Welcome to Computer Team!

and Computer Systems I: Introduction

Noah Singer

Montgomery Blair High School Computer Team

September 14, 2017

Overview

Introduction

Units

Computer Systems

Computer Parts

1 Introduction

2 Units

3 Computer Systems

4 Computer Parts

Section 1: Introduction

Computer Team

Computer Team is...

- A place to hang out and eat lunch every Thursday
- A cool discussion group for computer science
- An award-winning competition programming group
- A place for people to learn from each other
- Open to people of all experience levels

Computer Team isn't...

- A programming club
- A highly competitive environment
- A rigorous course that requires commitment

Structure

Welcome to Computer Team!

Noah Singer

Introduction

Units

Computer Systems

Computer Parts

- Four units, each with five to ten lectures
- Each lecture comes with notes that you can take home!
- Guided activities to help you understand more difficult/technical concepts
- Special presentations and guest lectures on interesting topics

Conventions

Welcome to Computer Team!
Noah Singer
Introduction
Units
Computer Systems
Computer Parts

Vocabulary terms are marked in **bold**.

Definition

Definitions are in boxes (along with theorems, etc.).

Important statements are highlighted in italics.

Formulas are written in

$$\sum f(a^n c_y) \int m^{a^t} h$$

Section 2: Units

Unit I: Computer Systems

Fundamental question: *How do computers function at a low level?*

Lectures:

1. Introduction to Computer Team!
2. Kernels and Operating Systems
3. Networks
4. CPU Architecture
5. Threading and Parallelization
6. Compilers

Unit II: Theory of Computation

Fundamental question: *What is a computer, and what can it do?*

Lectures:

1. Automata
2. Context Free Grammars and Parsing
3. Guided Activity: Lambda Calculus
4. Turing Machines and Computability
5. Guided Activity: Reduction and P vs. NP
6. Computability and the Complexity Hierarchy
7. Randomization and Probabalistic Algorithms

Unit III: Machine Learning+

Fundamental questions: *How can we get computers to solve complex problems?*

Lectures:

1. Introduction to Machine Learning
2. Guided Activity: Neural Networks
3. Advanced Neural Networks
4. Natural Language Processing
5. Fourier Analysis and Signal Processing

Unit X: Algorithms and Contest Preparation

Fundamental question: *What is the best way to solve a given problem?*

Lectures:

1. Graph algorithms
2. Dynamic programming
3. Computational geometry
4. Greedy algorithms
5. Divide-and-conquer and recursive algorithms
6. Online algorithms

Section 3: Computer Systems

Basic constructs

Definition (Data)

Information represented by a sequence of symbols.

- The most basic unit of data in computer science is the **bit**, which is either a 1 or a 0.
- A single bit represents whether a single electrical signal is on (1) or off (0).

Definition (Program)

A series of **instructions** that tell a computer how to manipulate data to transform some defined **input** to a defined **output**.

Fixed- vs. stored-program computers

- Early computers were **fixed-program computers** that could only execute a single, predefined program, hardwired into the computer's circuitry.
- The theory of the **Universal Turing machine** demonstrated that a single computer could execute all programs.
- Important pioneers, like John Von Neumann, developed the **stored-program computer**, which *treats programs themselves as data*.

Code execution

- The random access memory, or **RAM**, stores data, organized into sequential blocks of eight bits called **bytes** assigned location numbers called **addresses**.
- The central processing unit, or **CPU**, executes code sequentially, which is stored as data in RAM.
- Each instruction is encoded as an **opcode** representing the operation being performed and zero or more operands.
- A single instruction can do one or more things, including:
  - Arithmetic and logic
  - Comparisons
  - Control flow
  - Memory access
  - Peripheral access

Section 4: Computer Parts

CPU

- Performs computations quickly
- Keeps track of which code to execute and executes it sequentially
- Accesses memory to store and retrieve data
- Interacts with other hardware devices (**peripherals**)
- Protects sensitive data and enforces security procedures
- Contains **registers** for extremely fast data access
- Executes a fixed sequence of instructions on machine startup from the ROM

RAM

- Stores data organized into bytes
- Organized both by **physical addresses** and **virtual addresses** which are mapped to physical addresses
- **Volatile**: does not persist after power loss

Other components

Permanent storage

- Non-volatile unlike RAM but much slower
- May be removable or non-removable
- Ex.: Hard drives, floppy discs, USB drives, etc.

I/O peripherals

- Enable interaction with users
- Often connected to the CPU through *buses* like USB or PCI
- Ex.: keyboard, mouse, speakers, microphone, printer, etc.

Other components

Graphics processing unit (GPU)

- Render the computer's visual output for interaction with the user
- Generally features hardware acceleration for e.g. 3D graphics
- Often faster than the CPU at raw computations!

Power supply

- Provides a steady power source for the components
- Temporarily insulates against power fluctuations

Motherboard

- Allows all the components to communicate with each other

Conclusion

Welcome to Computer Team!

Noah Singer

Introduction

Units

Computer Systems

Computer Parts

We'll learn about all this in much more detail in the coming weeks, but for now...

Welcome to Computer Team!