Алгоритм Рабина-Карпа

1. **Алгоритм Рабина Карпа**

Алгоритм Рабина — Карпа — это алгоритм поиска строки, который ищет шаблон, то есть подстроку, в тексте, используя хеширование. Если говорить о наивном алгоритме сопоставления строк, то он сравнивает данный шаблон со всеми позициями в данном тексте. Каждое сравнение занимает время, пропорциональное длине шаблона, а количество позиций пропорционально длине текста. Следовательно, наихудшее время для такого метода пропорционально произведению двух длин. Конечно, во многих случаях это время можно значительно сократить, прерывая сравнение, как только обнаруживается несоответствие, но эта идея не может гарантировать какого-либо ускорения.

Алгоритм Рабина–Карпа же достигает своего ускорения за счет использования скользящего хэша, что позволяет быстро отфильтровывать позиции текста, которые не могут соответствовать шаблону, а затем проверяет совпадение в оставшихся позициях. Обобщения одной и той же идеи могут быть использованы для поиска более чем одного совпадения с одним шаблоном или для поиска совпадений для более чем одного шаблона.

Одно из простейших практических применений алгоритма Рабина — Карпа состоит в определении плагиата. Алгоритм Рабина — Карпа может довольно быстро найти в проверяемом тексте примеры вхождения некоторых предложений из исходных материалов. Для устранения чувствительности алгоритма к небольшим различиям можно игнорировать детали, такие как регистр или пунктуация, при помощи их удаления. Поскольку количество строк, которые мы ищем, k, очень большое, обычные алгоритмы поиска одиночных строк становятся неэффективными.

1. **Скользящий хеш**

Для алгоритма Рабина — Карпа важными являются низкая вероятность коллизий и эффективное вычисление хеш-значения последовательных подстрок текста.

Для быстрого вычисления хеша Рабин и Карп предложили использовать скользящий хеш, а именно полиномиальный.

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.

1. **Сложность**

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 хеш-функции (смотрите ниже), но в худшем случае он имеет эффективность O(nm), что является одной из причин того, почему он не слишком широко используется. Для приложений, в которых допустимы ложные срабатывания при поиске, то есть, когда некоторые из найденных вхождений шаблона на самом деле могут не соответствовать шаблону, алгоритм Рабина — Карпа работает за гарантированное время O(n) и при подходящем выборе рандомизированной хеш-функции (смотрите ниже) вероятность ошибки можно сделать очень малой.