptions are being made.

Complexity based cryptography has led to many important practical protocols and that are provably secure under stated complexity assumptions. Fortunatley, many complexity assumptions turn out to be equivalent.

In the Rackoff notes, the following theorem is stated as the fundamental theorem of cryptography. (To make this result precise, one needs precise definitions which we are omitting.)

Theorem: The following are equivalent:

- It is possible to do "computationally secure sssions"
- There exists pseudo-random generators; that is, create strings that computationally look random)
- There exist one way functions f; that is functions such that $f(x)$ is easy to compute but given $f(x)$ it is hard to find a z such that $f(z) = f(x)$
- There exist computationally secure digital signature schemes.

Can we break RSA today with quantum computers?

During the last class Olatunde indicated that RSA has been broken. I very much appreciate that he sent me two articles that seemed to say that RSA has been broken or will soon be broken using quantum computers.

One article says: New research shows that RSA-2048 encryption could be cracked using a one-million-qubit system by 2030, 20x faster than previous estimates. Here's what it means for enterprise security.
A quantum computer with one million noisy qubits running for one week can theoretically crack RSA-2048 bit encryption, representing twenty times fewer qubits than Google's 2019 estimate, according to new research from Google Quantum AI. Note the qualifications: "could be cracked" and "can theoretically crack" RSA with the size of the quantum computer that "they hope" will exist by 2030.

The other artcile says that researchers have broken RSA.
But then the article points out that Although they used a quantum computer to decrypt an RSA encryption, they used only a 50-bit integer for the RSA encryption. .... But most modern encryption technologies now use 1024- to 2048-bit integers.

Social Networks

Online social networks are a great idea in the sense of allowing friends to stay in touch and making new freinds. It is also a great idea in the sense of it being a "killer application". Like email and search engines, online social networks have helped make personal computing pervasive.

But like many great ideas, there are negative consequences. Being able to identify communities allows various entities to target groups with biased, misleaading or false information to reinforce the beliefs in an online community. It allows rumours to become "facts" widely believed by those in the community.

The main focus in our discussion of social networks will be how graph structure can reveal personal and individual information as well as communities.

What is a social network?

A social network is a network $G = (V, E)$ where the nodes in V are people or organizations. Social networks can be undirected or directed networks.

The edges can be relations between people (e.g. friendship) or membership of an individual in an organization.

Social networks can be of any size (e.g., a small network like the Karate Club we will soon show) or enormous networks like Facebook and Twitter (X). We usually think of Facebook as an undirected graph (where *friendship* is an undirected edge) and Twitter as a directed graph (i.e., where *follows* is a directed edge).

Understanding how networks evolve, the resulting structure of social networks, and computational aspects for dealing with large networks is an active field of study in CS as well as in sociology, political science, economics, epidemiology, and any field that studies human behaviour. J. Kleinberg's 2000 analysis with regard to the six degrees of separation phenomena is an early result that sparked interest in the algorithmic aspects of social networks.

The computational challenge presented by super large networks

The size of some modern networks such as the web and social networks such as Facebook are at an unprecedented scale.

As of Februay, 2023, Facebook has roughly 3 billion monthly active users and 2 billion daily active users worldwide. India has the most Facebook Users (roughly 330 million). The US has 180 million users. The average facebook user has about 155 friends which then implies about $2.9 \cdot \frac{155}{2} \approx 200$ billion edges. It is interesting to note that 90% of daily active users are outside USA and Canada. See
https://www.omnicoreagency.com/facebook-statistics/
for lots of interesting demographic and other facts about Facebook.

What does this imply for the complexity of algorithms involving such super large networks?

Linear is the new exponential

In complexity theory, we say (as an abstraction that polynomial time algorithms are "efficient" and "exponential time" is infeasible. There are, of course, exceptions but as an abstraction this has led to invaluable fundamental insights.

As problem instances have grown, there was a common saying that "quadratic (time) is the new exponential".

But with the emergence of networks such as the web graph and the Facebook network, we might now say that "linear is the new exponential" when it comes to extracting even the most basic facts about these networks. For example, how do we even estimate the average node degree? in a giant network?

Linear is the new exponential

In complexity theory, we say (as an abstraction that polynomial time algorithms are "efficient" and "exponential time" is infeasible. There are, of course, exceptions but as an abstraction this has led to invaluable fundamental insights.

As problem instances have grown, there was a common saying that "quadratic (time) is the new exponential".

But with the emergence of networks such as the web graph and the Facebook network, we might now say that "linear is the new exponential" when it comes to extracting even the most basic facts about these networks. For example, how do we even estimate the average node degree? in a giant network? We sample.

There are many facts about large networks that we would like to extract from the network. For example, how do we find "influential" or "interesting nodes" in a social network?

Sublinear time algorithms

What is sublinear time?

In general when we measure complexity, we do so as a funtion of the input/output size. For graphs $G = (V, E)$, the size of the input is usually the number of edges E. (An exception is that when the graph is presented say as an adjacency matrix, the size is n^2 where $n = |V|$.)

Since our interest is in massive information and social networks, we consider sparse graphs (e.g. average constant degree) so that $|E| = O(|V|)$ and hence we will mean sublinear time as a function of n. The desired goal will be time bounds of the form $O(n^\alpha)$ with $\alpha < 1$ and in some cases maybe even $O(\log n)$ or $polylog(n)$.

Given that optimal algorithms for almost any graph property will depend on the entire graph, we will have to settle for approximations to an optimum solution. Furthermore, we will need to sample the graph so as to avoid having to consider all nodes and edges. And we will need a way to efficiently access these massive graphs,

Coping with massive social graphs continued

One way to help coping with massive networks is to hope to utilize some substantial amount of parallelism. There is an area of current research concerning massive parallel computation (MPC) models where (in principle) we can achieve sublinear time by distributing computation amongst a large (i.e., conceptually a non constant) number of processors.

But even if we could muster and organize thousands of machines, we will still need random samplng, approximation, and have highly efficient "local information algorithms" (e.g., where say each processor is responsible for some nodes and learns about its local neighbourhood).

Finally, in addition to random sampling and parallelism, we will have to hope that social networks have some nice structural properties that can be exploited to as to avoid complexity barriers that exist for arbitrary (even sparse) graphs. These complexity barriers are hopefully clear from our discussion of complexity theory, *NP completeness* and *NP hardness*.