# Сравнение производительности алгоритмов поиска подстроки

Сарнацкий Михаил, ФТ-101-1

## 1 Постановка задачи

Задача заключается в сравнении производительности четырёх алгоритмов поиска подстроки — наивного, Кнута-Морриса-Пратта (КМР), Бойера-Мура и Рабина-Карпа — на наборах данных различного размера и типов (лучший, худший, случайный). Цель — измерить среднее время выполнения (в наносекундах) для фиксированной длины шаблона в 15 символов, проанализировать эффективность на данных размером от 1 КБ до 500 МБ.

## 2 Параметры системы

Тестирование проводилось на VPS timeweb.cloud со следующими характеристиками:

- **Конфигуратор**: 4 x 3.3 ГГц CPU, 16 ГБ RAM, 20 ГБ NVMe.
- **Образ**: Ubuntu 24.04.

Все вычисления были вынесены на VPS для минимизации влияния фоновых процессов на работу скрипта.

## 3 Описание алгоритмов

Для характеризации временных слостей алгоритмов обозначим $m$ длиной шаблона, а $n$ — длиной текста.

1. **Наивный алгоритм**: Простой подход, проверяющий каждую возможную позицию в тексте на совпадение с шаблоном. Худшая временная сложность — $O(m \cdot n)$.
2. **Кнута-Морриса-Пратта (КМР)**: Использует префикс-функцию для пропуска повторных сравнений, достигая сложности $O(n + m)$.
3. **Бойера-Мура**: Применяет эвристики плохого символа и хорошего суффикса для пропуска больших участков текста. Лучшая сложность — $O(n/m)$, худшая — $O(n \cdot m)$.
4. **Рабина-Карпа**: Использует хеширование для сравнения подстрок. Средняя сложность — $O(n+m)$, но в худшем случае возможна $O(n \cdot m)$ (из-за коллизий хешей).

# 4 Результаты

Данные тестирования включают измерения для размеров данных от 1 КБ до 500 МБ, трёх типов случаев: лучший, худший, случайный. Лучший и худший типы отобраны из отдельно для каждого из алгоритмов. Длина шаблона всезде составляла 15 символов. Каждый тест выполнялся 30 раз для обеспечения достоверности. Ниже приведена таблица со средним временем выполнения (в наносекундах) для выбранных размеров данных и типов случаев.

Таблица 1: Среднее время выполнения (нс) для выбранных размеров данных и типов случаев

| Алгоритм | Случай | 1 КБ | 100 КБ | 1 МБ | 100 МБ |
|---|---|---|---|---|---|
| Наивный | Лучший | 597.5 | 596.3 | 771.0 | 818.1 |
| | Худший | 3.32e5 | 1.24e8 | 1.38e9 | 1.24e11 |
| | Случайный | 1.01e5 | 1.47e8 | 1.35e9 | 1.12e11 |
| КМР | Лучший | 4.71e3 | 6.19e3 | 3.63e3 | 5.68e3 |
| | Худший | 7.76e5 | 2.55e8 | 1.61e9 | 1.37e11 |
| | Случайный | 2.54e6 | 9.53e7 | 1.06e9 | 3.63e10 |
| Бойера-Мура | Лучший | 2.38e4 | 2.70e7 | 2.74e8 | 2.66e10 |
| | Худший | 3.12e4 | 3.81e7 | 3.24e8 | 3.90e10 |
| | Случайный | 3.24e4 | 2.78e7 | 3.05e8 | 3.54e10 |
| Рабина-Карпа | Лучший | 6.57e3 | 5.81e3 | 5.37e3 | 9.38e3 |
| | Худший | 4.42e6 | 4.52e8 | 3.76e9 | 3.34e11 |
| | Случайный | 6.66e6 | 3.72e8 | 4.18e9 | 3.16e11 |

Графики, отображающие среднее время выполнения в зависимости от размера данных для каждого алгоритма и типа случая, предоставлены отдельно. Графики приведены в двухлогарифмическом масштабе для удобства просматривания. Каждая точка сопровождается вертикальным отрезком, представляющим доверительный интервал, рассчитанный с использованием стандартного отклонения и уровня доверия 95%.

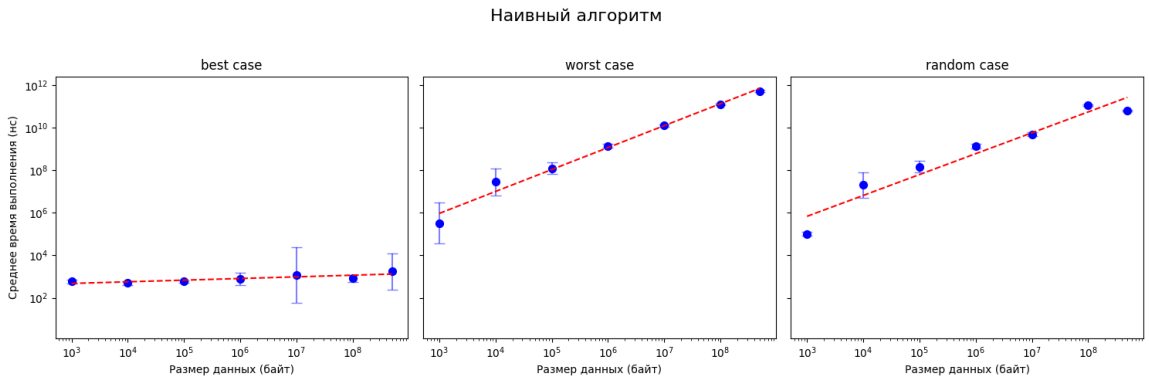


Рис. 1: Время работы наивного алгоритма

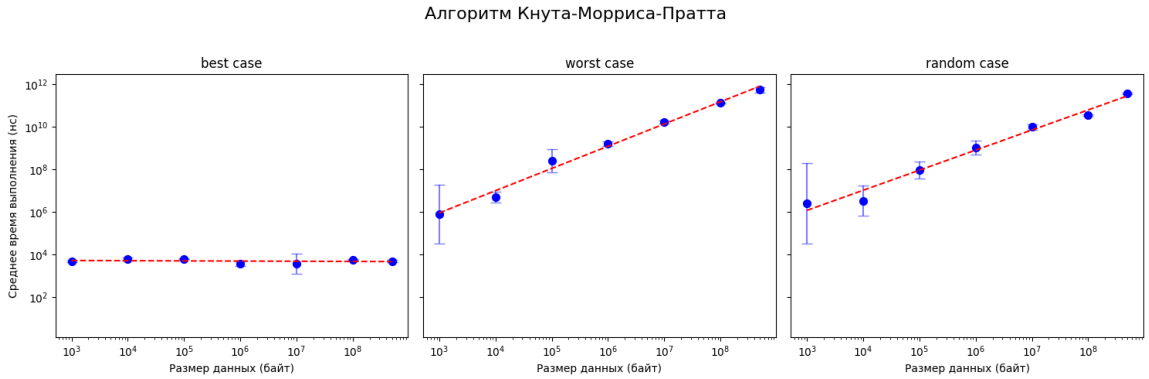


Рис. 2: Время работы алгоритма КМР

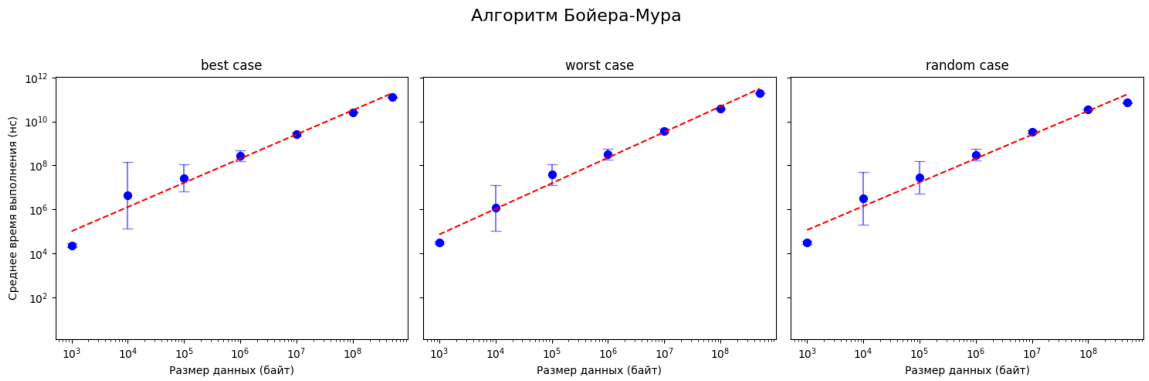


Рис. 3: Время работы алгоритма Бойера-Мура

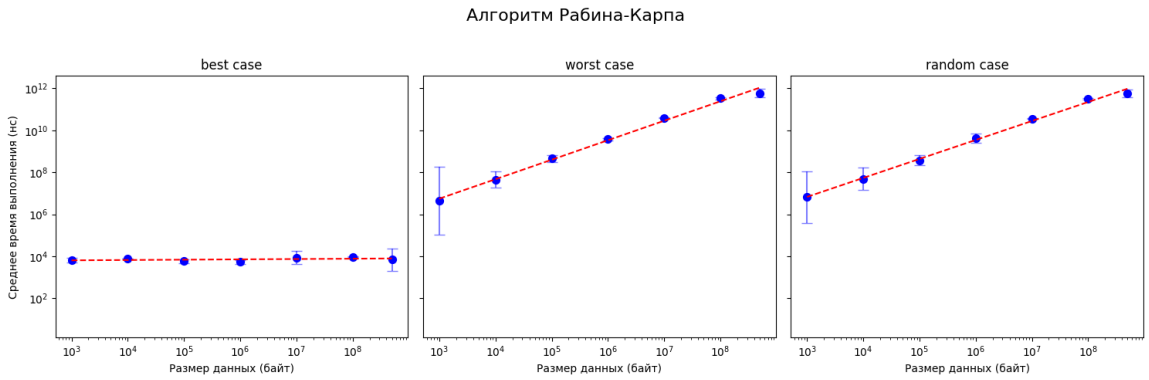


Рис. 4: Время работы алгоритма Рабина-Карпа

# 5 Обоснование результатов

Результаты соответствуют теоретическим сложностям алгоритмов:

**Наивный алгоритм:** Показывает отличные результаты в лучшем случае, но плохо масштабируется в худших и случайных случаях, особенно для больших данных.

**КМР:** Демонстрирует стабильную производительность в лучшем случае и во многом обгоняет наивный алгоритм на небольших размерах строк, но накладные расходы на предобработку заметны для очень больших наборов, из-за чего алгоритм сопоставим по работе с наивным.

**Бойера-Мура:** Характеризуется высокой вариабельностью. В лучшем случае производительность страдает от накладных расходов на эвристики для малых данных. На графике видно, что среднее время не зависит от характеристик строки.

**Рабина-Карпа:** Сравним с КМР в лучшем случае, но испытывает трудности в худших и случайных случаях для больших данных из-за возможных коллизий хешей, приводящих к медленной работе для всех объёмов данных, относительно других алгоритмов.

Самые широкие доверительные интервалы заметны у Бойера-Мура, самые же узкие - у наивного алгоритма, в силу простоты вычислетльной операции сравнения.

# 6 Анализ

**Лучший случай:** Наивный алгоритм неожиданно оказывается самым быстрым для малых и средних данных благодаря простоте и отсутствию накладных расходов на предобработку. КМР и Рабина-Карпа конкурентоспособны, тогда как Бойера-Мура самый медленный из-за сложных эвристик.

**Худший случай:** Бойер-Мур показывает лучшую работу на небольших данных, худшую - Рабин-Карп. На больших данных ассимптотика алгоритмов примерно одинакова

**Случайный случай:** Исходя из графиков и велечин доверительных интервалов, для случайных данных Бойер-Мур показывает лучшую работу для малых и средних размеров данных. Для больших размеров данных время работы всех алгоритмов примерно совпадает, но выделяются наивный алгоритм и Бойер-Мур, показывая наименьшие результаты.

Простота наивного алгоритма делает его быстрее, чем теоретически более эффективные КМР и Бойера-Мура в случайном случае для крупных данных, подчёркивая влияние констант и накладных расходов в реальных условиях.

# 7 Вывод

Алгоритм Бойера-Мура является наиболее устойчивым для всех типов случаев и размеров данных, что делает его предпочтительным для универсального поиска подстроки. Наивный — сильная альтернатива для больших случайных данных, тогда как производительность Рабина-Карпа ограничена коллизиями хешей в худших и случайных случаях. КМР, исходя из наблюдений, является наименее эффективным для поиска.

# 8 Примечание

Исходный код использованных алгоритмов, тесты, а также составление результатов измерения предоставлены в репозитории. Данные с эксперементов в виде CSV-файлов также представлены в репозитории.