Зверев Н.Д.

1-ИСП11-18

**Введение**

Алгоритм Рабина — Карпа — это алгоритм поиска строки, который ищет шаблон, то есть подстроку, в тексте, используя хэширование. Он был разработан в 1987 году Майклом Рабином и Ричардом Карпом.

Ричард Мэннинг Карп (род. 3 января 1935 года, Бостон, США) — американский учёный в области теории вычислительных систем, лауреат премии Тьюринга.

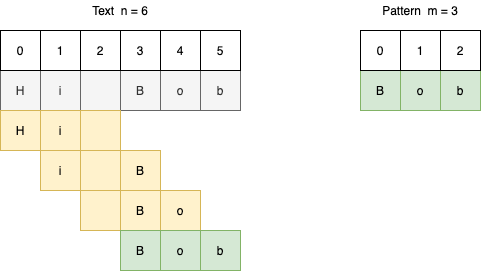
Михаэль Озер Рабин (род. 1 сентября 1931, Вроцлав) — израильский учёный в области теории вычислительных систем, математик, лауреат премии Тьюринга и многих других премий.

Алгоритм редко используется для поиска одиночного шаблона, но имеет значительную теоретическую важность и очень эффективен в поиске совпадений множественных шаблонов одинаковой длины. Для текста длины n и шаблона длины m его среднее и лучшее время исполнения равно O(n) при правильном выборе хэш-функции (смотрите ниже), но в худшем случае он имеет эффективность O(n\*m), что является одной из причин того, почему он не слишком широко используется.

Наивный алгоритм поиска строки сравнивает заданный шаблон со всеми позициями в тексте. Это приводит к далеко не идеальной сложности времени исполнения O(nm), где n = длина текста, а m = длина шаблона.

Алгоритм Рабина-Карпа совершенствует этот подход за счёт того, что сравнение хэшей двух строк выполняется за линейное время: для поиска совпадения это гораздо эффективнее, чем сравнение отдельных символов этих строк. Таким образом, алгоритм показывает лучшее время исполнения O(n+m).

Алгоритм Рабина-Карпа



1. Вычисляется хэш шаблона строки.
2. Вычисляется хэш подстроки в тексте строки, начиная с индекса 0 и до m-1.
3. Сравнивается хэш подстроки текста с хэшем шаблона.

Если они совпадают, то сравниваются отдельные символы для выявления точного совпадения двух строк.

Если они не совпадают, то окно подстроки сдвигается путём увеличения индекса и повторяется третий пункт для вычисления хэша следующих m символов, пока не будут пройдены все n символов.

**Хэш-функция**

С применением хеш-функции связаны два нюанса.

Во-первых, алгоритм хорош настолько, насколько хороша его хеш-функция. Если при использовании хеш-функции имеют место многочисленные ложные срабатывания, то сравнение символов будет выполняться слишком часто. В этом случае очень сложно считать этот метод более эффективным, чем наивный алгоритм.

Во-вторых, каждый раз, когда подстрока проходит по тексту, вычисляется новый хэш, что крайне неэффективно, ведь в этом случае производительность такая же, если не хуже, как у наивного алгоритма.

Обе эти проблемы решаются с помощью полиномиального хэша с операциями сложения и умножения. И хотя это не какой-то эксклюзив алгоритма Рабина-Карпа, которого нет в других алгоритмах, здесь он работает так же хорошо.

**Сравнение с другими алгоритмами**

**Преимущества:**

1. Быстрая скорость работы — O(n+m), где n — длина строки, m — длина образца;
2. Простая и понятная реализация;

**Недостатки:**

1. Возможно подобрать входные данные так, что количество ложных срабатываний будет недопустимо большим.

**Источники:**

1) Андрей Шагин «Алгоритм Рабина-Карпа с полиномиальным хэшем и модульной арифметикой»

2) CoderLessons.com Статья: «Алгоритм недели: поиск строки Рабина-Карпа»

3) Nuancesprog.ru Статья: «Алгоритм Рабина-Карпа с полиномиальным хэшем и модульной арифметикой»