

Отчет по тестированию алгоритмов поиска подстроки

Постановка задачи

Существует несколько алгоритмов поиска подстроки в строке. Необходимо сравнить их по производительности.

Параметры вычислительного узла

Операционная система: Windows 11

Процессор: AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz

Оперативная память: 16,0 ГБ

Описание тестируемых алгоритмов

Брутфорс

Прямой поиск подстроки. Он самый простой в реализации. Алгоритм сопоставляет искомую строку с частью исходной строки такой же длины, начиная с нулевого символа. Если эта попытка завершилась неудачей, сопоставление производится с первого символа и так далее.

Ахо-Корасик

Алгоритм строит конечный автомат, которому затем передаёт строку поиска. Автомат получает по очереди все символы строки и переходит по соответствующим рёбрам. Если автомат пришёл в конечное состояние, соответствующая строка словаря присутствует в строке поиска. Алгоритм хорошо себя показывает при поиске нескольких паттернов одновременно.

Рабин-Карп

Алгоритм напоминает по реализации Брутфорс, но сравнивает не сами строки, а их хеш.

Кнут-Моррис-Пратт

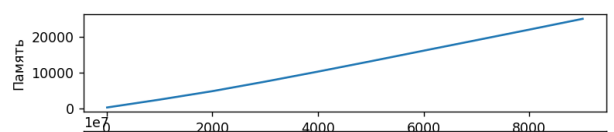
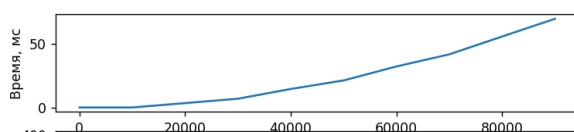
Основное отличие алгоритма от Брутфорса заключается в том, что сдвиг строки выполняется не на один символ на каждом шаге алгоритма, а на некоторую длину не подошедшей части подстроки, которая вычисляется при сравнении. Это позволяет ускорить поиск, избегая ненужных итераций.

Бойер-Мур-Хорспул

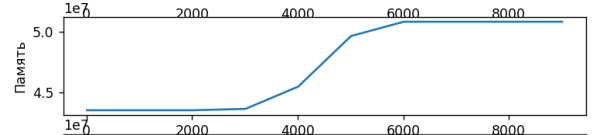
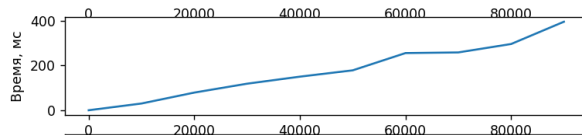
Упрощенная версия алгоритма Бойера-Мура. Строится таблица смещений для каждого символа, затем исходная строка и шаблон совмещаются по началу, сравнение ведется по последнему символу. Если последние символы совпадают, то сравнение идет по предпоследнему символу и так далее. Если же символы не совпали, то шаблон смещается вправо, на число позиций, взятое из таблицы смещений для символа из исходной строки, и тогда снова сравниваются последние символы исходной строки и шаблона. И так далее, пока шаблон полностью не совпадет с подстрокой исходной строки, или не будет достигнут конец строки.

Результат измерений

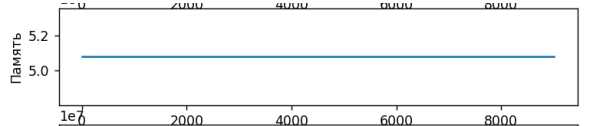
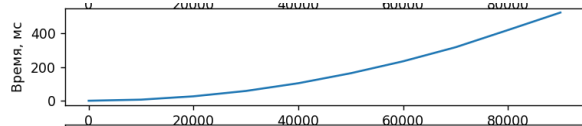
Брутфорс



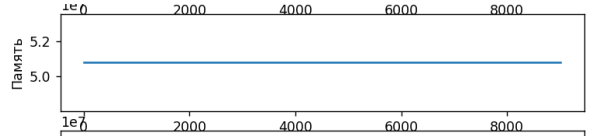
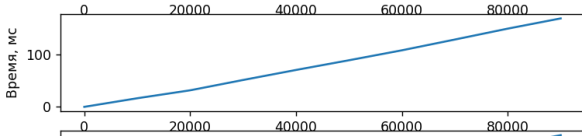
Ахо-Корасик



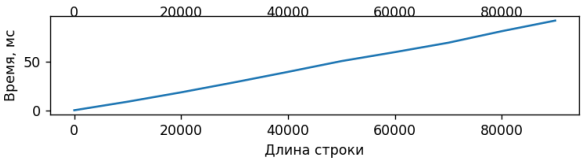
Рабин-Карп



Кнут-Моррис-Пратт



Бойер-Мур-Хорспул



Обоснование результатов

Брутфорс – квадратичная зависимость

Ахо-Корасик – линейная зависимость

Рабин-Карп – линейная зависимость

Кнут-Моррис-Пратт – линейная зависимость

Бойер-Мур-Хорспул – линейная зависимость

Анализ результатов

Брутфорс оказался неэффективным по использованию памяти, хоть и самым быстрым по времени.

Ахо-Корасик оказался самым экономным по затратам памяти, но заметно медленнее того же Брутфорса.

Рабин-Карп практически не уступает Ахо-Корасику.

А вот алгоритм Бойера-Мура-Хорспула оказался хоть и чуть медленнее Брутфорса, но это отличие незначительно в сравнении с затратами по памяти.

Из полученных данных можем сделать вывод, что оптимальным из представленных алгоритмов является Бойер-Мур-Хорспул.