Вар 6

Введение

Был проведен сравнительный анализ двух методов поиска подстрок в строке: алгоритма Бойера-Мура-Хорспула (BMH) и стандартного метода поиска C++ (std::string::find).

Методология

Исследовались зависимости:

Время выполнения в зависимости от длины строки T при фиксированной длине шаблона t (N увеличивается от 128 до 8192).

Время выполнения в зависимости от длины шаблона t при фиксированной длине строки T (N=8192).

Влияние размера алфавита на время выполнения при фиксированных N и M.

Замеры производились 10 раз для каждой конфигурации, и выбиралось минимальное время, чтобы минимизировать влияние многозадачности операционной системы.

Результаты

Зависимость от длины строки T:

Алгоритм BMH показывает значительно лучшее время выполнения при больших значениях N, особенно заметно на длинных строках.

std::string::find демонстрирует линейный рост времени выполнения с увеличением N.

Зависимость от длины шаблона t:

Алгоритм BMH менее чувствителен к увеличению длины шаблона по сравнению с std::string::find.

Влияние размера алфавита:

При малом размере алфавита BMH показывает существенные улучшения в скорости, так как плохие символы реже встречаются.

При увеличении размера алфавита разница между методами сокращается, но BMH все еще сохраняет преимущество.

Заключение

Алгоритм Бойера-Мура-Хорспула (BMH) значительно превосходит стандартный метод std::string::find в сценариях с большими строками и малыми алфавитами. BMH более эффективно справляется с задачей поиска, особенно на длинных строках и при небольших размерах алфавита, что делает его предпочтительным выбором для таких задач.