

*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО*

Кафедра комп'ютерної інженерії та електроніки

*ЗВІТ З ПРАКТИЧНИХ РОБІТ
з навчальної дисципліни
«Алгоритми та методи обчислень»*

Тема «Стиснення даних. Кодування і декодування Гафмена»

Студент гр. КІ-23-1 ПІБ Кобець О. О.

Кременчук 2024

Практична робота № 9

Тема. Стиснення даних. Кодування і декодування Гафмена

Мета: набути практичних навичок застосування алгоритму оптимального кодування Гафмена.

Завдання

8. Маємо текст АВАСАВВCАСВCАВAДАВАС. Закодувати текст, використовуючи алгоритм Гафмена. Побудувати двійкове дерево. Оцінити ефект від кодування порівняно з неоптимальним випадком.

АВАСАВВCАСВCАВAДАВАС

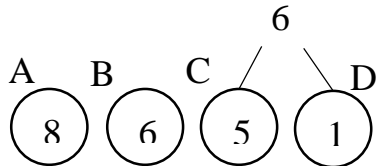
A – 8

B – 6

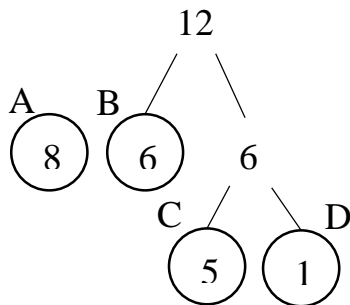
C – 5

D – 1

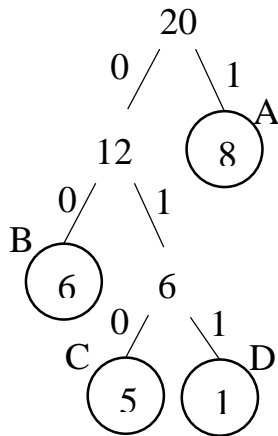
1.



2.



3.



A – 8	1
B – 6	00
C – 5	010
D – 1	011

Розмір у бітів:

A: 8 символів \times 1 біт = 8 біт

B: 6 символів \times 2 біти = 12 біт

C: 5 символів \times 3 біти = 15 біт

D: 1 символ \times 3 біти = 3 біти

Порівняння з оптимальним і неоптимальним випадком

Неоптимальний випадок: У нас 4 символи, отже, потрібно мінімум 2 біти на кожен символ. Для тексту з 20 символів: $20 \times 2 = 40$ біт.

Стиснене кодування (Гафмена): Розрахуємо загальну кількість бітів:
 $8 + 12 + 15 + 3 = 38$ біт.

Ефект стиснення:

$$\text{Ефективність} = \frac{40 - 38}{40} \times 100\% = 5\%$$

Контрольні питання

1. Що таке кодування Гафмена та як воно працює?

Кодування Гафмена — це метод стиснення даних, який використовує принцип заміни символів на двійкові коди змінної довжини, де більш часті символи отримують коротші коди, а рідкісніші — довші. Алгоритм працює наступним чином:

1. Визначають частоту кожного символу в тексті.
2. Будується дерево, де на кожному етапі з'єднуються два символи або групи символів з найменшою частотою.
3. Дерево продовжує розвиватися, поки не утвориться єдиний корінь.
4. Кожному символу призначають бінарні коди відповідно до шляху в дереві (ліва гілка — 0, права — 1).

2. Як визначається оптимальний двійковий код Гафмена для стиснення даних?

Оптимальний двійковий код Гафмена визначається на основі частот символів:

1. Спочатку для кожного символу визначають частоту.
2. Будується дерево Гафмена, де на кожному кроці з'єднуються два найменших за частотою елементи.
3. Коди для символів формуються на основі цього дерева, де більш часті символи отримують коротші коди.

3. Які переваги має кодування Гафмена над іншими методами стиснення даних?

Переваги кодування Гафмена:

1. **Ефективність:** Кодування Гафмена є оптимальним для випадків, коли частоти символів вхідного потоку різні, оскільки воно мінімізує середню довжину коду.
2. **Без втрат:** Алгоритм є безвтратним методом стиснення, тобто можна точно відновити вихідні дані після декодування.
3. **Швидкість:** Алгоритм порівняно швидкий при великих обсягах даних, особливо після побудови дерева Гафмена.

4. Як відбувається декодування даних, закодованих за допомогою кодування Гафмена?

Декодування за допомогою кодування Гафмена:

1. Декодування здійснюється шляхом проходження по дереву Гафмена, починаючи з кореня і слідуючи до листків.
 2. Якщо зустрічається біт "0", рухаються в ліву гілку дерева, якщо "1" — в праву.
 3. Коли досягається лист (символ), його додають до результату і починають знову з кореня дерева.
 4. Процес повторюється, поки не будуть оброблені всі коди.
5. Які є можливі недоліки кодування Гафмена?

Недоліки кодування Гафмена:

1. **Обмеження на варіативність частот:** Алгоритм Гафмена не працює ефективно, якщо всі символи мають однакові або близькі частоти, оскільки не буде значної різниці в довжині кодів.
 2. **Невелике стиснення при коротких текстах:** Для маленьких обсягів даних перевага алгоритму не така очевидна, і час на побудову дерева може перевищувати вигоди від стиснення.
 3. **Потреба в пам'яті:** Для збереження дерева Гафмена потрібно додаткове місце в пам'яті.
6. Для чого використовується побудова дерева в кодуванні Гафмена?

Побудова дерева в кодуванні Гафмена:

1. Дерево Гафмена є основною структурою даних для кодування і декодування. Воно визначає, які коди будуть призначені кожному символу.
2. Дерево допомагає оптимально організувати коди для символів залежно від їх частоти, що дозволяє досягти найменшого середнього розміру для кодованих даних.
3. Дерево надає ефективний спосіб для декодування даних, дозволяючи безпосередньо отримувати символи шляхом пересування по гілках дерева.