

Вывод по таблицам и графикам сравнения разных алгоритмов поиска вхождения шаблона в текст

Метод	Преимущества	Недостатки
naïve	Поиск подстроки выполняется быстро, когда её размеры достаточно близки к размеру текста	Является самой медленной в большинстве случаев работы, когда подстрока существенно меньше исходного текста
kmp_standart	Работает быстрее наивного алгоритма и z_function практически во всех случаях, kmp_updated на маленьких размерах текста без подстановок	Медленнее kmp_updated на больших размерах текста Медленнее наивного в тех случаях, когда подстрока по длине не сильно отличается от текста Экспоненциальная зависимость времени от количества подстановок
kmp_updated	Работает быстрее наивного алгоритма и z_function практически во всех случаях, а также является самым быстрым алгоритмом для больших размеров текста без подстановок	Медленнее наивного тех случаях, когда подстрока по длине не сильно отличается от текста Медленнее kmp_standart на маленьких размерах текста Экспоненциальная зависимость времени от количества подстановок
z_function	Работает быстрее наивной на больших размерах текста практически во всех случаях	Медленнее kmp_standard и kmp_updated практически во всех случаях Медленнее наивного в тех случаях, когда подстрока по длине не сильно отличается от текста, а также при большом алфавите и использовании подстановок

Графики измерения времени не являются плавными по нескольким причинам:

- устройство, на котором происходит компиляция, может быть загружено другими процессами
- могут возникать случайные выбросы, которые искажают результат тестирования
- Генерация шаблонов и текста сильно влияет на результат работы

Тестирование лучше проводить на практически незагруженном другими процессами устройстве, а также проводить поиск несколько раз, усредняя полученный результат