Computer Science Fundamentals:
Intro to Algorithms, Systems, & Data Structures

Christian J. Rudder

October 2024

Contents

Contents **1**

1 Building a Computer **5**

2 Computational Algorithms **6**
2.1 Information Theory 6
Defining Information 6

Bibliography **10**

This page is left intentionally blank.

Preface

Big thanks to **Christine Papadakis-Kanaris**
for teaching Intro. to Computer Science II,
Dora Erdos and **Adam Smith**
for teaching BU CS330: Introduction to Analysis of Algorithms
With contributions from:
S. Raskhodnikova, E. Demaine, C. Leiserson, A. Smith, and K. Wayne,
at Boston University

Please note: These are my personal notes, and while I strive for accuracy, there may be errors. I encourage you to refer to the original slides for precise information.
Comments and suggestions for improvement are always welcome.

Prerequisites

— 1 —
Building a Computer

— 2 —

Computational Algorithms

2.1 Information Theory

Defining Information

The following sections **heavily** reference Chris Terman's "Computation Structures" from the MIT OpenCourseWare, and Victor Shoup's "A Computational Introduction to Number Theory and Algebra" [2, 1].

Definition 1.1: Information

Information measures the amount of uncertainty about a given fact provided some data.

Example 1.1: Playing Deck of Cards

Given a 52-card deck, a card is drawn at random. One of the following data points is revealed:

a) The card is a heart (13 possibilities).

b) The card is not the Ace of Spades (51 possibilities).

c) The card is the "Suicide King," i.e., King of Hearts (1 possibility). ■

Definition 1.2: Quantifying Information

Given a discrete (finite) random variable X with n possible outcomes $(x_1, x_2, \ldots, x_n)$ and a probability $P(X) = p_i$ for each outcome x_i, the **information content** of X is defined as:

$$I(X_i) := \log_2\left(\frac{1}{p_i}\right)$$

Where $1/p_i$ is the probability of x_i, while Log base 2 measures how many bits (0 or 1) are needed to represent the outcome.

Example 1.2: Generalizing Information Content

A heart drawn from a 52-card deck may be represented as follows:

$$I(\text{heart}) = \log_2\left(\frac{1}{13/52}\right) \approx 2 \text{ bits}$$

More generally, we may redefine the information content as follows:

$$I(\text{data}) = \log_2\left(\frac{1}{M \cdot (1/N)}\right) = \log_2\left(\frac{N}{M}\right)$$

Where N is the total number of possible outcomes (e.g., 52 cards in a deck), and M is the number of outcomes that match the data (e.g., 13 hearts in a deck). Hence, $M \cdot (1/N)$ is the amount of information received from the data. Consider two more examples:

- **Information in one coin flip:** $\log_2(2/1) = 1$ bit ($N := 2, M := 1$).
- **Rolling 2 dice:** $\log_2(36/1) \approx 5.17$ or 6 bits ($N := 36, M := 1$).

■

Definition 1.3: Entropy

The **entropy** of a discrete random variable X is the average amount of information contained in all possible outcomes of X. It is defined as:

$$H(X) := E(I(X)) = \sum_{i=1}^{N} p_i \cdot \log_2\left(\frac{1}{p_i}\right)$$

Where function E is the expected value (i.e., average) of the information content $I(X)$ across all outcomes of X. This conveys how many bits b are needed to represent the outcomes of X:

- $b < H(X)$: Information is lost (i.e., not all outcomes can be represented).
- $b = H(X)$: An optimal representation.
- $b > H(X)$: Redundancy (i.e., not an efficient use of resources.).

Tip: For refreshers on $\sum$ consider our other text: Concise Works: Discrete Math.

Example 1.3: The Entropy of Four Choices

Consider a discrete random variable and it's possible outcomes $X := \{A, B, C, D\}$:

choice$_i$	p_i	$\log_2(1/p_i)$
A	1/3	1.58 bits
B	1/2	1 bit
C	1/12	3.58 bits
D	1/12	3.58 bits

Hence, the entropy of X is:

$$
\begin{aligned}
H(X) := \sum_{i=1}^{4} p_i \cdot \log_2\left(\frac{1}{p_i}\right) &= \left(\frac{1}{3} \cdot 1.58\right) + \\
&\quad \left(\frac{1}{2} \cdot 1\right) + \\
&\quad \left(\frac{1}{12} \cdot 3.58\right) + \\
&\quad \left(\frac{1}{12} \cdot 3.58\right) + \\
&\approx 1.626\,\text{bits}
\end{aligned}
$$

The entropy of X is approximately 1.626 bits, meaning that on average, we should be able to represent the outcomes of X using less than 2 bits per outcome.

■

Let's discuss how we might go about representing our outcomes:

Definition 1.4: Encoding

An **encoding** is an unambiguous mapping from a set of symbols to a set of bit strings:

- **Fixed-length encoding:** Uses a fixed number of bits to represent each symbol.
- **Variable-length encoding:** Uses a different number of bits for each symbol.

Example 1.4: Encoding Four Symbols

Consider the four symbols A, B, C, D and each possible encoding for them:

	Encoding for each symbol				Encoding for,
	A	B	C	D	"ABBA"
1.)	`00`	`01`	`10`	`11`	`00 01 01 00`
2.)	`01`	`1`	`000`	`001`	`01 1 1 01`
3.)	`0`	`1`	`10`	`11`	`0 1 1 0`

(1) Is a fixed-length encoding, (2) is a variable-length encoding, and (3) is also a variable-length encoding and uses fewer bits; **However**, it is ambiguous. Depending on how our program reads the string, it may group and misinterpret the bits.

E.g., (3) could be, "0 11 0" (A D A) or " 0 1 10" (A B C). Hence, an invalid encoding. ∎

Theorem 1.1: Binary Tree Encoding

Binary trees may represent unambiguous encodings, where each symbol is a leaf node, and each edge represents the next bit. Since each path is unique, the encoding is unambiguous.

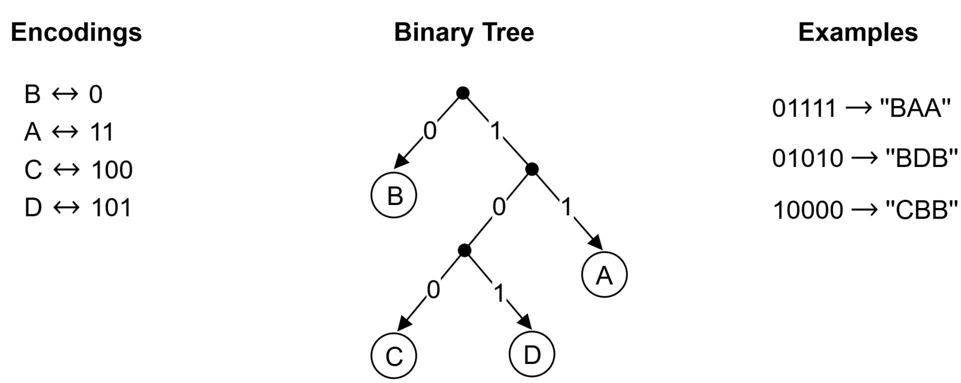

Figure 2.1: Encodings start at the root, each edge taken writes the next bit.

Bibliography

[1] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, version 2 edition, 2008. Electronic version distributed under Creative Commons Attribution-NonCommercial-NoDerivs 3.0.

[2] Chris Terman. 6.004 computation structures, 2017. Undergraduate course, Spring 2017.