**18. Поиск в строке. Алгоритм Рабина-Карпа.**

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

Вместо того, чтобы использовать более умный пропуск, алгоритм Рабина-Карпа пытается ускорить проверку эквивалентности образца с подстроками в тексте, используя хеш-функцию. Хеш-функция — это функция, преобразующая каждую строку в числовое значение, называемое хеш-значением (хеш); например, мы можем иметь хеш от строки «nest» равным 442 (сумма ASCII кодов ‘n’ – 110, ‘e’ – 101, ‘s’ – 115, ‘t’– 116). Алгоритм использует тот факт, что если две строки одинаковы, то и их хеш-значения также одинаковы. Таким образом, всё что нам нужно, это посчитать хеш-значение искомой подстроки и затем найти подстроку с таким же хеш-значением.

Однако существуют две проблемы, связанные с этим. Первая состоит в том, что, так как существует очень много различных строк, между двумя различными строками может произойти коллизия — совпадение их хешей. Например, слово **sent** имеет тот же код , что и **nest**. При использовании достаточно хороших хеш-фунцкий коллизии случаются крайне редко.

Вторая проблема заключается в пересчитывании хеша. Метод решения данной проблемы состоит в предположении того, что хеш-значение подстроки s[i+1..i+m]  уже содержит хеш-значение подстроки s[i..i+m-1]. Если использовать его для подсчёта следующего хеш-значения за постоянное время, тогда проблема будет решена.

Это достигается использованием так называемого [кольцевого хеша](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D1%8C%D1%86%D0%B5%D0%B2%D0%BE%D0%B9_%D1%85%D0%B5%D1%88). Самым простым примером кольцевого хеша является добавление значений каждого следующего символа в подстроке и последующее использование данной формулы для подсчёта каждого следующего хеш-значения за фиксированное время:

s[i+1..i+m] = s[i..i+m-1] - s[i] + s[i+m]

Одна популярная и эффективная кольцевая хэш-функция интерпретирует каждую подстроку как число в некоторой системе счисления, основание которой является большим [простым числом](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE). Например, если подстрока «**nest**» и основание системы счисления 101, хеш-значение будет

110 × 1013 + 101 × 1012 + 113 × 1011 + 116 × 1010 =114374940

Очевидно, что оно не совпадёт с хеш-значением подстроки «**sent**».

Например, если мы имеем текст «***abracadabra***» и ищем образец длины ***3***, мы можем рассчитать ***хэш*** подстроки «***bra***» из хэша подстроки «***abr***» (предыдущая подстрока), вычитая число добавленное для первой буквы 'a' из «abr», то есть 97×1012  (97 — ASCII для 'a' и 101 — основание, которое мы используем), умножение на основание и наконец добавляя последнее число для «bra», то есть 97 × 1010  = 97. Если подстроки в запросе длинны, этот алгоритм достигает большой экономии сравнимо с многими другими схемами хеширования.