Introduction to Computational and Algorithmic Thinking

LECTURE 1 – COMPUTATIONAL THINKING

1

Announcements

This lecture: Computational Thinking

Reading: Read Chapter 1 of Conery

Acknowledgement: These slides are extended version of slides prepared by Prof. Arthur Lee, Prof. Tony Mione for earlier CSE101 classes.

2

What is Computer Science

- **Computer science** is all about using computers and computing technology to solve challenging, real-world problems in science, medicine, business and society
- Computer science = computer programming 🡐 Not true
  - Computer programming is an important aspect of computer science
  - Computer programs often provide (parts of) the solutions to challenging technological problems
- Computer science is also not:
  - computer literacy
  - computer maintenance/repair

3

A modern computing problem

ELECTRONIC HEALTH RECORDS ARE BECOMING INCREASINGLY IMPORTANT AS TIME GOES ON

4

A modern computing problem

- Electronic health records are becoming increasingly important as time goes on
- Consider issues (technical + others) that arise providing a hw/sw system to medical professionals and others who need access to digital medical records:
  - What data will be stored? How? In what format?
  - How will the data be accessed and displayed?
  - Who will have access? How will the data be secured?
  - How will the data be backed up and preserved?
- Answering these questions requires **computational thinking**

5

Computing systems

Let's take a tour of modern computing systems

A computing system consists of two major parts: the **hardware** and the **software**

Some hardware elements of a computer:
- Screen, keyboard, mouse
- Central processing unit (CPU), main memory
- Hard drives and other storage units

Types of software:
- Applications software, like office productivity programs, video games, web browsers
- Systems software, like operating systems, database systems

6

Computing systems

Can hardware exist without software?

- Sure, but is it useful? ➔ It depends
  - General Purpose CPUs [Intel x86, core i5, i7, etc.] Not really
  - Specialized hardware [FPGAs, ASICs with functionality built into hardware] – Yes. Although functionality was developed with software based tools

Can software exist without hardware?

- In a literal sense, no – hardware is needed to execute software
- But, the underlying problem-solving techniques employed by the programmer to create the software do exist separately from the hardware and software

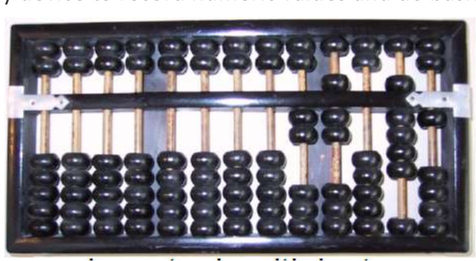
One more part to a computer system: data

- The software needs some kind of data to process: numbers, text, images, sound, video

7

A quick history of computing

- We think of computers as modern inventions
- Computing devices
  - go back thousands of years
  - have many of the same basic features of digital computers
- **Abacus** – an early device to record numeric values and do basic arithmetic (16th century B.C.)



- https://www.youtube.com/watch?v=GF6nCmcQ5es
- What does an abacus have to do with laptops, smartphones and tablet computers???

8

A quick history of computing

Modern computers borrow four important concepts from the abacus:

1. Storage
2. Data Representation
3. Calculation
4. User Interface

A quick history of computing

1. Storage
   - An abacus stores numbers, which are the most fundamental type of data in modern computing.
   - In a modern computer, all data – text, images, audio, video – is represented using binary numbers (1s and 0s)

2. Data Representation
   - The abacus represents numbers using beads on spindles.
   - Modern computers employ a variety of techniques for representing data on storage media:
     - Magnetic
     - Optical
     - Electrical

A quick history of computing

3. Calculation
   - By moving beads on abacus spindles, user can perform addition, subtraction, multiplication, and division
   - Modern computers contain powerful central processing units that perform calculations at astonishing speeds

4. User Interface
   - The beads and spindles on the abacus
   - Modern computers provide a wide variety of input and output devices for the user

A quick history of computing

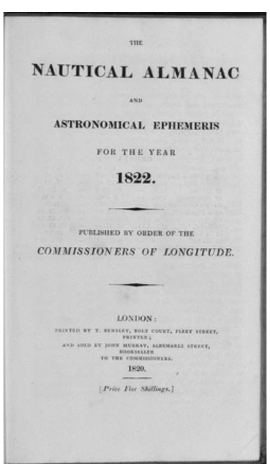
- In the 17th century people began tinkering with physical devices that could do computations and calculations
- Blaise Pascal
  - the French mathematician and philosopher
  - one of a few to design and build a physical calculator
- Calculator could only do addition and subtraction
  - Input is given using dials
  - Output is read on small windows above each dial

Programmable devices

- Pascal's calculator and other similar devices of that time were not programmable
- One of the first programmable devices in history was a loom
- Joseph Marie Jacquard's loom (1804) could be programmed by feeding in a set of punched cards
- This is not all that different from quitting a program that's running on your computer and starting another one!

- https://www.youtube.com/watch?v=MQzpLLhN0fY

Rise of Analytical Engine



THE

NAUTICAL ALMANAC

AND

ASTRONOMICAL EPHEMERIS

FOR THE YEAR

1822.

PUBLISHED BY ORDER OF THE

COMMISSIONERS OF LONGITUDE.

LONDON:

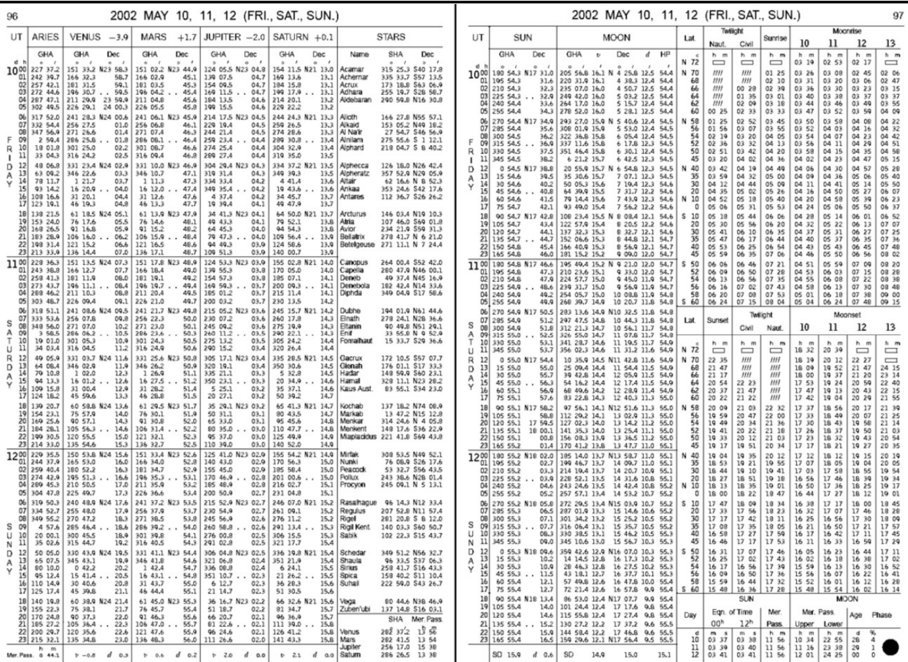
PRINTED BY T. BENSLEY, BOLT COURT, FLEET STREET, PRINTER;

AND SOLD BY JOHN MURRAY, ALBEMARLE STREET, BOOKSELLER TO THE COMMISSIONERS.

1820.

[*Price Five Shillings.*]

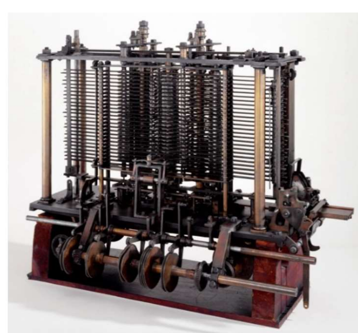
- Summer of 1821 – Mathematician Charles Babbage and astronomer John Herschel were working on creating a book of mathematical tables.
- Almanac contains tables denoting positions of the Moons, planets and stars – which are used by navigators to determine location at the sea.
- Manual work caused a number of errors.
- Babbage showed his frustration with the large number of errors by exclaiming, "I wish to God these calculations had been executed by steam!"
- What made Babbage think steam engines could help him solve mathematical problems?



15

Programmable devices

- Charles Babbage designed the Analytical Engine, a mechanical, programmable computer in 19th Century
  - It was never built in Babbage's time due to a lack of manufacturing capabilities (ahead of his time!)
  - Design called for punched cards to be fed into the machine to program it to perform mathematical calculations
  - Output would go to a printer or punched cards
  - See for details: https://www.thoughtco.com/first-computer-charles-babbages-1221836



16

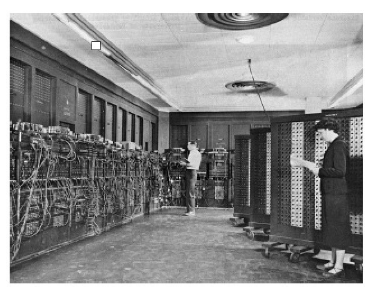
Programmable computers

- Many others
- Now, move forward to the 20th and 21st centuries
- A modern computer has three basic requirements:
  1. Must be electronic and not exclusively mechanical.
  2. Must be digital, not analog
     - Uses discrete values (digits), not a continuous range of values to represent data. (i.e. digital vs mercury-based thermometer)
  3. Must employ the **stored-program concept**
     - the device can be reprogrammed by changing the instructions stored in the memory of the computer

17

Programmable computers

- ENIAC (Electronic Numerical Integrator and Computer)
  - Built in the 1940s
  - Among the first computers to employ the stored-program concept
- A modern computer has four major kinds of components:
  - Input device(s) – examples?
  - Output device(s) – examples?
  - Memory – for data storage, both temporary & permanent
  - Processor – for doing computations



18

Programmable computers

- Again, the **stored-program concept** is the idea that programs (software) along with their data are *stored* (saved) in the memory of a computer
  - Not referring to storage on hard drives, flash drives or CDs
  - Referring to **main memory** of the computer, sometimes called the **RAM** (random access memory)
- A modern processor
  - reads the **machine instructions** *stored* as 1s and 0s in the main memory
  - executes those instructions in sequence
  - Key point: these instructions can be changed to easily reprogram the computer to do new tasks

A typical processor has a thousand or more different machine instructions.

19

Transistors

- A variety of devices have been used to represent digits and to control the operation of computing machines
- In the 1940s:
  - Bardeen, Brattain, and Shockley invented the **transistor**, which is an electronic switch with no moving parts
- In the 1950s and 1960s:
  - Kilby, Noyce, and others used transistors to develop **integrated circuits**
  - Devised a way to manufacture thousands – later, millions and billions – of transistors on a single wafer of silicon
- A single **chip** contains:
  - an integrated circuit
  - a ceramic or plastic case
  - external pins to attach it to a **circuit board**

20

Transistors

Noyce and businessman Gordon Moore commercialized this technology by co-founding Intel Corporation in 1968

Manufacturing technologies improved in the 1950s and 1960s:

engineers were able to pack many more transistors per unit area on silicon wafers

Moore's law: Moore observed that the number of components within an integrated circuit was doubling every 18 months.

- The trend has continued pretty steadily since then.
- But transistors can be only so small!

Combating miniaturization challenges:

Intel, AMD (Advanced Micro Devices) and others now make processors that feature multiple processing **cores** that perform calculations in parallel with each other

21

Moore's Law

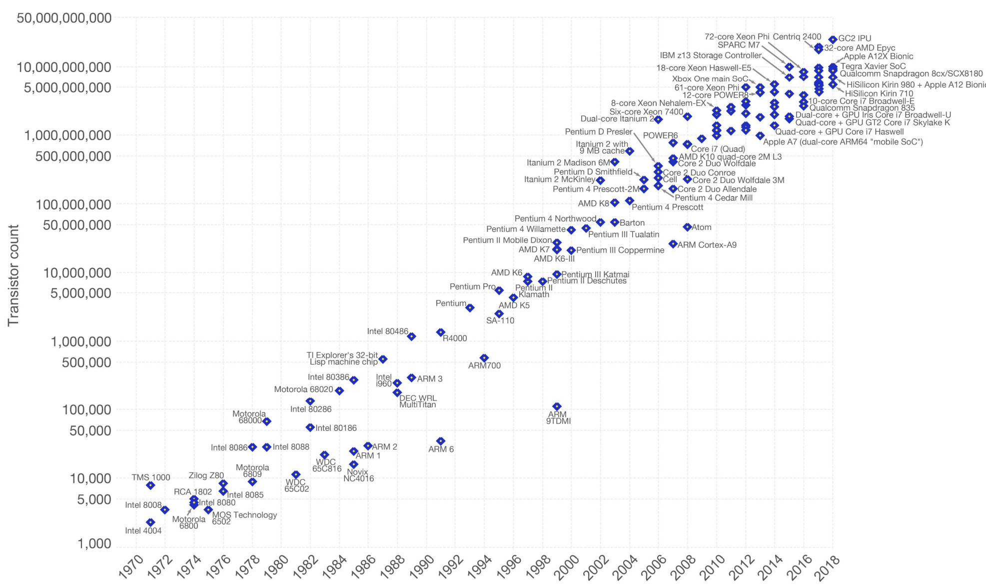
Figure taken from:

OurWorldinData

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.

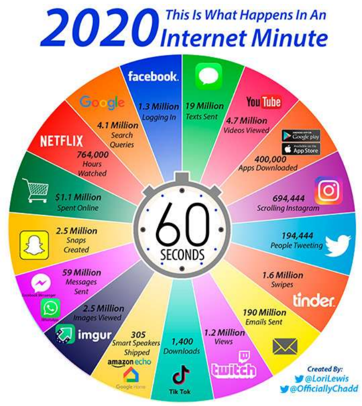
Our World in Data

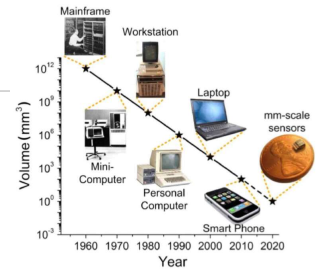


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

22

The information explosion in 21st century





https://www.internetlivestats.com/

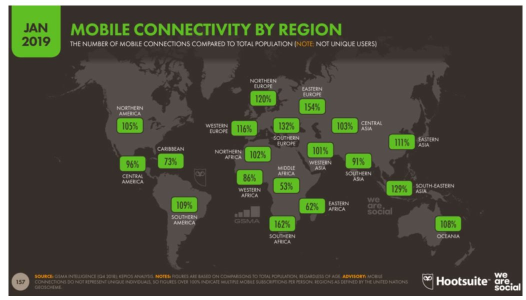


Figure 1 source: https://www.allaccess.com/merge/archive/31294/infographic-what-happens-in-an-internet-minute

Figure 2 source: https://wearesocial.com/global-digital-report-2019

23

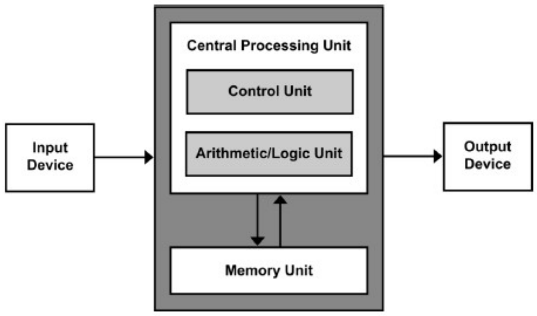
Computing systems

- A computing system consists of two major parts: the **hardware** and the **software**
- Some hardware elements of a computer
  - Screen, keyboard, mouse
  - Central processing unit, main memory
  - Hard drives and other storage units
- Type of software in use
  - Applications software, like office productivity programs, video games, web browsers
  - Systems software, like operating systems, database systems

24

Modern computer architecture

- The stored program approach used today is implemented using **von Neumann architecture**, named after U.S. mathematician John von Neumann
- This architecture contains input devices, output devices, a processor and a memory unit



- Will now look at how they work together to form a functioning computer

25

Modern computer architecture

- In modern computers (PCs), the major components in a von Neumann machine reside physically in a circuit board called the **motherboard**
  - The CPU, memory, expansion cards and other components are plugged into slots so they can be replaced
  - Hard drives, CD drives, and other storage devices are connected to the motherboard through cables
- The central processing unit is the “brain” of the machine
  - its **arithmetic/logic unit** (ALU) performs millions or billions of calculations per second
  - The **control unit** is the main organizing force of the computer and directs the operation of the ALU

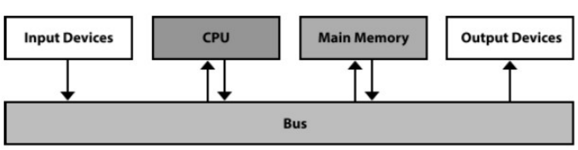
26

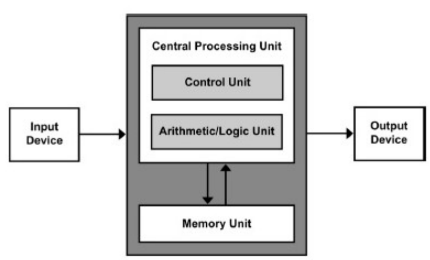
Modern computer architecture

27

Modern computer architecture

- The memory unit in this diagram refers to the main memory, not hard drives and other forms of **external storage**
- CPU, main memory, and I/O devices communicate over a shared set of wires known as the **system bus**





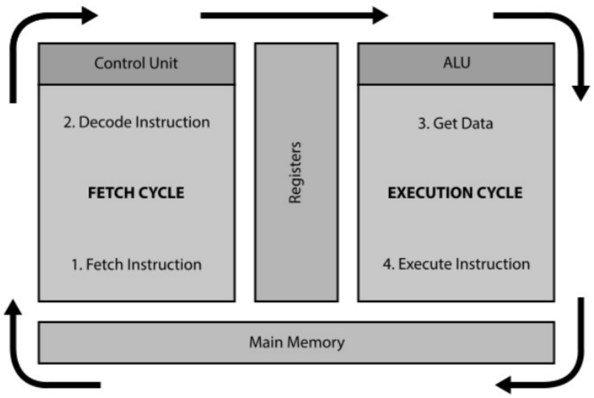
28

The fetch-decode-execute cycle

- The system bus carries electrical signals that encode machine instructions and data
  - The CPU **fetches** the instructions and data from memory as needed
  - The control unit **decodes** each instruction to figure out what it is (an addition, subtraction, whatever)
  - Data values (e.g., numbers to be added and their resultant sum) are stored temporarily in memory cells called **registers** within the CPU
  - The ALU **executes** the instruction, saving the result in the registers and main memory
- This whole process is known as the **fetch-decode-execute cycle**
- Illustration

https://chortle.ccsu.edu/java5/Notes/chap04/ch04_4.html

The fetch-decode-execute cycle



What about the software?

- Software consists of instructions for the CPU to execute
  - CPUs "understand" something called **machine language**, which consists of 0s and 1s
  - A single instruction for a modern computer might consist of some combination of 32 or 64 0s and 1s!
- Most programming now done using **high-level programming languages**, which consist of English and English-like words with some mathematical notation thrown in
- Will look into the basics of Python, a popular, easy-to-learn, high-level programming language

31

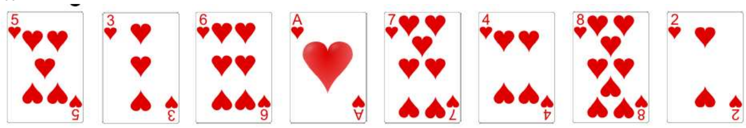
What is computational thinking?

- **Computational Thinking:**
  - How computer scientists think – how they reason and work through problems
- Computer science encompasses many sub-disciplines that support the goal of solving problems:
  - Computer theory areas ➔ these are the heart and soul of computer science
    - algorithms
    - data structures
  - Computer systems areas
    - hardware design
    - operating systems
    - networks
  - Computer software and applications
    - software engineering
    - programming languages
    - Databases
    - Simulation
    - artificial intelligence
    - computer graphics
- A major goal of this course is to help you develop your computational thinking and problem solving skills

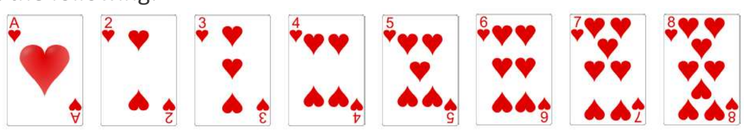
32

A classical problem: Sorting data

- Suppose we have a deck of cards we want to put in order
- **Sorting**: important problem – arises frequently in computer science
- Example: Use the Ace thru 8 of Hearts
- Given:



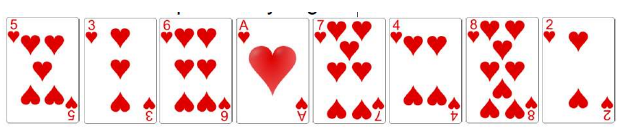
- Want the following:



33

A classical problem: Sorting data

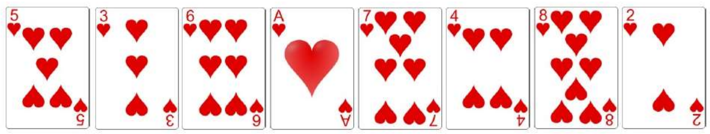
- You want to explain to a young child how to put the cards in order
  - What steps would you give?



34

A classical problem: Sorting data

- One sorting technique is called **selection sort**
- It repeatedly searches for and swaps cards in the list



35

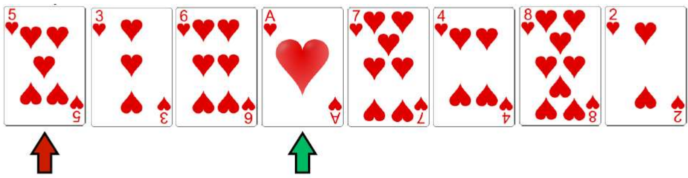
Selection sort

- First, find the smallest item and exchange it with the card in the first position

36

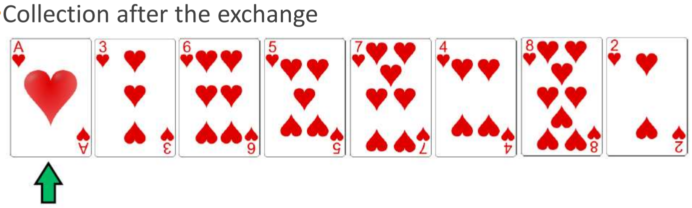
Selection sort

- Select the smallest item and exchange it with the card in the first position



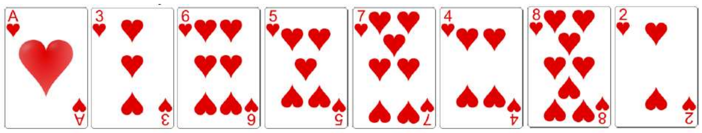
Selection sort

- Collection after the exchange



Selection sort

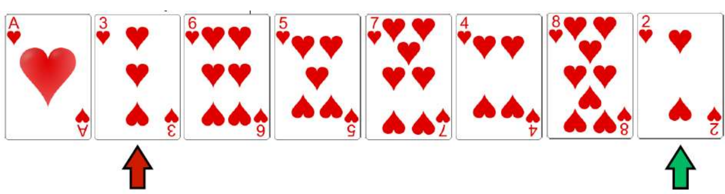
- Next, select the second-smallest item and exchange it with the card in the second position



39

Selection sort

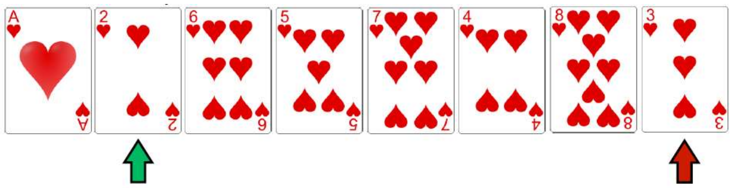
- Select the second-smallest item and exchange it with the card in the second position



40

Selection sort

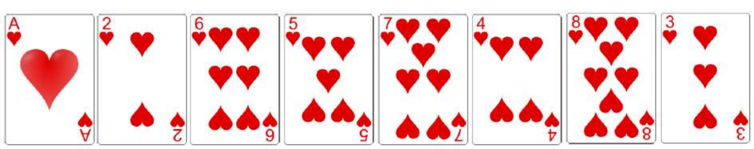
- Collection after the exchange



41

Selection sort

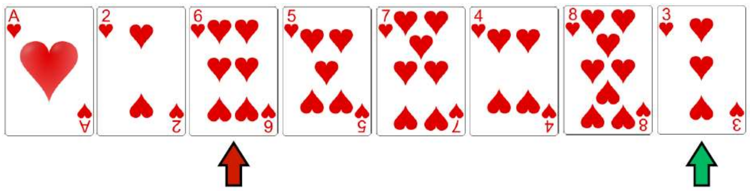
- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



42

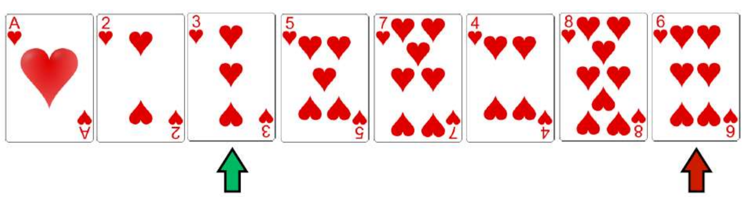
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



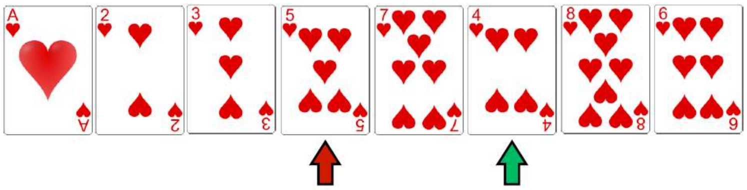
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



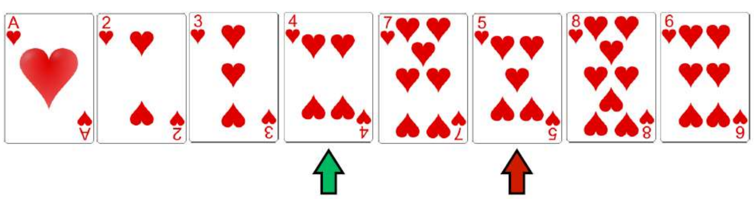
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



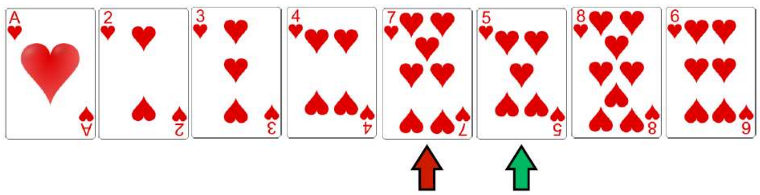
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



Selection sort

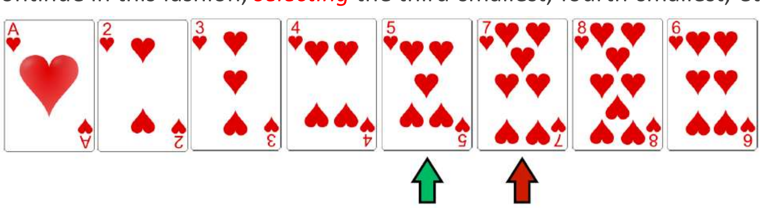
- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



47

Selection sort

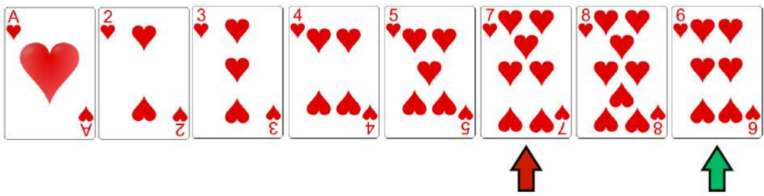
- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



48

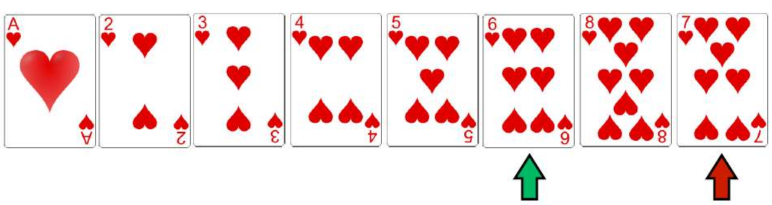
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



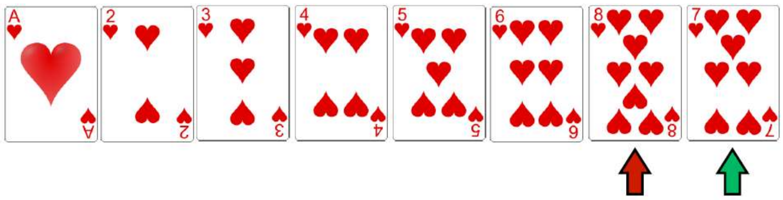
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



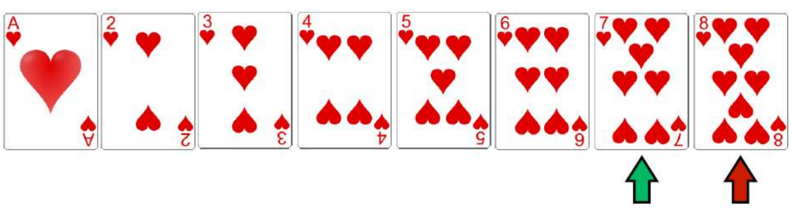
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



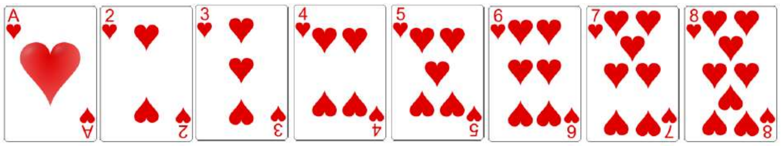
Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted



Selection sort

- Continue in this fashion, *selecting* the third-smallest, fourth-smallest, etc., until the list is sorted

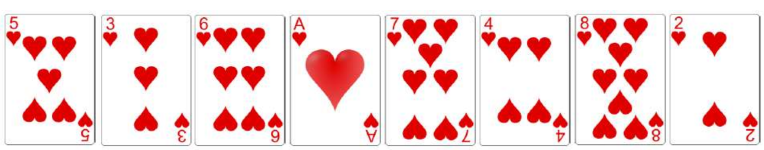


Finished!

53

A classical problem: sorting data (2)

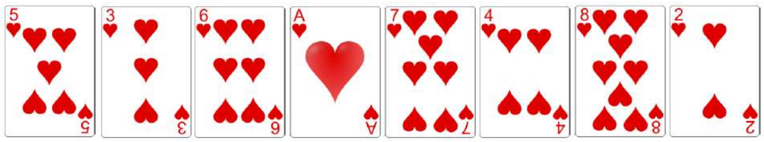
- Another sorting technique is **insertion sort**
- It repeatedly inserts the “next” card into its correct spot



54

Insertion sort

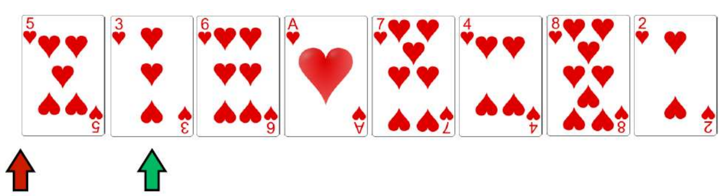
- We begin by leaving the first card (#5) where it is



55

Insertion sort

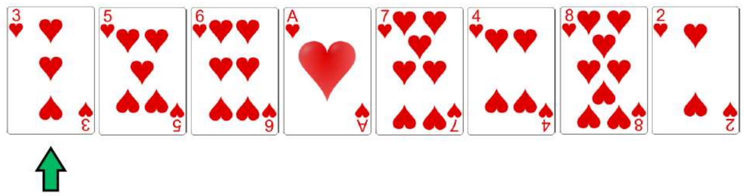
- The second card (#3) is smaller than the first card
- Insert it in front of the first card



56

Insertion sort

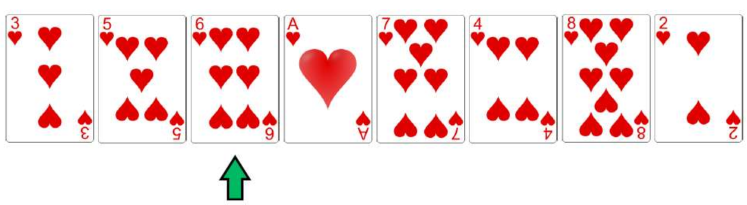
- The second card (#3) is smaller than the first card
- Insert it in front of the first card



57

Insertion sort

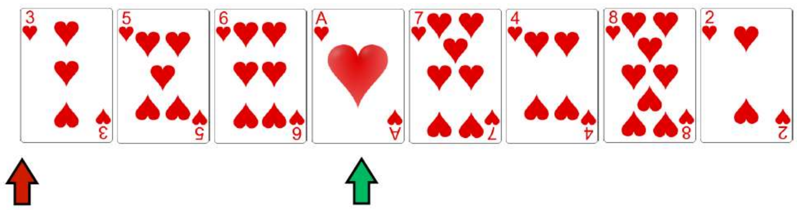
- The third card (#6) is larger than the first two cards
- So, we don't need to move it



58

Insertion sort

- The fourth card (#1) is smaller than the first three cards
- Insert it in front of the first card, shifting the others

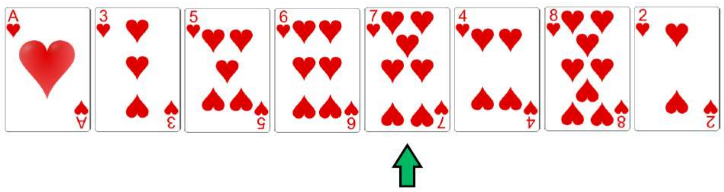


Insertion sort

- The fourth card (#1) is smaller than the first three cards
- Insert it in front of the first card, shifting the others

Insertion sort

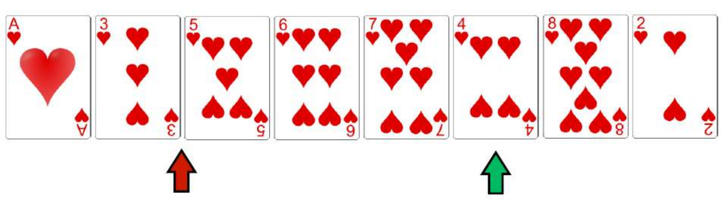
- The fifth card (#7) is larger than the first four cards
- So, we don’t need to move it



61

Insertion sort

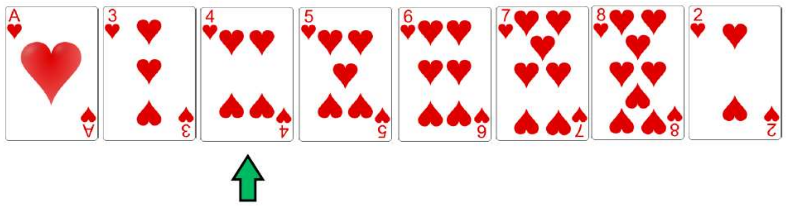
- The sixth card (#4) should be inserted in between the second (#3) and third (#5) cards



62

Insertion sort

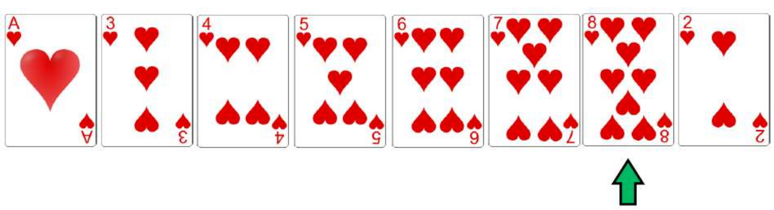
- The sixth card (#4) should be inserted in between the second (#3) and third (#5) cards



63

Insertion sort

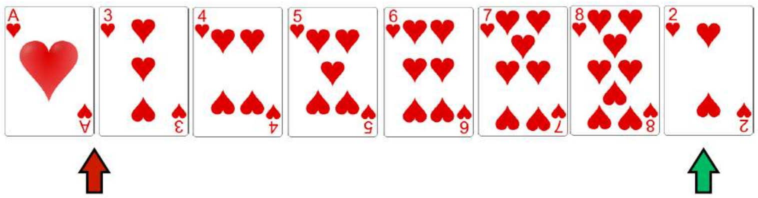
- . The seventh card (#8) is larger than the first six cards
- . So, we don't need to move it



64

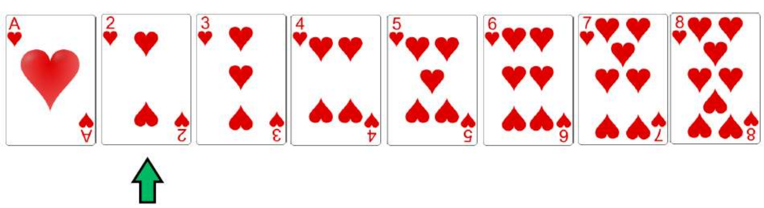
Insertion sort

- The eighth card (#2) should be inserted in between the first (#1) and second (#3) cards



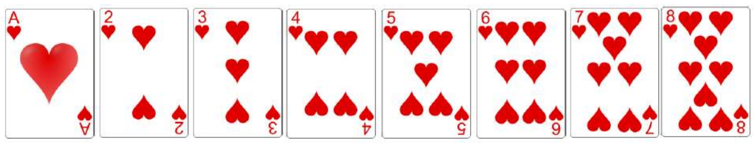
Insertion sort

- The eighth card (#2) should be inserted in between the first (#1) and second (#3) cards



Insertion sort

- The eighth card (#2) should be inserted in between the first (#1) and second (#3) cards



Finished!

67

Sorting algorithms

- We have just confirmed there can be different ways to solve the same computational problem
  - ➔ can derive many different **algorithms** for solving the sorting problem
- Algorithm:
  - A set of concrete steps
  - Steps solve a problem or accomplish some task
  - Solve in a finite amount of time
  - The earliest algorithm known as Euclid's algorithm dates from 300 BC and used to find the Lowest Common Denominator of two numbers.
- The Selection Sort and Insertion Sort algorithms are only two ways of sorting a list of values
- New problem: Wish to sort a list of student records by their GPAs.
  - Would both of these algorithms work?
  - Yes! A hallmark of a good algorithm is that it is general ➔ can solve a wide variety of similar problems

68

Limits of Computation

- List some examples that computer can do.
- List some examples that computer can not do.

Limits of Computation

What computer can do?	What computer cannot do?
Send email to a person if email address is known.	Find email of a person we met at a coffee shop.
Calculate difference investment options based on historical data.	Choose a perfect investment or predict success and future of companies.
Find information about colleges offering computer science course.	Make a perfect decision on the best school to attend.
Solve well defined problems.	Solve ambiguous problems.

Unsolvable Problems

If a computer tries to analyze every possible sequence of moves in response to this opening in a game of chess, it will have to consider over 1043 different games.

Computer solving one trillion combinations per second will compute the perfect game of chess if we are patient enough to wait 10^{21} years, so it is only unsolvable in a practical sense.

Rise of Quantum Computers: https://www.youtube.com/watch?v=WVv5OAR4Nik

To sum up…

- Computer science is the discipline of how to solve problems using computers
- We strive for efficient, general solutions that will work on a wide variety of problem types
- Although computer science has existed as a field for about 70 years, its roots in mathematics and computation go back thousands of years!
- CS is a very peculiar field
  - it relies partly on old mathematical ideas
  - it advances in development of new techniques at an extraordinary pace
- The semester you will be exposed to many of the modern topics in CS and also to some of the older mathematical content that is still very relevant today
- Watch BBC documentary exploring the development of computer: https://www.youtube.com/watch?v=ApJSz_0rkiA

Questions?