



Санкт-Петербургский государственный университет

Программа «Технологии программирования»

Алгоритм Рабина-Карпа

Касаткина Анна Николаевна, 23.Б10

Санкт-Петербург
2023

Алгоритм Рабина-Карпа

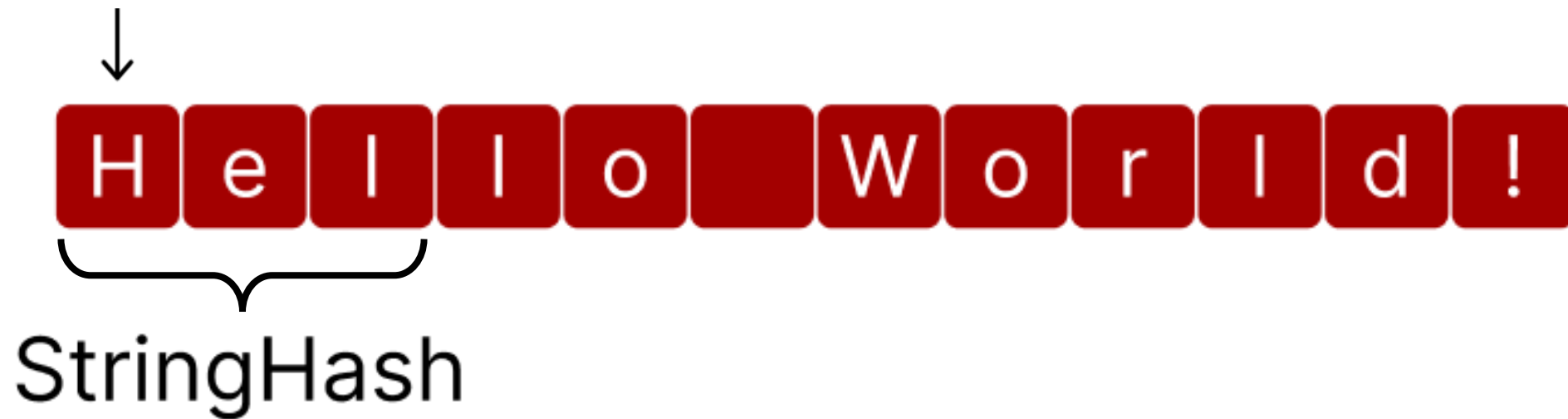
- Был разработан в 1987 году Михаэлем Рабином и Ричардом Карпом
- Предназначен для поиска подстроки в тексте, используя хеширование
- Одно из простейших практических применений алгоритма Рабина — Карпа состоит в определении плагиата

- Первый шаг - вычисление хеша искомого шаблона

Н e l l o W o r l d !

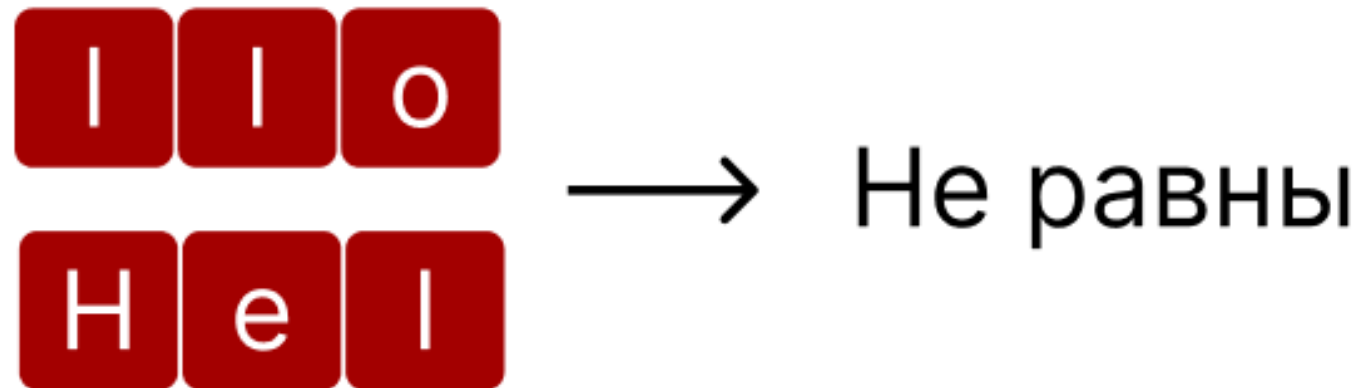
l l o → PatternHash

- Вычисляем хеш первой подстроки



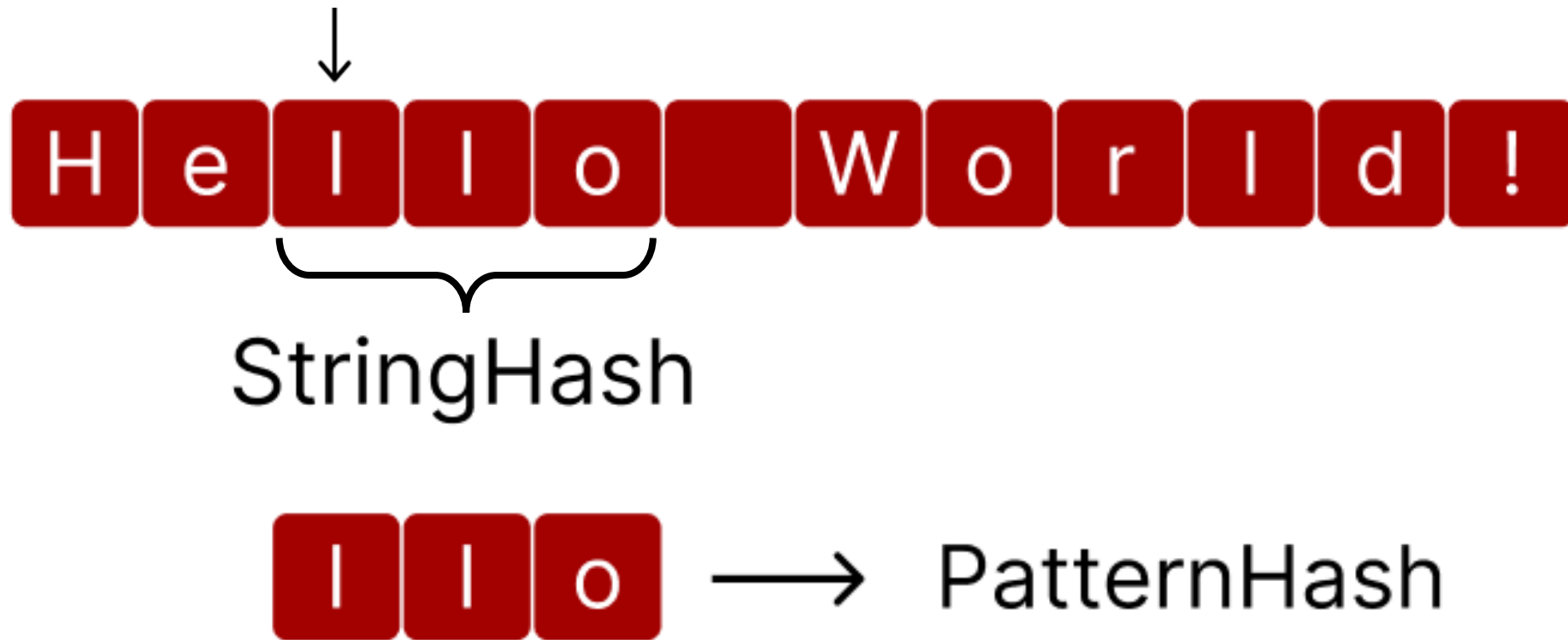
l l o → PatternHash

- Сравниваем значения хеша подстроки и шаблона
- Если значения равны, посимвольно сравниваем строки для исключения возможности коллизии



l l o
H e I → Не равны

- Продолжаем работу до следующего равенства



Скользящий хеш

- Скользящий хеш (или кольцевой) — хеш-функция, вычисляемая на основе части диапазона входных данных, так что при сдвиге диапазона вычисление хеша будет зависеть от ранее вычисленного хеша и проходить гораздо быстрее.

$$\text{hash}(p[1..m]) = \left(\sum_{i=1}^m p[i] x^{m-i} \right) \bmod q$$

Хеш-значения последовательных подстрок $s[i..i+m-1]$ и $s[i+1..i+m]$

$$\text{hash}(s[i+1..i+m]) = ((\text{hash}(s[i..i+m-1]) - s[i] \cdot x^{m-1}) \cdot x + s[i+m]) \bmod q$$

Скользящий хеш

$$\boxed{\text{H}}\boxed{\text{e}}\boxed{\text{l}} = \boxed{\text{l}} + X \left(\boxed{\text{e}} + X * \boxed{\text{H}} \right)$$

$$\boxed{\text{e}}\boxed{\text{l}}\boxed{\text{l}} = \boxed{\text{l}} + X \left(\boxed{\text{l}} + X * \boxed{\text{e}} \right)$$

$$\boxed{\text{l}}\boxed{\text{l}}\boxed{\text{o}} = \boxed{\text{o}} + X \left(\boxed{\text{l}} + X * \boxed{\text{l}} \right)$$

Скользящий хеш

$$\boxed{\text{H}}\boxed{\text{e}}\boxed{\text{I}} = \boxed{\text{I}} + X (\boxed{\text{e}} + X * \boxed{\text{H}})$$

$$\boxed{\text{e}}\boxed{\text{I}}\boxed{\text{I}} = (\boxed{\text{H}}\boxed{\text{e}}\boxed{\text{I}} - \boxed{\text{H}} * X^2) * X + \boxed{\text{I}}$$

↑
Удаляемый элемент

↑
Добавляемый элемент

Скользящий хеш

- Таким образом для вычисления нового хеша на основе старого требуется:
 1. Из старого хеша вычесть значение $s[1] \cdot x^{m-1}$
 2. Полученный результат умножить на x
 3. Прибавить значение $s[m+1]$
 4. Вычислить остаток от деления на q

- Алгоритм редко используется для поиска одиночного шаблона, эффективен в поиске совпадений множественных шаблонов одинаковой длины
- При хорошей хеш-функции работает за гарантированное время $O(n)$ и вероятность ошибки можно сделать очень малой.

Для текста длины n и шаблона длины m :

- Среднее и лучшее время исполнения равно $O(n)$
- В худшем случае $O(n*m)$