УДК 004.75

Богдан Бугай,
Юрій Гордієнко

МЕТОД ОПТИМІЗОВАНОГО ЗНАХОДЖЕННЯ ПОДІБНИХ БІНАРНИХ РЯДКІВ У ПЕВНІЙ МНОЖИНІ

METHOD OF OPTIMIZED SEARCH SIMILAR BINARY STRINGS IN A CERTAIN SET

У статті розглядається питання швидкого знаходження подібних рядків у певній множині бінарних рядків з однаковою довжиною, для яких відстань Геммінга не перевищує задане значення.

Ключові слова: відстань Геммінга, бінарний рядок, хеш таблиця.
Табл.: 2. Бібл.: 2.

The article deals with the problem of the rapid finding of similar strings in a certain set of binary strings with the same length, for which the Hamming distance does not exceed a given value.

Key words: Hamming distance, binary string, hash table.
Tabl.: 2. Bibl.: 2.

Актуальність теми дослідження. Порівняння бінарних рядків за допомогою відстані Геммінга ключовий момент в задачах, де вони приймаються в якості характеристики певного текстового або мультимедійного контенту, особливо в задачах де використовується перцептивний хеш в якості характеристики досліджуваного елемента.

Постановка проблеми. Відсутність швидкого методу для знаходження подібних бінарних рядків в певній множині для яких відстань Геммінга не буде перевищувати задану.

Аналіз останніх досліджень і публікацій. Протягом останніх років не з’являлося статей присвячених цій темі.

Виділення недосліджених частин загальної проблеми. Дана стаття присвячена вивченню та аналізу запропонованого підходу для оптимізації пошуку подібних рядків в певній множині бінарних рядків з однаковою довжиною, для яких відстань Геммінга не перевищує задане значення.

Постановка задачі. Для певного бінарного рядка потрібно знайти всі рядки в певній множині для яких відстань Геммінга не буде перевищувати задане значення.

Викладення основного матеріалу. Бінарний рядок складається з елементів множини $\{0, 1\}$. Для нього можна виділити три основні характеристики: кількість нулів, кількість одиниць та загальна довжина рядка.

Якщо взяти певну множину бінарних рядків з однаковою довжиною, то їх можна розділити на групи, в яких кількість одиниць в рядку дорівнює номеру групи.

Таблиця 1

Ідентифікатор рядка	Бінарний рядок
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

В Таблиці 1 представлена деяка вибірка бінарних рядків, для простоти бінарний рядок в десятковій системі числення відповідає ідентифікатору. Далі розбиваємо рядки на групи по кількості одиниць.

Таблиця 2

Кількість одиниць	Номер ідентифікатора рядка
0	0
1	1, 2, 4
2	3, 5, 6
3	7

У програмній реалізації Таблиця 2 може бути хеш-таблицею або деякою базою даних, яка надає ефективний доступ за ключем. Даний процес групування показаний для примітивної вибірки, але суть залишається незмінною для вибірки будь-якої величини.

Відстань Геммінга обчислюється як число позицій, в яких відповідні значення двох рядків різні [1]. Для двох бінарних рядків S_1 та S_2 можна підрахувати за формулою [2]:

$$d_h(S_1, S_2) = \sum_{k=1}^{p} |S_1(k) - S_2(k)| \quad (1)$$

де p - це довжина рядку.

Приклад підрахунку:

$$d_h(001{,}0110) = 1 \quad (2)$$

Відстань Геммінга має наступні властивості:

- $d(S_1, S_2) \geq 0$
- $d(S_1, S_1) = 0$
- $d(S_1, S_2) = d(S_2, S_1)$

Так як відстань для двох однакових рядків повинна бути нуль, то відповідно в них кількість одиниць та нулів однакова. Для рядків, між якими відстань повинна бути 1, на одну одиницю чи нуль повинно бути більше ніж в цільового рядка. З цього випливає, що для знаходження всіх подібних рядків між якими є задана відстань Геммінга d потрібно виконати наступний алгоритм:

1) Підраховуємо кількість одиниць в рядку і позначаємо її як C_s;
2) Вибираємо категорії в яких ідентифікатори належать множині S:

$$S = \{C_s - d, \dots, C_s, \dots, C_s + d\} \quad (3)$$

де $C_s - d \geq 0$ та $C_s + d \leq$ length(S)

3) Послідовно проходимо по кожній з категорій та підраховуємо відстань Геммінга для кожного рядка категорії з цільовим рядком;

4) Рядки, для яких відстань буде менша або рівна d, включаємо в результуючу множину;

Даний метод дозволяє проходити лише частину всієї множини для пошуку рядків подібних до цільового. В залежності від відстані Геммінга та характеру запиту ця частина може змінювати свої розміри.

Для тесту згенеруємо рядки, значення яких в десятковій системі числення лежатимуть в межах між 0 та 65535. Код для генерації на мові програмування Python:

```
table = dict()
size = 16
for i in range(2 ** size):
    s = '{0:{fill}{size}b}'.format(i, fill=0, size=size)
    count = sum(int(char) for char in s)
    if count in table:
        table[count].append(s)
    else:
        table[count] = [s]
```

Припустимо, в даній множині нам потрібно знайти подібні рядки для рядка 1111111100000000 з допустимою похибкою 1, то за алгоритмом подібні рядки потрібно шукати в 7, 8 та 9 категорії. Сумарна довжина цих категорій дорівнює $11440 + 12870 + 11440 = 35750$. Це означає, що 29785 рядків не потрібно сканувати. Для прикладу були взяті категорії з найбільшою кількістю рядків, якщо кількість одиниць в рядку прямує до 0 або до довжини рядку, то операцій сканування буде ще менше. Наприклад для рядка 1111111111110000 та з допустимою похибкою 1 потрібно провести $4368 + 1820 + 560 = 6748$ операцій, так як були обрані 11, 12 та 13 категорії. Зменшення кількості операцій відчутне.

Висновки. У роботі продемонстровано, як за допомогою розбиття бінарних рядків на категорії по кількості одиниць в рядку можна досягти приросту в продуктивності пошуку в задачах, де потрібно знаходити подібні рядки із заданою відстанню Геммінга. Даний алгоритм підходить для задач, де використовується пошук по перцептивних хешам.

Представлений алгоритм дуже просто можна покласти в основу багатопоточної програми, так як категорії, по яких потрібно провести сканування незалежні, тобто кожен потік може опрацьовувати свою категорію, потрібно лише використати будь-який примітив синхронізації для результуючої множини для уникнення станів гонки даних.

Є декілька напрямків для майбутньої роботи. Перш за все - покращення пошуку подібних рядків у межах певної категорії.

Список використаних джерел

1. Ionescu M., Ralescu A., (2004). Fuzzy Hamming Distance in a Content-Based Image Retrieval System. International Journal of Information Technology and Knowledge Management, IEEE (p. 2).

2. Xu Z., (2007). Fuzzy Optimization and Decision Making. Springer (p. 3).

ДОВІДКА ПРО АВТОРІВ

Бугай Богдан Андрійович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Buhai Bohdan - student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: bohdan.buhai@yahoo.com

ГордієнкоЮрійГригорович – професор, кафедраобчислювальноїтехніки, НаціональнийтехнічнийуніверситетУкраїни «КиївськийполітехнічнийінститутіменіІгоряСікорського».

Gordienko Yuri– professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: yuri.gordienko@gmail.com

Buhai Bohdan,
Gordienko Yuri

METHOD OF OPTIMIZED SEARCH SIMILAR BINARY STRINGS IN A CERTAIN SET

Target setting. The comparison of binary strings with Hamming's distance is a key point in tasks where they are taken as a characteristic of a particular text or multimedia content, especially in tasks where the perceptual hash is used as a characteristic of the investigated element.

Problem formulation. The absence of a fast method for finding similar binary strings in a certain set for which the Hamming distance does not exceed the given one.

Actual scientific researches and issues analysis. During recent years, there were no articles devoted to this topic.

Uninvestigated parts of general matters defining. This article is devoted to the study and analysis of the proposed approach to optimize the search for similar strings in a certain set of binary strings with the same length, for which the Hamming distance does not exceed a given value.

The research objective. For a certain binary string need to find all the rows in a certain set for which Hamming distance does not exceed the given value.

The statement of basic materials. An analysis of the division method into categories in the task where similar binary strings are to be found is carried out. An algorithm for search optimization is described. The results were well-interpreted and informative.

Conclusions. The work of the algorithm and the obtained results are analyzed. The presented approach proved its effectiveness. The results of the experiments are summarized and following steps described to improve the performance of the algorithm.

Key words: Hamming distance, binary string, hash table.