Unit 1 (Intro) Textbook (Ch 0)

Saturday, October 5, 2019 6:53 PM

Section 0.1 The Role of Algorithms

- An *Algorithm* is a set of steps that defines how a task is performed.
  - Examples include cooking recipes, travel directions.
- A representation of an algorithm is called a *program*.
- Computer programs are usually printed on paper or displayed on screens.
- Programs are developed in a manner compatible with the machine.
- The process of developing a program, encoding it to be machine-compatible, and inserting it into a machine is called *Programming*.
  - Programs are referred to as *software*, in contrast to the machinery itself, known as *Hardware*.
- The study of algorithms began in mathematics.
  - Mathematicians used algorithms long before computers existed.
    - One of the best known of these algorithms is used for long division of two multi-digit numbers.
    - Another example is the Euclidean Algorithm, developed in Ancient Greece, for finding the greatest common divisor of two positive integers.
- Once an Algorithm for performing a task has been found, the performance of that task is reduced to merely following directions, without having to understand why the algorithm works.
- *Incompleteness Theorem*: in any mathematical theory encompassing our traditional arithmetic system, there are statements that cannot be proven true or false by algorithms. This theorem began the field that today is known as Computer Science.

Section 0.2 The History of Computing

- One of the earliest computing devices was the abacus.
  - It consists of beads strung on rods, which can be moved back and forth to represent different values.
  - This makes the abacus merely a data storage system, as it relies on a human operator.
- Between the Middle Ages and the modern era, a few inventors, such as Blaise Pascal, Gottfried Wilhelm Leibniz, and Charles Babbage, began to experiment with the technology of gears. These machines represented data through gear positioning.
  - Output from Pascal and Leibniz's machines had to be observed and transcribed by hand. Babbage, however, envisioned machines that could print results of computations on paper to avoid transcription errors.
  - As for algorithms, Pascal's machine could only do simple addition, and Leibniz's machine had its algorithms firmly embedded in its design.
  - Babbage's Difference Engine could be modified to perform different calculations, but his Analytical Engine was programmable using holes in paper cards.
    - In fact, Augusta Ada Byron (Ada Lovelace), who demonstrated how to use the Analytical Engine to perform different calculations, is often considered the world's first programmer.
    - However, the concept of using holes in paper was not originated by Babbage, but rather by Joseph Jacquard, who made a weaving loom in which the steps used in the process used to weave was determined by holes in wood or cardboard.
    - Jacquard's idea was also used by Herman Hollerith to speed up the tabulation process in the 1890 US Census.
    - These later became known as punch cards and remained a popular method of programming computers well into the 1970s.
- 19th Century Technology was unable to reproduce the gear-based machines cost-effectively. But with the advent of electronics in the early 1900s, this barrier was overcome.

    - Early examples of this include Stibitz's electromechanical machine (1940) and the Mark I (1944).
        - These, however, were quickly made obsolete, as other researchers applied the technology of vacuum tubes to construct fully electronic computers.
    - The first of the vacuum tube machines was the Atanasoff-Berry machine, built between 1937 and 1941.
    - Other examples include the Colossus, used to decode German messages during World War II, as well as later ENIAC, developed by John Mauchly and J. Presper Eckert at the Moore School of Engineering, University of Pennsylvania.
- After this, the history of computing has been closely linked to advancing technology, such as the invention of transistors, and then the development of complete single-unit, or integrated circuits. With these developments, the room-sized machines of the 1940s now only were the size of a cabinet.
    - Many of the components within a computer became readily available as integrated circuits called chips.
- A major step toward popularizing computing was the development of desktop computers, which began with hobbyists who built homemade computers from combinations of chips.
    - It was within this hobby that Steve Jobs and Stephen Wozniak built an inexpensive home computer and in 1976 founded Apple to manufacture and market their products.
    - Other companies that marketed similar products were Commodore, Heathkit, and Radio Shack. However, businesses continued to rely on IBM and its mainframe computers.
    - In 1981, IBM released its first desktop computer called the PC, whose underlying software was developed by the recently founded company Microsoft.
        - The PC was a hit, and established the desktop computer in the minds of the business community.
        - Today, the term PC is widely used to refer to various machines whose design has evolved from IBM's initial model, which still continue to be marketed with software mainly from Microsoft. PC is often used interchangeably with the terms desktop or laptop.
- Near the end of the century, communication was revolutionized by a system called the Internet that connected computers around the world.
    - Within the Internet, British scientist Tim Berners-Lee proposed a system in which documents stored on internet-connected computers could be linked together, called the World Wide Web.
        - To make this information more accessible, software systems called search engines were developed to sift through the web, categorize their findings, and use the results to assist users in finding particular topics. Major players in this field include Google, Yahoo, and Microsoft.
- Simultaneously, the miniaturization of computers continued. Today, tiny computers are embedded within a wide variety of devices, including automobiles with GPS or Hands-free functionality.
    - However, perhaps the most important application of computer miniaturization is found in smartphones, general-purpose computers within which telephony is one of many applications. More powerful than the supercomputers of prior decades, these devices are equipped with a rich variety of sensors and interfaces.

Section 0.3 An Outline of Our Study

- The textbook begins by discussing hands-on topics such as hardware and leading into more abstract topics such as algorithm complexity and computability.
    - Therefore, our study follows a pattern of larger and larger abstract tools.
- We begin by covering how data is encoded and stored within modern computers, then cover the basic internal operation of a single computer.
    - The general idea of these chapters are technology independent, and will remain relevant regardless of the future of technology.
- Chapter 3 covers the software that controls the overall operation of the computer, called the operating system.
- Chapter 4 studies how computers are connected to each other to form networks and how networks are connected to form the internet. This includes network protocols, the Internet's structure, the world wide web, and security.

- Chapter 5 explores algorithms more formally, including how they are discovered, identifying fundamental algorithmic structures, develop techniques for representing algorithms, and introduce concepts such as algorithm efficiency and correctness.
- Chapter 6 considers the subject of algorithm representation and the programming process, learning about different programming paradigms and their languages.
- Chapter 7 introduces software engineering, which deals with the problems encountered when developing large software systems.
- Chapter 8 and 9 look at ways in which data can be organized in a computer system, data abstractions (Ch 8) and database systems (Ch 9)
- Chapter 10 explores the subject of graphics and animation, which deals with creating and photographing virtual worlds.
- Chapter 11, Artificial Intelligence, we learn that to develop more useful machines CS has turned to the study of human intelligence for insight, and attempts to mimic human processes and transfer them to machines.
- Chapter 12 closes our study by investigating the theoretical foundations of Computer Science, which helps us to understand the limits of algorithms, and thus machines.

Section 0.4 The Overarching Themes of Computer Science

There are seven big ideas, or overarching themes, of Computing:

- Algorithms
- Abstraction
- Creativity
- Data
- Programming
- Internet
- Impact

Algorithms

- Computer science is, simply put, the science of algorithms.
  - The scope of CS draws from Mathematics, Engineering, Psychology, Biology, Business Administration, and Linguistics.
  - Researchers in different branches of CS may have very distinct definitions of the science.
- Questions:
  - Which problems can be solved by algorithmic processes?
  - How can the discovery of algorithms be made easier?
  - How can the techniques of representing and communicating algorithms be improved?
  - How can algorithms be used to manipulate information?
  - How can algorithms be applied to produce intelligent behavior?
  - How does the application of algorithms affect society?

Abstraction

- *Abstraction* refers to the distinction between the external properties of an entity and the details of the entity's internal composition.
- It is an abstraction that allows us to ignore the internal details of a device such as a computer and use it as a single, comprehensible unit.
- It is by means of abstraction that such complex systems are designed and manufactured in the first place. Computers and other devices are constructed from components, each of which represents a level of abstraction.
- It is by applying abstraction that we are able to construct, analyze, and manage large, complex, computer systems that would be overwhelming if viewed in their entirety at a detailed level.
- At each level of abstraction, we view the system in terms of components, called abstract tools, whose details we ignore. This allows us to concentrate on how each component interacts with other components at the same level and how they form a higher-level component.
- Abstraction is not limited to science and technology.

Creativity

- The field of computer science is an inherently creative one.
  - Discovering and applying new algorithms is a human activity that depends on our innate desire to solve problems in the world around us.
  - CS not only extends forms off expression spanning the visual, language, and musical arts, but also enables new modes of digital expression that pervade the modern world.
  - Creating large software systems is more like making a sculpture than following a cookbook recipe. Envisioning it requires careful planning, and building it requires time, attention to detail, and practiced skill.

Programming

- Translating human intentions into executable computer algorithms is now broadly referred to as programming, although today's tools bear little resemblance to those of the 50s and 60s.
- While CS consists of much more than programming, the ability to solve problems by devising executable algorithms is a foundational skill for all Computer Scientists.
- Hardware is only capable of executing simple algorithmic steps, but the abstractions provided by programming languages allow humans to reason about and encode solutions for more complex problems.
- Questions:
  - How are programs built?
  - What kinds of errors can occur in programs?
  - How are errors in programs found and repaired?
  - What are the effects of errors in modern programs?
  - How are programs documented and evaluated:

Internet

- The Internet connects devices around the world and has had a profound impact on the way society stores, retrieves, and shares information.
- Commerce, news, entertainment, and communication now increasingly depend on this interconnected web of smaller networks.
- The reach of the internet also poses risks to privacy and personal information, making cryptography and cybersecurity become of growing importance in our world.

Impact

- CS also has enormous impacts on society, blurring many distinctions on which our society has based decisions in the past and is challenging many of society's long-held principles.
- In Law, CS also generates questions about the degree to which intellectual property can be owned and the rights and liabilities that accompany ownership.
- In ethics, it generates numerous options that challenge the traditional principle on which social behavior is based.
- In government, in generates debates regarding the extent to which computer technology and its applications should be regulated.
- In philosophy, it generates contention between the presence of intelligent behavior and intelligence itself.