

# Отчёт

Данные предоставлены в формате [строка] [время без опт.] [время с опт.].

[illegible][illegible]

```
3 test:
" 1 1
aabbab 32 3
aababaaaaaaaaaaaaabb ... 4
abbaabbbbbbbbbbbbabaaababaabababababbaab ... 7
```

```
6 test:
acbx 32 3
accdcbx 426 5
accdcdcdccbx ... 10
accdcdcdcccccdcdcdcccccdcdcdcccccdcdcdcccccdcdcdcccccdcdcdcccccdcdcdccbx
... 37
```

Я уже не стал записывать, но в случае неправильных строк ситуация аналогичная, что довольно логично, потому как чтобы добраться до ошибки нужно всё равно распарсить всю строку.

В общем эти тесты демонстрируют мощьность этих оптимизаций, которые делают алгоритм просто пригодным для пользования на практике, т.к. ранее глубина 10 для строки была уже смертельной для кода.

Из-за тяжести рекурсий и экспоненциального разрастания алгоритма. Эти оптимизации сработают в абсолютном большинстве регулярных выражений, если только не придумывать специальные регулярки, которые не подойдут ни под одну оптимизацию и смогут разрастаться.