Data

AP Computer Science Principles ▪ Unit 2

AP Computer Science Principles

1	0	1	1

What we will cover

1. Binary numbers
2. Data compression
3. Extracting information
4. Using programs with data

0	1	1	0	1

bits

1 Binary

Everything is bits

A **bit** 位 is a single 0 or 1. **Binary** 二进制 (base 2) uses place values that are **powers of two** 2 的幂.

128	64	32	16	8	4	2	1	
0	0	1	0	1	1	0	1	$= 45$

$32 + 8 + 4 + 1 = 45$. Read binary → decimal by adding the place values where a bit is 1.

place value	128	64	32	16	8	4	2	1
bits	**1**	0	0	**1**	0	**1**	**1**	0

$150 = 128 + 16 + 4 + 2 \Rightarrow$ 10010110

Fixed bits, overflow & sampling

Convert & overflow

1011_2: place values $8, 4, 2, 1$; bits at $8, 2, 1$ give $8 + 2 + 1 = 11$.

With n bits you can make 2^n values, 0 to $2^n - 1$. Exceeding it is an **overflow error** 溢出错误 (4 bits max 15; $15 + 1$ overflows).

Analog 模拟 signals become **digital** 数字 by **sampling** 采样 – many quick measurements. Eight bits group into one **byte** 字节.

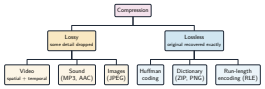
2 Compression

Data compression

Compression 压缩 stores the same data in **fewer bits** – less space, faster transmission.

Lossless 无损
every bit restored exactly (text, code)

Lossy 有损
discards unnoticed data; much smaller (photos, music)



Lossy trades **size** 大小 against **quality** 质量. Many methods remove **redundancy** 冗余 – e.g. "100 white pixels" → "white, 100 times".

3 Extracting information

Extracting information from data

Programs turn raw **data sets** 数据集 into **information** 信息, finding **patterns** 模式 and **trends** 趋势.

Filtering 筛选
keep the rows you want

Cleaning 清洗
fix duplicates, blanks, errors

Beware: a **correlation** 相关 never proves **causation** 因果, and **bias** 偏差 in how data was collected can mislead. **Metadata** 元数据 & **visualization** 可视化 help.

4 Using programs

Using programs with data

A program **processes** 处理 data: records in, new information out.

Iterate to summarise

A loop uses **iteration** 迭代 to visit each **element** 元素: summing $70, 85, 90, 60, 95$ gives 400; $\div 5 = 80$ (the average).

A **filter** 过滤器 keeps a **subset** 子集 that meets a condition; programs also **combine** 合并 sources, **transform** 转换 values, and **clean** 清洗 first – dirty data gives a wrong answer.

Worked example – choosing a compression

Which compression for a source-code file? For a holiday photo?

Code → lossless. Every bit matters – a single changed character breaks the program. Restoring the exact original is non-negotiable. **Photo** → lossy. A small quality loss is invisible, and the file gets much smaller.

The trade is size against fidelity. Lossy loss is permanent – you can never get the discarded data back.

Key points

Binary place values are powers of two; watch **overflow**

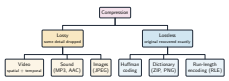
Lossless keeps every bit; **lossy** trades quality for size

Correlation is not **causation**; clean data first

Programs **iterate**, **filter** & transform data

Check yourself

1. Convert 1011_2 to decimal.
2. How many values can n bits represent?
3. Lossless or lossy for a program file – and why?
4. Does a correlation in the data prove causation?



Answers **1.** 11 **2.** 2^n (from 0 to $2^n - 1$) **3.** lossless – every bit must be restored exactly **4.** no