**Алгоритм Рабина-Карпа**

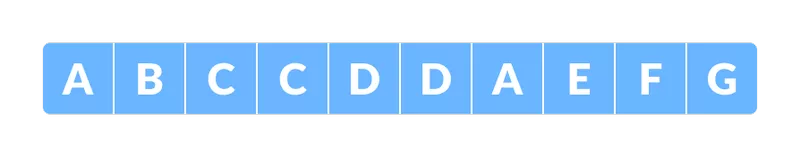
**Алгоритм Рабина-Карпа** — это алгоритм, используемый для поиска / сопоставления шаблонов в тексте с использованием хеш-функции. В отличие от наивного алгоритма сопоставления строк, он не проходит через каждый символ на начальном этапе, а фильтрует символы, которые не совпадают, а затем выполняет сравнение.

**Хеш-функция** — это инструмент для сопоставления большего входного значения меньшему выходному значению. Это выходное значение называется хеш-значением.

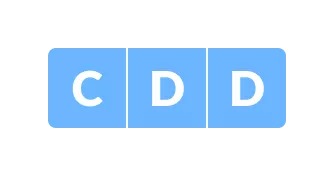
**Как работает алгоритм Рабина-Карпа?**

Берется последовательность символов и проверяется возможность наличия искомой строки. Если возможность найдена, выполняется сопоставление символов.

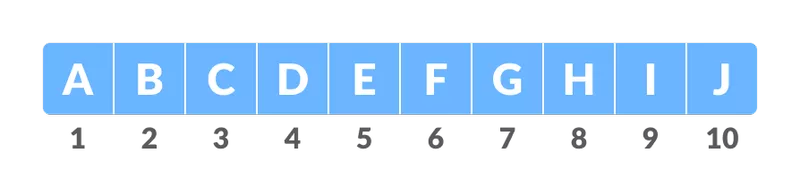
1. Пусть текст будет:



И строка, которую нужно искать в приведенном выше тексте, будет:

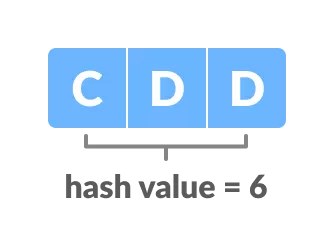


1. Назначим numerical value(v)/weight символам, которые мы будем использовать в задаче. Здесь мы взяли только первые десять алфавитов (то есть от A до J).



1. m - длина шаблона, а n - длина текста. Здесь m = 10 и n = 3.

Пусть d будет количеством символов во входном наборе. Здесь мы взяли набор входных данных (A, B, C,…, J). Так, d = 10. Вы можете принять любое подходящее значение для d.



1. Рассчитаем хеш-значение паттерна. Хеш-значение текста

хэш-значение для шаблона (p) = Σ (v \* dm-1) mod 13 = ((3 \* 10 2 ) + (4 \* 10 1 ) + (4 \* 10 0 )) mod 13 = 344 mod 13 = 6

В приведенном выше вычислении выберите простое число (здесь 13) таким образом, чтобы мы могли выполнять все вычисления с арифметикой с одинарной точностью.

1. Вычислить хеш-значение для текстового окна размером m.

Для первого окна ABC хэш-значение для текста (t) = Σ (v \* dn-1) mod 13 = ((1 \* 10 2 ) + (2 \* 10 1 ) + (3 \* 10 0 )) mod 13 = 123 мод 13 = 6

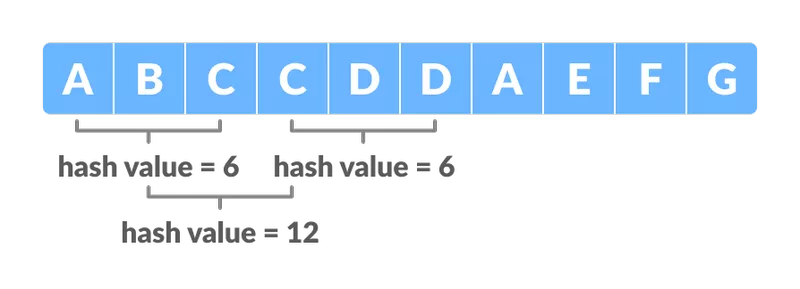
1. Сравнить хеш-значение шаблона с хеш-значением текста. Если они совпадают, то выполняется сопоставление символов.  
   В приведенных выше примерах хэш-значение первого окна (т.е. t) совпадает с p, поэтому выберите соответствие символов между ABC и CDD. Поскольку они не совпадают, перейдите в следующее окно.
2. Вычисляем хеш-значение следующего окна, вычитая первый член и добавляя следующий член, как показано ниже.

t = ((1 \* 10 2 ) + ((2 \* 10 1 ) + (3 \* 10 0 )) \* 10 + (3 \* 10 0 )) mod 13 = 233 mod 13 = 12

Чтобы оптимизировать этот процесс, мы используем предыдущее значение хеш-функции следующим образом.

t = ((d \* (t - v (символ, который нужно удалить) \* h) + v (символ, который нужно добавить)) mod 13 = ((10 \* (6-1 \* 9) + 3) mod 13 = 12 Где , h = d м-1 = 10 3-1 = 100.

1. Для BCC, т = 12 ( ≠ 6). Поэтому переходите к следующему окну.  
   После нескольких поисков мы получим совпадение с окном CDA в тексте.



https://ru.wiki-base.com/7779856-rabin-karp-algorithm