Завдання №3: Реалізація алгоритму стискання данних Хаффмана

1 Мета роботи

Опанувати методи роботи із двійковими потоками даних та алгоритми кодування даних зі змінними та/або нефіксованими довжинами кодових слів.

2 Основне завдання

У даній роботі вам необхідно реалізувати класичний статичний алгоритм Хаффмана. Реалізація повинна бути у вигляді автономної утиліти, яка дозволяє стискати файли у архіви та розпаковувати їх назад. Вимоги до інтерфейсу утиліти, роботи з іменами файлів тощо такі само, як і для `Base64`- чи `RLE`-кодека із завдання №1. Для полегшення роботи можна вважати, що ваша програма працює із файлами, розмір яких не перевищує 2^{32} байтів (приблизно 4 Гб).

3 Вимоги до реалізації алгоритму Хаффмана

При реалізації алгоритму Хаффмана врахуйте такі деталі.

1. Використовується байтовий алфавіт (тобто, символами є числа від 0 до 255); це типовий підхід для роботи із двійковими файлами довільної природи.

2. Програма повинна спочатку знайти кількість кожного символу у наданому файлі (можна шукати і частоти, як зазвичай описують алгоритм Хаффмана, але тоді ми прирікаємо себе на використання чисел із плаваючою точкою; за можливості завжди уникайте цього). Для зберігання кількостей використовуйте 32-бітові беззнакові цілі числа.

3. У вихідному файлі на початку необхідно зберегти таблицю кількостей; відповідно, перші $256 \cdot 4 = 1024$ байти повинні містити дану таблицю. Стиснені дані повинні йти після таблиці.

4. Декодер, в свою чергу, повинен спочатку вичитати таблицю кількостей та за нею відновити двійкове дерево, а вже після починати розпаковувати дані.

Файли для стискання повинні розглядатись як послідовності байтів, а стиснуті файли — як двійкові потоки даних (із використанням типу даних, написаного у завданні №2).

Опціональне завдання (за бонусні бали): придумайте та реалізуйте схему зберігання таблиці частот або дерева Хаффмана, яка б займала якомога менше місця.

4 Практичний аналіз реалізації алгоритму Хаффмана

За допомогою вашої реалізації алгоритму Хаффмана проведіть первинний порівняльний аналіз якості стискання даних з різних джерел. Для цього зберіть по 10 достатньо великих файлів різних типів (наприклад: виконувані файли `.exe`, текстові файли `.txt`, офісні документи `.docx`, розширені текстові файли `.rtf` тощо, вибір залишається за вами), стисніть усі файли вашою утилітою та обчисліть співвідношення між розмірами оригінальних файлів та їх стиснутих образів. Урахуйте окремо розмір стиснутого образу та розмір стиснутого образу із метаданими (в першу чергу, збереженим масивом частот чи деревом Хаффмана).

5 Вимоги до оформлення звіту

У звіті ви повинні відобразити:

- опис та особливості вашої реалізації алгоритму Хаффмана; зокрема, повинно бути описано, як ви зберігаєте побудований код у стиснутому файлі;
- обрані типи файлів для порівняльного аналізу (не менше п'яти різних);
- результати порівняльного аналізу: для кожного типу файлів — усереднене значення коефіцієнту стискання, а також мінімальне та максимальне його значення;
- висновки з проведеного порівняльного аналізу.