УДК 512.12:004.424.47

ИСПОЛЬЗОВАНИЕ АРИФМЕТИЧЕСКИХ ПРОГРЕССИЙ ПРОСТЫХ ЧИСЕЛ ПРИ ХЕШИРОВАНИИ ПО ОСТАТКУ ОТ ДЕЛЕНИЯ

Климинтионак В.С, студент гр.451003, Кужик Д.С, студент гр.451003

Белорусский государственный университет информатики и радиоэлектроники[1]
г. Минск, Республика Беларусь

Баркова Е.А. – канд. физ.-мат. наук

Аннотация. Главной целью данной научной работы является исследование и оптимизация методов хеширования с использованием арифметических прогрессий простых чисел. В работе рассматриваются хеш-функции, их влияние на скорость и точность хеширования, а также изучается проблема коллизий. Разработаны новые алгоритмы: алгоритм поиска арифметических прогрессий простых чисел и алгоритм оценки распределения хэш-функций.

Ключевые слова. Хеширование, хеш-функция, хеш-таблица, коллизия, распределение, арифметическая прогрессия, теорема Грина-Тао, дисперсия, эффективность, оптимизация, алгоритм, простые числа.

Впервые идея хэширования возникла в 1953 г. Ханс Петер Лун отправил в IBM внутреннюю записку, предложив помещать информацию в «ячейки» для ускорения ее поиска.

Хэширование – это процесс преобразования данных в уникальный (или почти уникальный) хеш-код фиксированной длины с использованием специальной хеш-функции.

Хеш-функция – это функция, получающая на входе данные, обычно строку, и возвращающие число.

То есть множество всех хеш-кодов, полученных из всех входных данных при помощи данной хеш-функции.

Хранить информацию в таком виде удобнее, так как за 1 операцию можно узнать в какой «ячейке» хранится информация, если получится так, что в одной «ячейке» несколько слов, то искомое можем найти простым перебором. Что куда быстрее чем перебирать целый словарь. Ситуация, когда в одной «ячейке» сразу несколько слов – называется коллизией.

Коллизия – это явление, когда при разных входных данных, хеш-функция возвращает, одно и то же число – хеш-код. От коллизий избавится невозможно, но в “хороших” хеш-функциях вероятность их возникновения стремится к теоретическому минимуму.

Но не менее важной характеристикой хэш-функций является распределение.

это мера, которая отражает разброс количества коллизий, на разные значения хеш-кода.

Примеры с хорошим и плохим распределением, (рисунок 1), (рисунок 2).

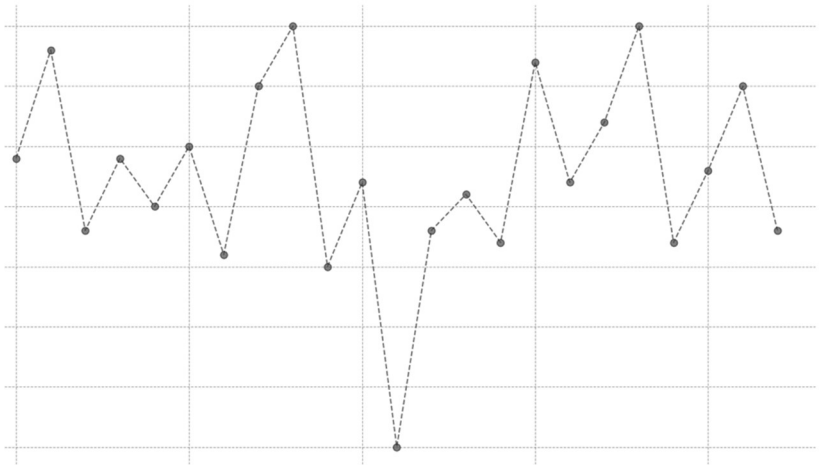

Рисунок 8 – Пример плохого распределения

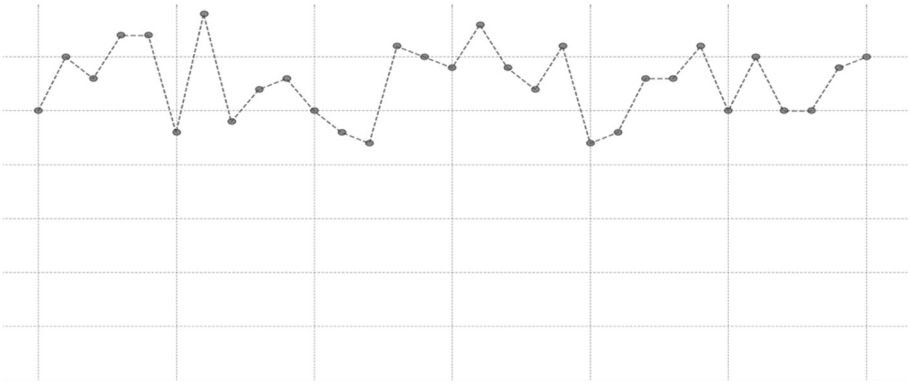

Рисунок 2 – Пример хорошего распределения

Большинство популярных хэш-функции имеют сложность O(n) и гарантируют минимальное число коллизий. Рассмотрим хэш-функцию со сложностью

O(1) — f(x) = x mod N. Это одна из самых простых хэш-функций. Но распределение в ней зависит от выбора N.

Сформулируем теорему: Арифметическая прогрессия вида S = a + km дает больше возможных остатков от деления. Докажем:

— Для составных N (где d = НОД(m, N) > 1):

Прогрессия порождает только N / d уникальных остатков.

Пример: при N = 6, m = 2, остатки 1, 3, 5 повторяются

— Для простых N:

Так как m и N взаимно просты, последовательность km mod N дает полную перестановку остатков.

Пример: N=5, m=2, следовательно, остатки 0, 1, 2, 3, 4 покрывают весь диапазон.

Таким образом, простые числа являются оптимальным хэш-ключом.

Впервые о прогрессиях простых чисел заговорили в 1770 году, а именно в переписке Лагранжа и Варинга. Однако первые достижения в этой области были сделаны лишь в 1938 году, Николаем Григорьевичем Чудаковым который доказал, что множество простых чисел содержит бесконечно много арифметических прогрессий длиной три. Вопрос о прогрессиях длиной более трех, до недавнего времени оставался открытым.

В 2004 году Бен Грин и Теренс Тао доказали существование арифметических прогрессий произвольной длины в множестве простых чисел. В своем доказательстве они использовали комбинацию идей из разных математических областей.

Теорема Грина-Тао утверждает, что последовательность простых чисел содержит арифметические прогрессии произвольной длины.

Самая длинная такая прогрессия на момент написания статьи имеет 27 элементов. Для ее нахождения понадобились тысячи компьютеров волонтеров. И нашли ее в 2019 году.

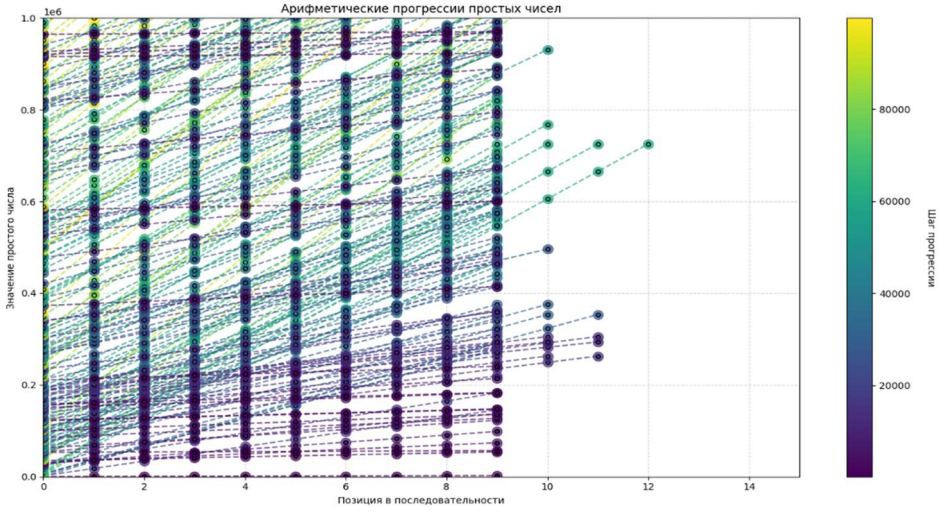

Рисунок *9* – Арифметические прогрессии в множестве простых чисел

На данном графике (рисунок 3) изображен результат работы разработанного алгоритма, по поиску арифметических прогрессий в множестве простых чисел. В данном случае была найдена прогрессия длиной в 12 элементов.

После полученных результатов была выдвинута гипотеза о том, что арифметические прогрессии простых чисел могут применяться при хешировании вида x mod N.

Для сравнения эффективности хеш-функции для различных N, очень важным критерием является распределение.

Для сравнения эффективности распределения хеш-функции, было решено использовать формулу Дисперсии.

Дисперсия – это мера, которая показывает разброс величины, относительно ее математического ожидания и вычисляется по формуле:

$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\overline{x})^2}{n-1}}$, где x – математическое ожидание, xi – текущее значение,

n - количество экспериментов.

В данном эксперименте в качестве математического ожидания выступает идеальное распределение, то есть, когда на разные значения хеш-кода приходится одинаковое количество коллизий. А в качестве текущего значения – количество коллизий, приходящееся на одно значение хеш-кода.

Сравним два метода расширения хеш-таблиц. Допустим мы хешируем методом x mod N и нам понадобилось увеличивать N постепенно (для экономии памяти), тогда это можно сделать двумя методами.

Было написано программное средство и проведен тест, для которого использовали миллиард случайно сгенерированных чисел. Которые генерировались в диапазоне 0 - 655 миллионов. Тем самым имитируя поток входных данных.

Таблица 1 – Пример распределения двух методов

Первый метод		Второй метод	
N	Распределение	N	Распределение
3701	13.760%	4001	12.566%
6221	6.387%		
8741	3.823%	8009	4.391%
11261	2.651%		
13781	1.945%	16007	1.565%
16301	1.522%		

1-й метод – это заранее найдена арифметическая прогрессия простых чисел

2-й метод – это заранее найдены числа для расширения, путем умножения начального числа на два и поиска ближайшего простого числа.

Видно, что процент распределения сравним, а значит главное отличия методов: использование памяти и трудность нахождения нового N. Но возникает проблема. Так как заметно, что с ростом числа N, распределение улучшается, так как количество возможных остатков от деления (хеш-кодов) возрастает. Но и различия для приблизительно равных чисел минимальны, что и приводит к тому, что мы можем подобрать любые простые числа, которые больше предыдущих. Разница лишь в скорости роста хеш-таблицы.

А также арифметические последовательности конечны, но очень просто программно реализуемы, то есть нужно знать только шаг, количество элементов и начальное число прогрессии.

Так же можно применять любые закономерности простых чисел, где арифметическая прогрессия может выступать только способом их нахождения. Но в некоторых случаях, когда мы знаем входные данные, можно подобрать «хорошую» хеш-функцию. Где арифметическая последовательность простых чисел может стать оптимальным решением.

Список использованных источников:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. Introduction to Algorithms. *– 3rd ed. – Cambridge: MIT Press, 2009. – 253 p.*

2. Knuth, D. E. The Art of Computer Programming, Volume 3: Sorting and Searching. *– 3rd ed. – Boston: Addison-Wesley, 1998. – 744 p.*

3. Stinson, D. R. Cryptography: Theory and Practice. *– 3rd ed. – Boca Raton: CRC Press, 2006. – 137 p.*

4. Granville, A., P. Primes in Arithmetic Progressions. *– In:* Mathematical Surveys and Monographs*, 2007.*

5. Как работает хэширование? [Электронный ресурс]. – Режим доступа: https://habr.com/en/companies/ruvds/articles /747084/

UDC 512.12:004.424.47

USE OF ARITHMETIC PROGRESSIONS OF PRIME NUMBERS IN FACTORING ON THE REMAINDER FROM DIVISION

Klimintsiyonak V.S, Kuzhik D.S.

Belarusian State University of Informatics and Radioelectronics[1], Minsk, Republic of Belarus

Barkova E.A. – PhD in Physics and Mathematics

Annotation. The main objective of this scientific work is to study and optimize hashing methods using arithmetic progressions of primes. The article considers hash functions, their influence on hashing speed and accuracy, as well as the problem of conflicts. New algorithms have been developed: the algorithm for finding arithmetic progressions of prime numbers and the algorithm for evaluating the distribution of hash functions.

Keywords. Hashing, hash function, hash table, collision, distribution, arithmetic progression, Green-Tao theorem, dispersion, efficiency, optimization, algorithm, simple numbers.