

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Структури даних, аналіз і алгоритми комп'ютерної обробки інформації»

«Алгоритм Хаффмана»

Виконав:	Сирота Ангеліна Олександрівна	Перевірила:	Бичков Олексій Сергійович
Група	ІПЗ-21	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Умова завдання

Написати програму реалізації алгоритму Хаффмана.

Аналіз

Вводимо рядок символів, з яким будемо працювати.

Використовуючи словник запам'ятовуємо кожен символ рядка і частоту, з якою він зустрічається (ключ: символ, значення: частота). З отриманих значень створюємо список вузлів бінарного дерева. За допомогою алгоритма Хаффмана будуємо бінарне дерево.

Алгоритм працює наступним чином:

- Усі вузли зі списку сортуються у порядку спадання їх частот і розглядаються останні 2, чиї частоти найменші.
- Створюється новий вузол, частота якого є сумою частот двох взятих вузлів. У посиланнях щойно створеного батьківського вузла на лівий і правий вузол вказуються ті 2 дочірні вузли, на основі яких було створено батьківський.
- Дочірні вузли видаляються зі списку вузлів. Замість них додається батьківський.
- Ця операція повторюється доти, доки у списку вузлів не залишиться лише 1 вузол.

На основі дерева будується таблиця Хаффмана (також використовується словник), яка містить символ рядку і бінарний код, отриманий шляхом пошуку цього символу у глибину:

- У функцію передаються значення шуканого символу і пустого рядка, який буде заповнюватися «0» та «1» в ході пошуку.
- Якщо дані вузла, що розглядається співпадають зі значенням шуканого символу, то виводиться рядок, що містить бінарний код цього символу.
- Інакше, якщо вузол має лівий нащадок, додаємо у рядок «0» і викликаємо ту ж саму функцію для пошуку елемента у лівому піддереві. Аналогічно і з правим піддеревом.
- Після завершення рекурсії, якщо шлях не є нульвим, повертаємо його.

- Якщо ж дані вузла не співпадають з шуканим символом, повертаємо 0.

На основі даних цієї таблиці кодується рядок. Замість кожного символу записується його бінарний код.

Для розшифровки використовується дерево Хаффмана. Розглядається кожен символ у закодованому рядку. Якщо він є «0», то розглядаємо ліве піддерево, а якщо «1» - праве. Дійшовши до листка записуємо його значення у рядок і повертаємось у корінь дерева.

Код алгоритму

Весь код викладений на github. Його можна переглянути [тут](#).

Тестування

```
Введ?ть рядок: huffman

Таблиця Хаффмана
символ | код
-----+-----
h      | 11
u      | 100
f      | 00
m      | 101
a      | 010
n      | 011

-----

Закодований рядок: 111000000101010011

-----

Розшифрований рядок: huffman
-----
```

```
Введ?ть рядок: aaaaaabbbbbbbcd

Таблиця Хаффмана
символ | код
-----+-----
a      | 1
b      | 00
c      | 011
d      | 010

-----

Закодований рядок: 1111111000000000000000011010010010

-----

Розшифрований рядок: aaaaaabbbbbbbcd
-----
```

Аналіз отриманих помилок

В ході реалізації алгоритму Хаффмана для побудови бінарного дерева спочатку було використано пряме сортування і, відповідно, бралися перші 2 елементи. Це призвело до неправильної побудови бінарного дерева. Символ, який зустрічався найбільшу кількість не мав найменший бінарний код. Було вирішено використати зворотнє сортування і брати останніх 2 елементи.

Висновок

В ході цієї лабораторної роботи було написано програму, яка реалізує алгоритм Хаффмана. В ході реалізації було побудовано бінарне дерево Хаффмана і на основі нього побудовано таблицю Хаффмана, яка містить бінарні коди для кожного символу. За допомогою таблиці було створено закодований рядок. Дешифрування рядка відбувалося за допомогою дерева Хаффмана.