ЗМІСТОВИЙ МОДУЛЬ 2 • ПОДАННЯ ІНФОРМАЦІЇ

04

Теоретичні основи кодування

Від правил подання повідомлень до ефективних кодів і стиснення даних.

01 Загальні положення. Класифікація кодів

02 Ефективне кодування

КОДОВІ КОМБІНАЦІЇ ОДНОГО ДЖЕРЕЛА

a_1	0,40	→	**0**
a_2	0,20	→	**100**
a_3	0,15	→	**101**
a_4	0,15	→	**110**
a_5	0,05	→	**1110**
a_6	0,05	→	**1111**

$$\bar{n} = 2{,}3$$

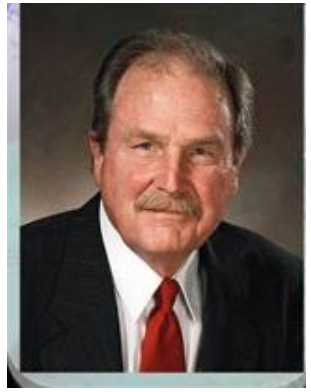
Дослідники статистичного кодування



Роберт Маріо Фано

1917–2016 • Турин — Нейплз

Італійсько-американський учений; почесний професор електроніки та комп’ютерних наук MIT. Його ім’я пов’язане з алгоритмом Шеннона–Фано. У 1963–1968 роках очолював Project MAC у MIT.



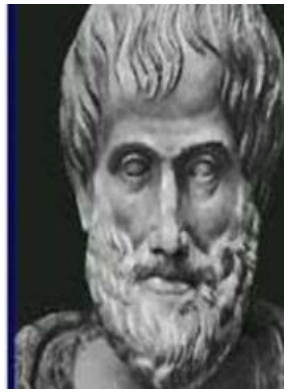
Девід Альберт Хаффман

09.08.1925–07.10.1999 • США

У 1952 році створив алгоритм префіксного кодування з мінімальною надмірністю. Код Хаффмана мінімізує середню довжину кодового слова за заданих імовірностей символів.

Уточнення біографії: Фано керував Project MAC, а не Массачусетським технологічним інститутом.

Логіка та двійкова система: передумови



Арістотель

384–322 рр. до н. е.

Формальна й модальна логіка, правила виведення, аксіоматичний метод і силогістика. У лекції ці ідеї показано як передумови формалізації міркувань.



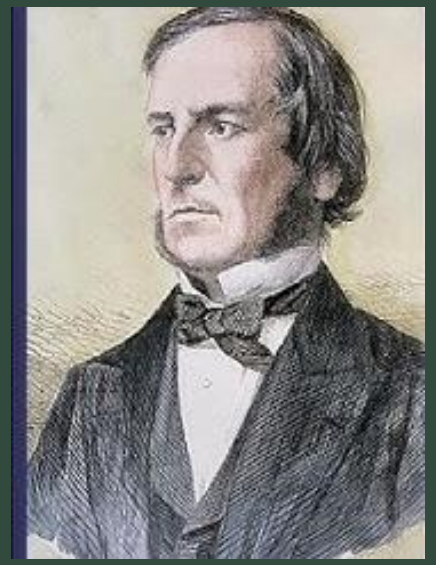
Готфрід Вільгельм Лейбніц

1646–1716

Двійкове подання чисел використовує цифри 0 і 1. Статтю «Explication de l’arithmétique binaire» опубліковано 1703 року.

Історичну лінію з початкового слайда збережено; датування праці про двійкову арифметику уточнено.

Булева алгебра: логічні операції



Джордж Буль

1815–1864

МАТЕМАТИЧНА ОСНОВА ЛОГІКИ

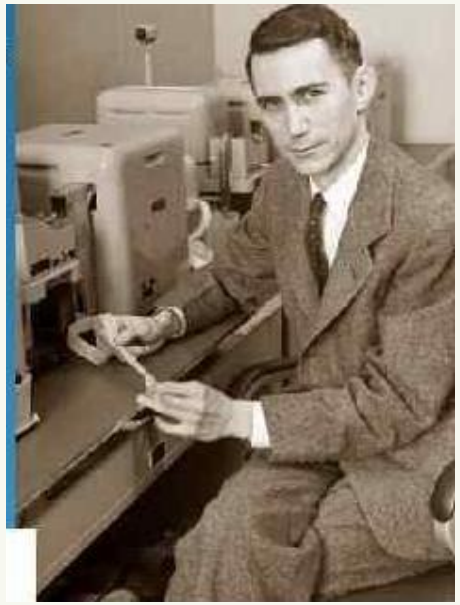
Булева алгебра — розділ математичної логіки, що вивчає систему логічних операцій над висловлюваннями.

1847 «Математичний аналіз логіки»

1854 «Дослідження законів думки»

У лекції підкреслено її значення для цифрової електроніки, програмування, теорії множин і статистики.

Клод Шеннон: від логіки до інформації



Клод Елвуд Шеннон

1916–2001

1937 — **Релейні й перемикальні схеми**

Магістерська робота в MIT: булева алгебра для аналізу схем. Робота з диференціальним аналізатором Буша.

1948 — **Математична теорія зв’язку**

Стаття в Bell System Technical Journal стала основою кількісного опису інформації та кодування.

bit — **Двійкова одиниця**

У статті Шеннона назву «bit» подано як скорочення binary digit, запропоноване Джоном Тьюкі.

Що означає кодування?

Мета кодування — пристосувати форму повідомлення до характеристик каналу зв’язку або пристрою, призначеного для перетворення чи збереження інформації.

Повідомлення

Множина дискретних повідомлень.

→

Кодові комбінації

Кожному повідомленню відповідає свій набір елементів — літер коду.

→

Сигнали

Елементарні сигнали переносять закодовану інформацію.

Кодом називають також правила утворення кодових комбінацій. Кодуванням може бути і перехід від одного коду до іншого: наприклад, від двійкового до двійково-десяткового.

Три цілі кодування повідомлень

01

Подання множини повідомлень

Одержати необхідну множину повідомлень за обмеженої кількості елементів кодового алфавіту; мінімальна основа коду m = 2.

02

Ефективність системи

Зменшити надлишковість переданих повідомлень і досягти максимальної ефективності системи передачі інформації.

03

Надійність передавання

Забезпечити необхідну вірогідність передавання інформації каналами з перешкодами.

Функціональна класифікація кодів

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ	ТИП КОДУ
Кодування повідомлень дискретних та неперервних джерел	**Коди джерела**
Кодування відліків неперервних сигналів під час аналого-цифрового перетворення	**Коди АЦП**
Кодування цифрових послідовностей для їх передавання лініями зв’язку	**Коди лінії**
Кодування блоків двійкових символів у модуляторах сигналів цифрової модуляції	**Модуляційні коди**
Коректування помилок, що виникають під час передавання сигналів каналами зв’язку	**Коректувальні коди**

П’ять паралельних відповідностей, а не ланцюг перетворень і не ієрархія підтипів.

Примітивні коди: натуральне подання

Натуральні коди — подання числового номера знака в позиційній системі числення з основою m. Перед кодуванням знаки джерела нумерують.

Знак джерела	m = 10	m = 8	m = 4	m = 2
a_0	0	00	00	0000
a_1	1	01	01	0001
a_2	2	02	02	0010
a_3	3	03	03	0011
a_4	4	04	10	0100
a_5	5	05	11	0101
a_6	6	06	12	0110
a_7	7	07	13	0111
a_8	8	10	20	1000

Стандартні коди: телеграфні алфавіти

Стандартні коди подають знаки джерела стандартизованими кодовими комбінаціями. Стандартизацію здійснюють національні або міжнародні організації.

Код / алфавіт	Відомості
Морзе	Стандартний телеграфний код.
Бодо	Перший міжнародний п’ятирозрядний телеграфний код.
МТА № 2, МТА № 3	Міжнародний телеграфний алфавіт
МТК № 2, МТК № 5	Міжнародний телеграфний код відрізняється від МТА наявністю кирилиці.

Стандартні коди: обмін інформацією

КОИ-7

Код відображення інформації

Семирозрядний код

ДКОИ-8

Двійковий код відображення інформації

Восьмирозрядне подання знаків.

ASCII

American Standard Code for Information Interchange

Американський стандартний код для обміну інформацією.

EBCDIC

Extended Binary Coded Decimal Interchange Code

Розширений двійково-десятковий код обміну інформацією.

Ефективне кодування та надмірність

Ефективними (економними) називають коди джерел повідомлень, які зменшують надмірність повідомлень під час кодування.

Це дозволяє ефективніше використовувати канали зв’язку та пристрої пам’яті. В обчислювальній техніці таке кодування називають стисненням інформації.

Статистична залежність

Зв’язки між знаками в повідомленнях джерела.

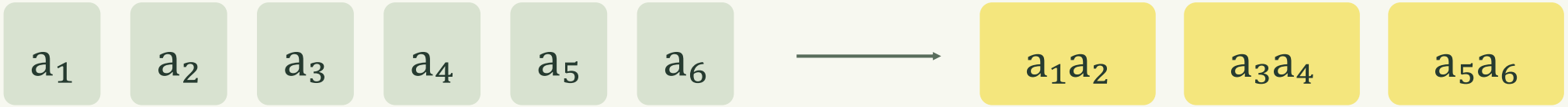
Нерівноймовірність

Різні ймовірності появи знаків у повідомленнях.

Ефективні коди також називають статистичними, оскільки їх будують з урахуванням імовірностей знаків.

I етап: укрупнення алфавіту

Деякі знаки джерела A об’єднують у знакосполучення з кількох знаків. Такі блоки розглядають як нові символи укрупненого алфавіту.



ЗНАКИ ПОЧАТКОВОГО АЛФАВІТУ

ЗНАКОСПОЛУЧЕННЯ

Якщо знаки джерела A незалежні, перший етап не потрібний. В іншому разі об’єднання має спиратися на модель статистичних залежностей.

Уточнення: саме по собі збільшення розміру блоку не доводить незалежності сусідніх блоків.

II етап: вимоги до ефективного коду

Нерівномірна довжина

У розглянутій схемі зменшення надмірності досягають нерівномірними кодами.

Імовірність і довжина коду

Чим більша ймовірність знака, тим менше двійкових символів має містити його кодова комбінація.

Баланс «0» та «1»

Для наближення ентропії окремого вихідного біта до 1 їхні ймовірності мають бути близькими до 0,5.

Префіксність

Жодне коротке кодове слово не є початком довшого. Тому для декодування не потрібні розділові знаки між словами.

Баланс окремих бітів не гарантує їхньої статистичної незалежності — це різні характеристики.

Числові характеристики рівномірного коду

Кількість кодових комбінацій	$M = m^n$
Довжина повного двійкового коду	$n = \log_2 M$
Місткість коду для алфавіту джерела	$M_A \le M$
Тривалість двійкового символу	$T_б = \frac{T_{зн}}{n}$

m — основа коду; Тзн — тривалість видачі знака джерелом. Для алфавіту: nмін = $\lceil \log_2 M_a \rceil$.

Характеристики нерівномірного коду

Середня довжина

Середня кількість кодових бітів на знак джерела.

$$\bar{n} = \sum_{k} P(a_k)\, n_k$$

Коефіцієнт стиснення

Порівняння рівномірної та середньої довжини коду.

$$\eta = \frac{n}{\bar{n}}$$

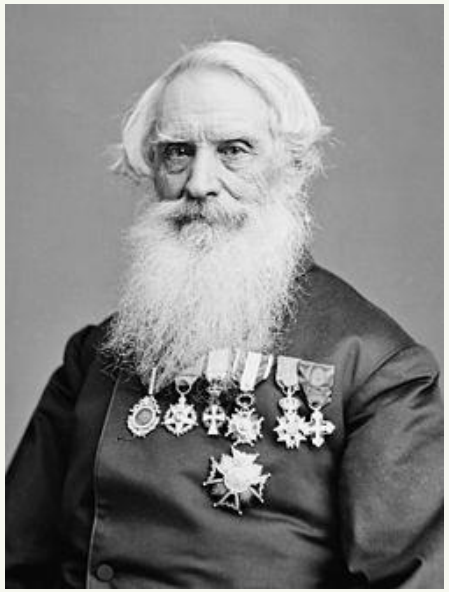
Коефіцієнт ефективності

Частка інформації в біті двійкового коду.

$$\mu = \frac{H(A)}{\bar{n}}$$

n_k — довжина k-ї кодової комбінації; $P(a_k)$ — імовірність знака; $H(A)$ — ентропія джерела, біт/знак.

Код Морзе та роль префіксності



Семюел Фінлі Бріз Морзе

1791–1872

Код Морзе (1837) наведено як ранній приклад статистичного коду. Його літерні комбінації не є префіксними без урахування пауз.

24 травня 1844 року було передано повідомлення «Дивовижні справи твої, Господи!».

SOS • • • — — — • • •

Save our souls — мнемонічне пояснення, а не походження позначення SOS.

Алгоритм Шеннона–Фано: п’ять кроків

1	**Упорядкування**	Розмістити знаки джерела за спаданням їхніх імовірностей.
2	**Поділ**	Розділити знаки на дві групи з приблизно рівними сумарними імовірностями.
3	**Призначення бітів**	Верхній групі приписати «0», нижній — «1» або навпаки.
4	**Повторення**	Повторювати поділ і приписування бітів для кожної підгрупи до окремих знаків.
5	**Формування слова**	Виписати всі біти, приписані знаку, зліва направо.

Різні допустимі поділи на групи можуть давати різні кодові комбінації.

Приклад: вихідні дані та завдання

Алфавіт джерела має $M_a = 6$ знаків. Знаки у повідомленнях незалежні. Тривалість видачі одного знака — 1 мс.

Знак	a_1	a_2	a_3	a_4	a_5	a_6
Імовірність	0,40	0,20	0,15	0,15	0,05	0,05

ПОТРІБНО ОБЧИСЛИТИ

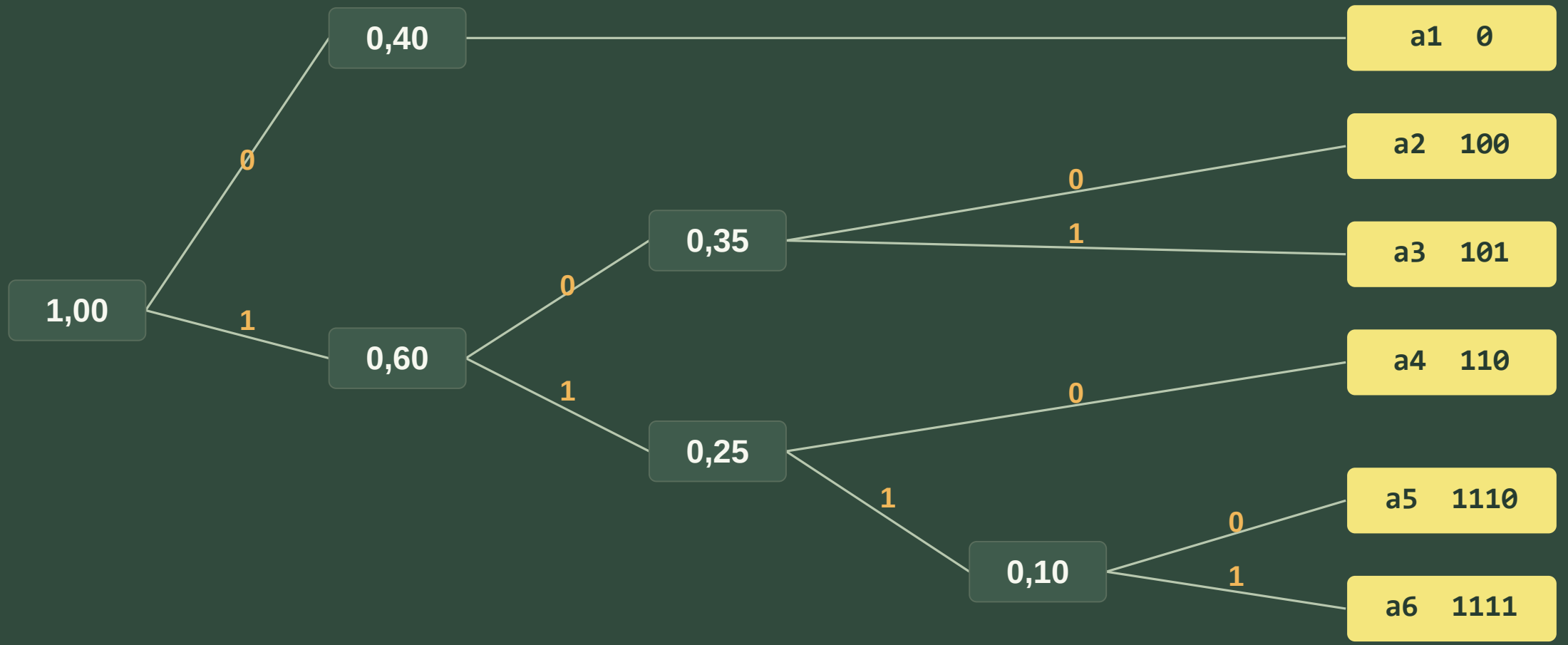
Код Шеннона–Фано; ентропію джерела; середню довжину кодової комбінації.

Швидкість на виході кодера; надмірність до і після кодування; коефіцієнти ефективності та стиснення.

Таблиця: код Шеннона–Фано

Знак	$P(a_i)$	Біт 1	Біт 2	Біт 3	Біт 4	Кодове слово	n_k
a_1	0,4	0	—	—	—	0	1
a_2	0,2	1	0	0	—	100	3
a_3	0,15	1	0	1	—	101	3
a_4	0,15	1	1	0	—	110	3
a_5	0,05	1	1	1	0	1110	4
a_6	0,05	1	1	1	1	1111	4

Як поділяються групи Шеннона–Фано



Дерево є редагованою візуалізацією саме того варіанта поділу, який наведено в таблиці.

Ентропія джерела A

$$H(A) = -\sum_{i=1}^{M_a} P(a_i) \log_2 P(a_i)$$

ПІДСТАНОВКА ІМОВІРНОСТЕЙ ПРИКЛАДУ

$$H(A) = -(0{,}4 \log_2 0{,}4 + 0{,}2 \log_2 0{,}2$$

$$+ 2 \cdot 0{,}15 \log_2 0{,}15 + 2 \cdot 0{,}05 \log_2 0{,}05)$$

$$H(A) = 2{,}246439\ldots \approx 2{,}25$$

Одиниця: біт на знак джерела. Значення 2,25 є коректним округленням.

Середня довжина коду і швидкість

$$\bar{n} = \sum_{k=1}^{M_a} n_k \, P(a_k)$$

$$\bar{n} = 1 \cdot 0{,}4 + 3 \cdot (0{,}2 + 0{,}15 + 0{,}15) + 4 \cdot 2 \cdot 0{,}05$$

$$\bar{n} = 2{,}3 \ \text{біт/знак}$$

$$R = \frac{\bar{n}}{T} = \frac{2{,}3}{10^{-3}} = 2300 \ \text{біт/с}$$

Умова прикладу: T = 1 мс. Для рівномірного коду шести знаків потрібні n = 3 біти на знак.

Надмірність A, ефективність та стиснення

Надмірність джерела A

$$K_{\text{надм}}(A) = 1 - \frac{H(A)}{\log_2 6} \approx 0{,}131$$

Коефіцієнт ефективності

$$\mu = \frac{H(A)}{\bar{n}} = \frac{2{,}246439\ldots}{2{,}3} \approx 0{,}977$$

Коефіцієнт стиснення

$$\eta = \frac{n}{\bar{n}} = \frac{3}{2{,}3} \approx 1{,}304$$

. Для шести знаків nмін = $\lceil \log_2 6 \rceil = 3$.

Для розрахунку коефіцієнта надмірності джерела B (по виходу кодера) необхідно обчислити ентропію цього джерела $H(B)$. Знайдемо ймовірності символів „1“ і „0“ на виході джерела B.

$$P(1) = 0{,}2 \cdot \frac{1}{3} + 2 \cdot 0{,}15 \cdot \frac{2}{3} + 0{,}05 \cdot \frac{3}{4} + 0{,}05 = 0{,}354;$$

$$P(0) = 0{,}4 + 0{,}2 \cdot \frac{2}{3} + 2 \cdot 0{,}15 \cdot \frac{1}{3} + 0{,}05 \cdot \frac{1}{4} = 0{,}646.$$

Ентропія джерела B

$$H(B) = -\left(0{,}354 \cdot \log_2 0{,}354 + 0{,}646 \cdot \log_2 0{,}646\right) = 0{,}94 \text{ дв. од.}$$

Коефіцієнт надмірності джерела B

$$K_{\text{надм}}(B) = 1 - 0{,}94 = 0{,}06.$$

Отримані значення показують, що кодування зменшує надмірність.

Вихід кодера

УТОЧНЕННЯ ЧИСЛОВОГО РОЗРАХУНКУ ОРИГІНАЛУ

* Для наочного підрахунку взято 100 знаків із частотами, що точно відповідають заданим імовірностям.

Знак	Код	Очікувано знаків*	Нулів у кодах	Одиниць у кодах
a_1	0	40	40	0
a_2	100	20	40	20
a_3	101	15	15	30
a_4	110	15	15	30
a_5	1110	5	5	15
a_6	1111	5	0	20
Разом	—	100	115	115

$$P_B(1) = 115 / 230 = 0{,}5; P_B(0) = 0{,}5 \qquad H_B(1) = -2 \cdot 0{,}5 \log_2 0{,}5 = 1$$

Маргінальна надмірність біта: 0.

Код Хаффмана: критерій оптимальності

Код Хаффмана (1952) — оптимальний префіксний код для дискретного джерела без пам’яті за критерієм мінімальної середньої довжини кодової комбінації знака.

Що мінімізується?

Середня довжина кодового слова за заданого розподілу ймовірностей і посимвольного префіксного кодування.

Де застосовується?

Кодування зображень і відео (JPEG, MPEG, H.261), аудіосигналів та факсимільних повідомлень.

Алгоритм Хаффмана: чотири кроки

1	**Упорядкувати**	Розмістити знаки за спаданням імовірностей.
2	**Об’єднати два мінімальні вузли**	Створити спільний вузол із сумарною імовірністю; двом гілкам приписати 0 та 1.
3	**Повторювати**	Знову обирати два найменші вузли, доки не буде досягнуто кореня дерева.
4	**Прочитати код**	Виписати біти шляху від кореня до потрібного знака. У схемі лекції — справа наліво.

Рівні ймовірності та перестановка позначок 0/1 допускають різні, але однаково оптимальні коди.

Приклад Хаффмана: дані та об’єднання

Знак x_i	a	b	c	d	e	f
Імовірність p_i	0,05	0,15	0,05	0,40	0,15	0,20
$-\log_2 p_i$, біт	4,32	2,74	4,32	1,32	2,74	2,32

Ентропія цього джерела $H(X) \approx 2{,}25$ біт/знак.

ПОСЛІДОВНІСТЬ СУМ ІМОВІРНОСТЕЙ

$$0{,}05 + 0{,}05 = 0{,}10$$

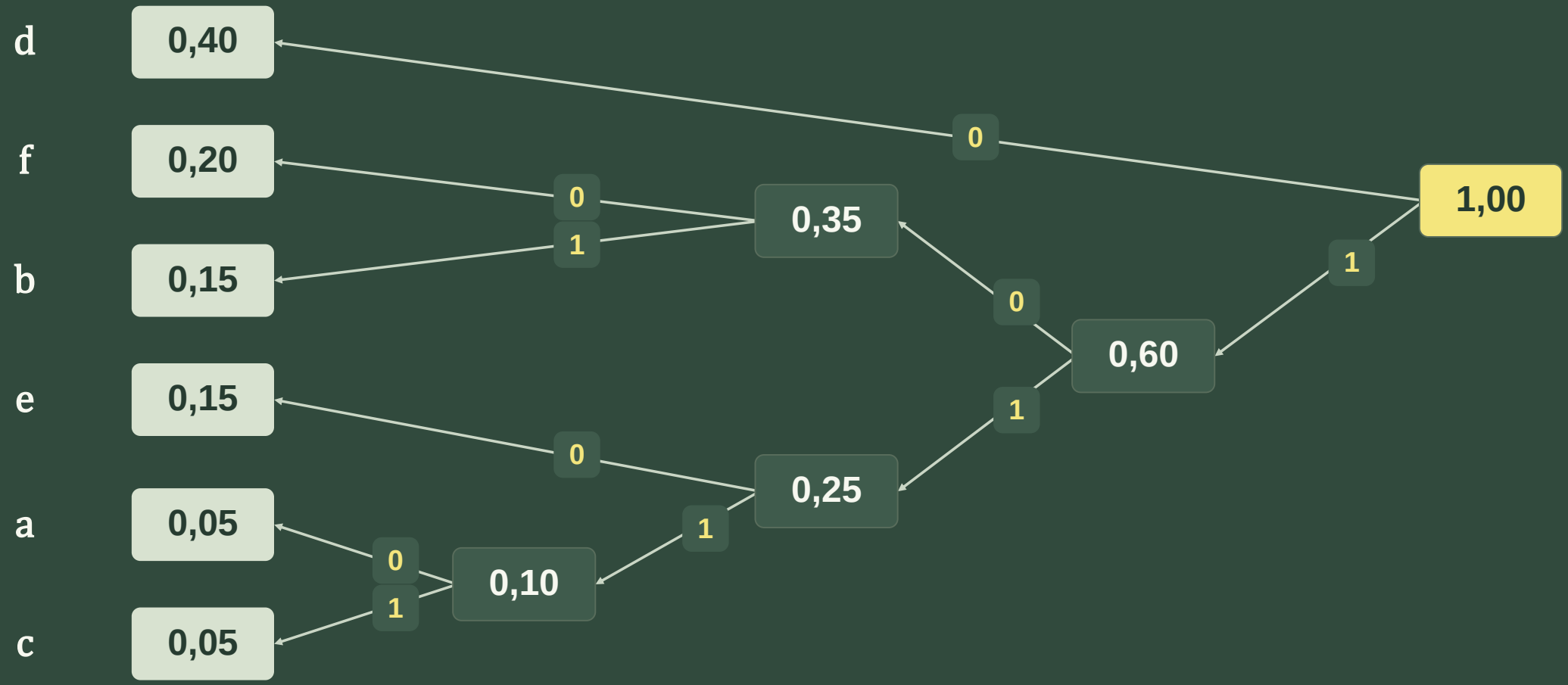
$$0{,}10 + 0{,}15 = 0{,}25$$

$$0{,}15 + 0{,}20 = 0{,}35$$

$$0{,}25 + 0{,}35 = 0{,}60$$

$$0{,}40 + 0{,}60 = 1{,}00$$

Дерево Хаффмана: від кореня до знака



Код кожного знака — послідовність бітів на гілках від кореня 1,00 до відповідного листка.

Кодові слова Хаффмана

Знак	d	f	b	e	a	c
Імовірність	0,40	0,20	0,15	0,15	0,05	0,05
Кодове слово	0	100	101	110	1110	1111
Довжина слова	1	3	3	3	4	4

$$\bar{n} = 0{,}4 \cdot 1 + 0{,}2 \cdot 3 + 2 \cdot 0{,}15 \cdot 3 + 2 \cdot 0{,}05 \cdot 4 = 2{,}3$$

Кодові слова і їхні довжини збігаються з оригінальною таблицею; відмінності між рівноймовірними знаками допустимі.

Обмеження кодів Хаффмана

Змінна довжина

Час видачі та обробки окремих кодових слів неоднаковий; це потрібно враховувати в системі.

Поширення помилок

Бітова помилка може порушити межі кодових слів і спричинити помилки декодування наступних символів.

Імовірнісна модель

Потрібно знати або оцінити статистику джерела. У практичних задачах вона може бути невідомою чи змінюватися.

Ці труднощі мотивують використання адаптивних і словникових методів стиснення, розглянутих далі.

Словникові алгоритми: LZ77 і LZ78

Ідею словникових алгоритмів запропонували Якоб Зів та Авраам Лемпель. Повторювані фрагменти повідомлення замінюються посиланнями на вже відомі фрази.

1977 • LZ77

Алгоритм започаткував сімейство методів, що використовують попередню частину вхідного потоку для кодування повторів.

1978 • LZ78

Альтернативний підхід започаткував сімейство методів із явним поповненням словника фраз.

Словникові алгоритми мають не найвищі коефіцієнти ущільнення, але швидку реалізацію словникових методів, особливо під час декодування. Ефективність залежить від даних і параметрів.

Евристичні (словникові) алгоритми ущільнення (типу LZ77, LZ78), як правило, шукають в файлі рядки, що повторюються, і будують словник фраз, що вже зустрічались. Зазвичай такі алгоритми мають цілий ряд специфічних параметрів (розмір буфера, максимальна довжина фрази і т.п.), підбір яких залежить від досвіду автора роботи, і ці параметри добираються таким чином, щоб досягти оптимального співвідношення часу роботи алгоритму, коефіцієнта стиснення та переліку файлів, що добре стискаються. В наш час алгоритми сімейства LZ77 широко використовуються в програмах універсальних архіваторів. Це, як правило, модифікації LZSS з використанням кодів Хаффмана (LHARC) Шеннона-Фано (PKZIP) або арифметичного кодування (HA).

У алгоритмах сімейства LZ78 ланцюжки в словнику збільшуються на один символ кожного разу, коли ланцюжок із словника зустрічається у вхідному потоці. У алгоритмі LZ78, так само, як і в LZ77, літеральні події і елементи словника чергувалися.

У 1984 р. Т. Велч запропонував алгоритм LZW (за початковими літерами прізвищ Лемпел (Lempel) і Зів (Ziv) (його авторів) і Уелч (Welch) у якому всі літеральні події є елементами словника. Таким чином, у вихідний потік поміщаються тільки посилання на елементи словника, що спрощує алгоритм. Велч показав, що алгоритм може бути легко реалізований апаратно. Програмна реалізація, як правило, виконується на основі вектора двійкових дерев. Корінням дерев служать елементи вхідного алфавіту.

[Майданюк В. П. Методи і засоби комп’ютерних інформаційних технологій. Кодування зображень. Навчальний посібник. – Вінниця: ВДТУ, 2001. – 63 с.]

Параметри та поєднання словникових методів

Алгоритми шукають повторювані рядки та будують словник фраз. Розмір буфера й максимальна довжина фрази впливають на час роботи та коефіцієнт стиснення.

ІСТОРИЧНІ ПРИКЛАДИ З ПОСІБНИКА, НАВЕДЕНОГО В ЛЕКЦІЇ

Приклад	Словниковий підхід	Наступне кодування
LHARC	Модифікації LZSS	Хаффмана
PKZIP	Модифікації LZSS	Шеннона–Фано
HA	Словникове стиснення	Арифметичне кодування

Часто стиснення – це лише один з етапів загального процесу обробки даних у системах телекомунікацій. Методи стиснення також широко використовують універсальні архіватори, системні утиліти для роботи з жорсткими та компакт-дисками, текстові редактори, програми для роботи зі звуковими, відео- та графічними даними тощо.

Для прикладу розглянемо один з найпростіших методів стиснення даних – метод кодування довжин серій (RLE – run length encoding), який ще називають методом кодування груп. Це адаптивний алгоритм стиску, який зменшує фізичний розмір груп однакових символів.

Групи однакових символів кодують 3-ма байтами – <pref, numb, symb>, де pref – унікальний код, який визначає початок групи, numb – кількість символів у групі, symb – код символу. Наприклад, кодом повідомлення "abccccddddbbcaaaa" буде рядок "ab#4c#4dbbc#4a". Цей метод буде ефективним за наявності значної кількості чотирисимвольних і більше повторень.

Можливі різні схеми групового кодування:

а) кодування вздовж осі X або Y, коли зображення має безліч горизонтальних або вертикальних ліній;

б) групове кодування, коли зображення складаються з елементів однакових розмірів (виду шахівниці).

RLE: приклад

ПОВІДОМЛЕННЯ ДО КОДУВАННЯ

abccccddddbbcaaaa

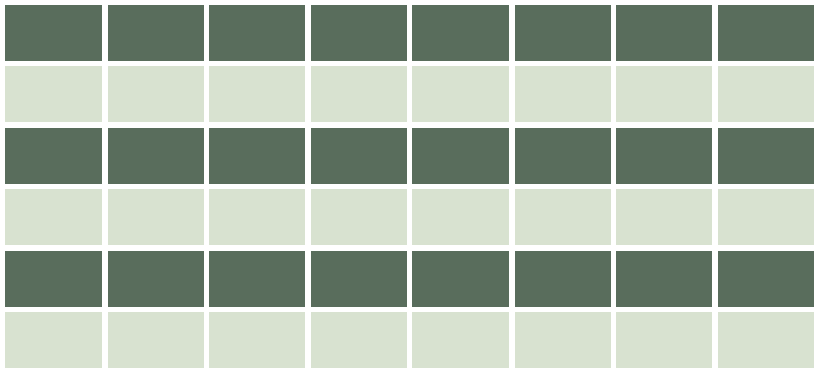
РЕЗУЛЬТАТ — # ЯК ПРЕФІКС СЕРІЇ

ab#4c#4dbbc#4a

У трьохбайтовому поданні виграш дають серії від чотирьох однакових символів. Короткі фрагменти у прикладі залишено без заміни.

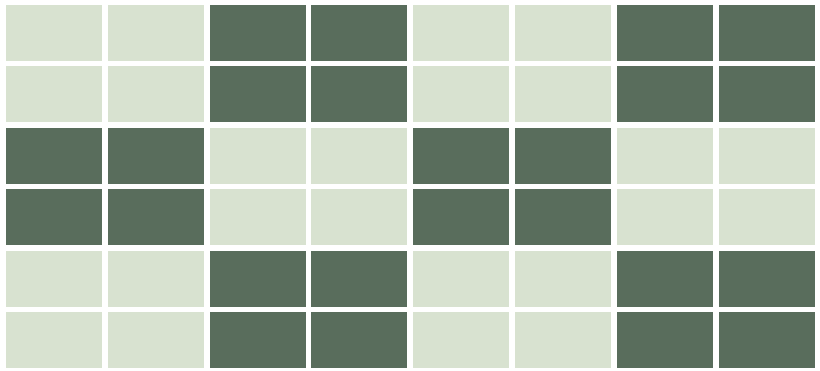
Групове кодування зображень

Сканування вздовж осі X або Y



Доцільне, якщо зображення містить багато горизонтальних або вертикальних ліній.

Кодування однакових елементів



Використовує повторювані блоки однакового розміру, наприклад структуру шахівниці.

Архівація даних: поняття й приклади

Архівація даних — каталогізація та стиснення даних з метою їхнього збереження або передавання.

WinRAR • Євген Рошал

Формат RAR, неперервне стиснення, шифрування, багатотомні та SFX-архіви. У джерелі також зазначено підтримку ZIP і розпакування інших форматів.

WinZip • Corel Corporation*

Автоматичне розпізнавання типів файлів, drag-and-drop та контекстне меню. Формати у переліку лекції: ZIP, TAR, UUencoded, XXencoded, BinHex, MIME.

Приклад архівації: 15 байтів → 9 байтів

ПОЧАТКОВЕ ПОВІДОМЛЕННЯ • ASCII

B	B	B	B	B	L	L	L	L	L	A	A	A	A	A
42	42	42	42	42	4C	4C	4C	4C	4C	41	41	41	41	41

НАВЧАЛЬНА ТРІЙКА: ПОЗИЦІЯ • КІЛЬКІСТЬ • ASCII-КОД

01 05 42	06 05 4C	0B 05 41
Позиція 1: п’ять «B»	Позиція 6: п’ять «L»	Позиція 11: п’ять «A»

Виправлено 0A → 0B. Залишається 9 байтів замість 15: економія 6 байтів, тобто 40%.

Графічні формати для збереження зображення

Всі графічні формати діляться на дві великі групи: растрові та векторні

Файли першої з них містять опис кожної точки зображення (представлені у вигляді матриці (bitmap), що складається з маленьких квадратиків різного кольору – пікселів). Найрозповсюдженішими форматами цієї групи є GIF і JPEG, що використовуються в Інтернеті, а також BMP (стандартний формат Windows) та TIFF, який використовується при зберіганні відсканованих зображень а також в поліграфії.

Файли векторних форматів містять не растрові точки, як у фотознімка, а математичні формули, що описують координати ліній. Наприклад, пряма представлена координатами двох точок, а коло – координатами центру та радіуса, тому досягається дуже малий розмір файлу. Векторні графічні формати використовують для передачі схем і рисунків, що складаються з набору ліній, описаних математичними формулами. У векторних форматах зберігаються логотипи, схематичні рисунки, текст, що призначений для виводу на друк та інші подібні об’єкти.

Стандартними засобами перетворити фотозображення у векторний формат практично неможливо, оскільки для цього потрібні спеціальні програми – конвертори. Найрозповсюдженішими розширеннями векторних файлів є: AI для пакета Adobe Illustrator; CDR для пакета CorelDRAW і WMF (ще один «стандартний» формат Windows).

[Основи теорії інформації та кодування : навч. посібник / [І. А. Прокопишин, Р. Є. Рикалюк, В. Ф. Чекурін, К. А. Червінка]. – Електрон. вид. – Львів : ЛНУ ім. Івана Франка, 2023. – 156 с.] URL: https://surl.li/jrhbxk

Висновки: основні цілі кодування

Адаптувати форму повідомлення

До характеристик каналу зв’язку або пристрою.

Подати потрібну множину повідомлень

За обмеженої кількості елементів кодового алфавіту.

Підвищити ефективність

Зменшенням надлишковості переданих повідомлень.

Забезпечити надійність

Необхідну вірогідність передавання інформації в каналах з перешкодами.