AP Computer Science Principles Summary

Robin Wiethüchter (AI Assisted)

May 6, 2025

DISCLAIMER: WORK IN PROGRESS

This document is currently under development. Sections may be incomplete, contain errors, or be subject to significant changes.

2 Big Idea 2: Data

Computers store, process, and transmit information digitally using binary data. This Big Idea focuses on how data is represented, manipulated, and used.

2.1 Bits and Bytes

The fundamental unit of digital information is the **bit** (binary digit), which can represent one of two states: 0 or 1. Bits are grouped together to represent more complex data.

- **Byte**: A group of 8 bits. A common unit for measuring data size.
- **Number of States**: With n bits, you can represent 2^n distinct states or values. For example, 8 bits (1 byte) can represent $2^8 = 256$ different values (often 0-255). Adding one more bit doubles the number of possible values.

2.2 Number Systems

Computers use binary internally, but humans often use other systems for convenience.

- **Binary (Base-2)**: Uses only digits 0 and 1. Each position represents a power of 2 (e.g., $1101_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = 13_{10}$).
- **Decimal (Base-10)**: The system we use daily, with digits 0-9. Each position represents a power of 10.

Overflow Error: Computers use a fixed number of bits (e.g., 8 bits) to represent numbers. This limits the largest value they can store (for 8 bits, often 255). An overflow error happens when a calculation's result is too large for the available bits. When this occurs, the result often "wraps around." For example, adding 1 to the maximum value might result in 0, or even a negative number if signed numbers are being used. This unexpected wrap-around leads to incorrect results.

Round-off Error: Similarly, computers use a finite number of bits to represent real numbers (like decimals or fractions), which can have infinite precision. This means computers often store an *approximation* rather than the exact value. This small difference is called a round-off error. While often tiny, these errors can accumulate during calculations, potentially leading to noticeable inaccuracies in the final result (e.g., 0.1 + 0.2 might be stored as something like 0.30000000000000004 instead of exactly 0.3).

2.3 Sampling and Representing Analog Data

To represent continuous, real-world phenomena (like sound waves or visual scenes) digitally, we need to convert them into discrete values. This involves:

- **Sampling**: Measuring the phenomenon's value at regular intervals in either time (for sound) or space (for images). The **sampling rate** (or resolution) determines how frequently these measurements are taken. More samples generally lead to a more accurate representation but require more data.

This process inherently involves approximation, as the continuous reality is represented by discrete digital data. The number of bits used to store each sample (bit depth) determines the level of detail or number of distinct values that can be represented.

Each box represents one bit (0 or 1)

1	0	1	1	0	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\begin{gathered} 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ = 128 + 0 + 32 + 16 + 0 + 0 + 0 + 1 \\ = 177_{10} \end{gathered}$$

Figure 2.1: Converting binary to decimal: Each bit's value is multiplied by its position value (powers of 2)

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 & = 11_{10} \\ & \color{red}{1} & \color{red}{1} & & \color{red}{1} & \\ + & 1 & 1 & 0 & 1 & +13_{10} \\ \hline 1 & 0 & 0 & 0 & 0 & = 24_{10} \end{array}$$

Figure 2.2: Binary addition example (1011 + 1101 = 11000) showing carry bits in red

Representing Images

Images are represented by sampling visual information across a 2D space. The image is divided into a grid of **pixels** (picture elements), each representing a sample point. For each pixel, color information is captured, often using RGB (Red, Green, Blue) values, each stored with a certain number of bits (bit depth) to represent different color intensities. Image file formats (JPEG, PNG, GIF) use different encoding and compression techniques.

Representing Sound

Sound is represented by sampling an analog sound wave's amplitude (height) at regular time intervals (the sampling rate). Each sample's amplitude is then stored using a specific number of bits (bit depth). Higher sampling rates and bit depths capture the original sound more accurately, resulting in higher fidelity but larger file sizes.

2.4 Data Compression

Compression reduces the number of bits needed to store or transmit data.

- **Lossless Compression**: Allows perfect reconstruction of the original data (e.g., ZIP, PNG). Algorithms look for redundancies (like repeated patterns) and encode them more efficiently.
- **Lossy Compression**: Discards some information deemed less important to achieve higher compression ratios (e.g., JPEG, MP3). This is acceptable for images and audio where humans may not notice the slight loss of quality. The original data cannot be perfectly recovered.