

# Практична робота №9. Стиснення даних. Кодування і декодування Гафмена

2025.11.25, м.Кременчук

Створив: Огнівський О.Є.

**Мета:** набути практичних навичок застосування алгоритму оптимального кодування Гафмена.

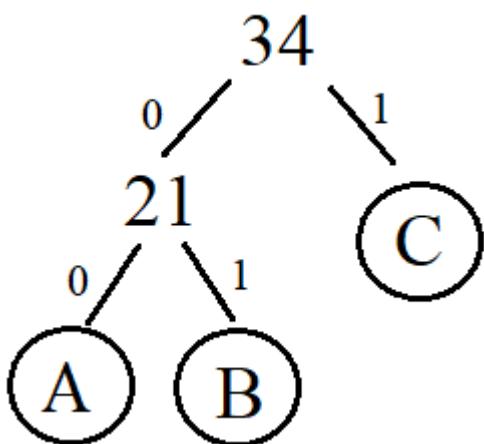
## Задача для самостійного розв'язання

## Частоти символів

- a -> 10 разів
  - b -> 11 разів
  - c -> 13 разів

Сума = 34

## Побудова двійкового дерева (Хаффман)



Закодуємо кожен символ на код:

- A  $\rightarrow$  00
  - B  $\rightarrow$  01

- C -> 1

Для всього тексту отримуємо бітову послідовність довжини 55 біт ( $10 \cdot 2 + 11 \cdot 2 + 13 \cdot 1 = 55$ ).

## Оцінити ефект від кодування порівняно з неоптимальним випадком

$\lceil \log_2(3) \rceil = 2$  біти на символ

- A -> 00
- B -> 01
- C -> 10

Тоді довжина =  $34 \cdot 2 = 68$  біт

Зекономлено:  $68 - 55 = 13$  біт

## Контрольні питання

---

### 1. Що таке кодування Хафмена та як воно працює?

Кодування Хафмена - це оптимальний алгоритм побудови префіксного двійкового коду для символів, який мінімізує середню довжину коду. Частіші символи отримують коротші коди, рідкісніші - довші.

Як працює алгоритм:

1. Підраховуються частоти символів.
2. Кожен символ стає вузлом у пріоритетній черзі.
3. Два найрідкісніші вузли об'єднуються в один.
4. Крок повторюється, поки не залишиться один корінь - дерево Хафмена.
5. Коди формуються шляхом обходу дерева (лівий = 0, правий = 1).

### 2. Як визначається оптимальний двійковий код Хафмена?

Оптимальний код автоматично формується під час побудови дерева Хафмена. Символи з найменшими частотами опиняються на найбільшій глибині, а часті - близче до кореня.

### 2. Які переваги має кодування Хафмена?

- Оптимальність для заданих частот.
- Однозначне декодування завдяки префіксності.
- Висока швидкість ( $O(n \log n)$ ).
- Простота реалізації.
- Використовується в багатьох форматах (gzip, PNG).

### 3. Як відбувається декодування даних?

Спосіб 1 - через дерево:

- починаємо з кореня,
- читаємо біти: 0 - ліворуч, 1 - праворуч,
- у листку - знайдений символ.

Спосіб 2 - через таблицю кодів:

- читаємо біти та звіряємо з таблицею «код → символ».

### 4. Які недоліки має кодування Хафмена?

- Не враховує контекст (залежності між символами).
- Малоефективний при рівних частотах символів.
- Потрібно зберігати/передавати таблицю кодів.
- Не оптимальний для малих алфавітів.
- Доведеться перебудовувати коди при зміні статистики.

### 5. Для чого використовується дерево в кодуванні Хафмена?

- Для формування оптимальних кодів.
- Для забезпечення префіксності кодів.
- Для швидкого та однозначного декодування.
- Для структурування частот так, щоб середня довжина коду була мінімальною.