

Great Ideas in Computing

Allan Borodin

University of Toronto CSC196
Fall 2026

Class 10: October 14 (2026)

Announcements and Todays Agenda

Announcements

- On Wednesday, October 21, Professor Tom Ristenpart will give a guest presentation on computer security.
- Assignment 2 is due October 21.
- The first term test will take place October 23 during the usual Friday tutorial time and place.

Today Agenda

- Killer Applications as a Great Idea.
- Social Networks as a killer application.
- The computational and mathematical aspects of networks/graphs.

What are some killer applications and why is a killer application a great idea

I view a killer application (in computing) as an application that becomes available to the general public and becomes widely used.

It is rather arbitrary but note that I am not listing the internet as a killer application as I view it as an idea and a technology that makes applications possible for the general public. I also note that some killer applications originate in military applications (e.g., navigations systems).

What would you list as killer applications?

What are some killer applications and why is a killer application a great idea

I view a killer application (in computing) as an application that becomes available to the general public and becomes widely used.

It is rather arbitrary but note that I am not listing the internet as a killer application as I view it as an idea and a technology that makes applications possible for the general public. I also note that some killer applications originate in military applications (e.g., navigations systems).

What would you list as killer applications?

Here is a list mainly by chronological order.

- Word processors
- Computer games
- Email
- Search engines
- Navigation systems
- Wikipedia
- Social Networks
- LLMs

Why do I consider a killer application as great idea?

I am not accounting for (but will just note) the possible misuse of any application or technology; Even an application like email can have significant negative aspects. For example, email can and does make fraud easier, social networks can and do further bias.

As each of these applications became more popular, companies can sell more hardware and software and more sales results in more investment and great advances in technology which leads to more applications..

I would say that these applications (coupled with the great advances in computer hardware, software tools and advances in communication) made the use of computing pervasive and thus computers have become a commodity (like a telephone); that is, computers became indispensable. And while the costs for computers have decreased (especially relative to their computational power), our costs/month have probably increased. **Is that your experience?**

I will discuss two of these killer applications; namely, social networks and search engines.

Social Networks

Confession: I have accounts on Facebook and Twitter/X but I haven't used either for quite a while. Family members tell me when I need to know something.

Even though I am not much of a user, I know that online social networks are a great idea in the sense of allowing friends and family to stay in touch, and making new friends. And, as I am emphasizing, it is also a great idea in the sense of it being a “killer application”.

As I have said, like other great ideas, there can be negative consequences. Being able to identify communities allows various entities to target groups with biased, misleading or false information to reinforce the beliefs in an online community. It allows rumours and lies 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. (Social networks allows us to discuss graphs concepts and results.)

What is a social network?

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

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

Social networks can be of any size (in terms of $|V|$ and $|E|$). Some networks are enormous, 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. Jon Kleinberg's 2000 analysis with regard to the *six degrees of separation* phenomena is an early result that sparked CS 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 world wide web (WWW) and social networks such as Facebook are at an unprecedented scale.

In May, 2006, Facebook had 3.1 billion monthly active users and 2.11 billion daily active users worldwide. India has the most Facebook Users (581.6 million). The US has 2.54 million users. The average facebook user has about 155 friends which then implies about $2.9 \cdot \frac{155}{2} \approx 200$ billion edges. See <https://www.omnicoreagency.com/facebook-statistics> and <https://www.demandsage.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. (Recall the table of time bounds we showed in the L6 slides.)

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 or the diameter** in a giant network? The diameter of a connected graph is the maximum (over all $u, v \in V$) of the shortest path between u and v).

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. (Recall the table of time bounds we showed in the L6 slides.)

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 or the diameter** in a giant network? The diameter of a connected graph is the maximum (over all $u, v \in V$) of the shortest path between u and v). 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” nodes?

Sublinear time algorithms

What is sublinear time?

In general when we measure complexity, we do so as a function 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 (or equivalently of $m = |E|$). 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 $\text{polylog}(n)$.

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 sampling, 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. We will discuss these complexity barriers later when we discuss complexity theory and, in particular, *NP completeness* and *NP hardness*.