# Полиномиальный хеш

Отчет по семестровой работе

**Автор**:

*Белов Макар*

1)Определение и применение

Хеш-код – целое число, вычисляемое по символам строки. Если две строки равны, то их хеш-коды тоже равны. Хеширование строк позволяет эффективно отвечать на вопрос о равенстве строк, сравнивая их хеш-коды.

Также этот алгоритм можно использовать для нахождения подстроки в строке, z-функций, лексикографически минимального циклического сдвига строки, всех палиндромов в строке и для сравнения подматриц в матрицах.

2)Принцип работы

Рассмотрим полиномиальное хеширование строк, при котором хеш-функция вычисляется как перевод из n-ичной системы в десятичную. Пусть дана строка **s**, основание **BASE** и кольцо вычетов по модулю **MOD**.

Тогда хеш-код строки вычисляется следующим образом:

**(s[0] \* BASE0 + s[1] \* BASE1 + ... + s[n – 1] \* BASE(n-1)) % MOD**

Пусть максимальная длина строки **SIZE** – 106. Заведём массив **hash[SIZE]**, хранящий 64-битовые числа. Далее заполняем массив простейшей динамикой по вышеописанным формулам.

Также понадобится массив степеней выбранного **BASE**. Заведём **powers[SIZE]**, хранящий 64-битовые числа и заполним его по динамике: **powers[i] = powers[i – 1] \* BASE % MOD**.

3)Оценка сложности

Так как хеш — это всего лишь число, для их сравнения потребуется O(1) времени.

Поиск подстроки в строке за O(n + m).

Нахождение z-функции за O(n log n).

Поиск лексикографически минимального циклического сдвига строки за O(n log n).

Поиск всех палиндромов в строке за O(n log n).

**4)График**

**5)Плюсы и минусы**

Главный недостаток полиномиального хеширования – это коллизии.

Коллизия – ситуация, когда строки по факту различны, но их хеши совпадают. В таком случае алгоритм заключает, что строки одинаковы, хотя на самом деле это не так.

Избавиться от коллизий при длине строк ~106 невозможно, потому что количество различных строк больше количества различных хеш-кодов. Вероятность коллизии можно свести к минимуму (почти к нулю), если написать ещё один хеш, т. е. написать первый хеш с основанием 2011 по модулю 109 + 7, а второй хеш – с основанием 2017 по модулю 109 + 9 и использовать оба хеша в сравнениях.

Главным плюсом алгоритма является его скорость работы. При помощи полиномиального хеширования мы можем сравнить две строки за О(1), в то время как линейный поиск работает за О(n).

Ссылка на презентацию:

<https://docs.google.com/presentation/d/1kd19KFmazUFWARvMgnLcQPkarH8GYr8hwn0p_oR6A5s/edit?usp=sharing>