

# Отчет

## Постановка задачи:

Реализовать несколько алгоритмов поиска подстроки в строке и сравнить их по производительности, использованию памяти.

## Параметры вычислительного узла:

Операционная система: Windows 11

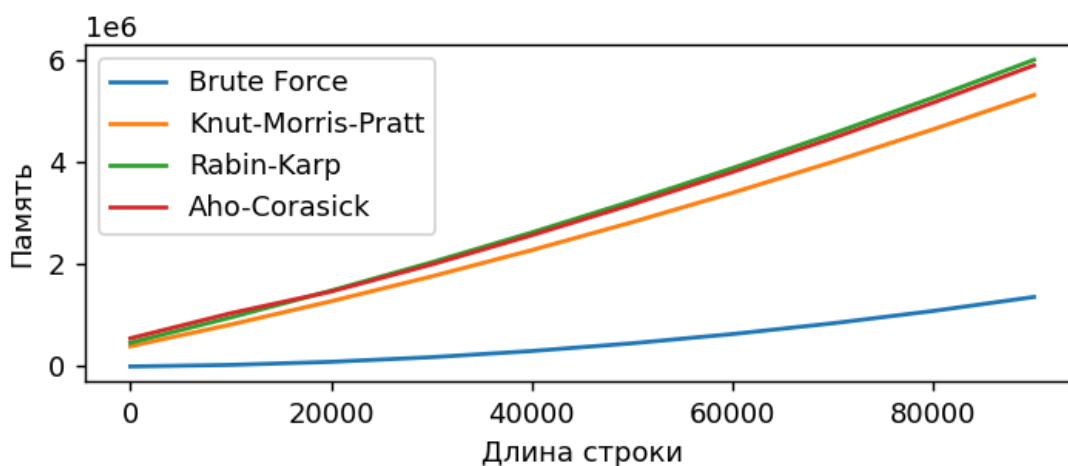
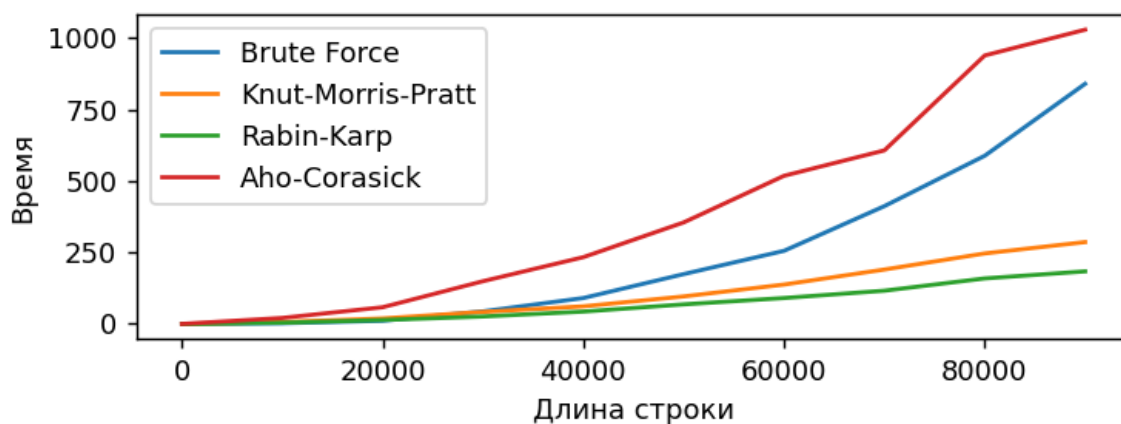
Процессор: intel Core i5 -1155G7

Оперативная память: 8 гб

## Описание алгоритмов:

- 1) БрутФорс – метод поиска в строке подстроки напрямую, идет сравнение каждого символа подстроки с символами строки пока не произойдет совпадение.
- 2) Рабин Карп – работает с использованием хэш функции. Символы строки и подстроки переводим в хэши, считаем хэш подстроки и хэш подстроки в строке длиной равной искомой подстроке(окно), сравниваем, если хэши равны, то сравниваем сами символы.
- 3) Кнут-Моррис-Патт – сравнивает префикс функции искомой подстроки и строки. Если значение префикс функции будет равно длине подстроки, то подстрока найдена.
- 4) Ахо-Корасик – строит для набора строк бор(дерево) в общем случае, по бору строит автомат. Затем алгоритм передает автомату все символы искомой подстроки. Под воздействием какой-то входной буквы переходим в другое состояние — т.е. в другую позицию в наборе строк. Автомат приходит в конечное состояние - строка есть в строке поиска.

## Результаты:



## Обоснование результатов:

### Сложности агоритмов

- брутфорс имеет квадратичную зависимость
- рабин карп, кнут-моррис-пратт и ахо-корасик линейную зависимость

Худшим алгоритмом оказался брутфорс, имеющий квадратичную зависимость. Лучшими по памяти являются алгоритмы Рабин Карп и Ахо Корасик. Лучшими по времени является Рабин Карп.