

2

Код написан на питончике, по поводу него все должно быть понятно: старался 1 в 1 переписывать код с лекции и добавил оптимизации. Результаты получились следующие:

Без оптимизаций:

```
0(0|1)*0
Time: 0.000322 100 False
Time: 0.035383 0000000 True
Time: 0.002474 10110 False
Time: 0.008017 010100 True

(228|111)*|((13|37*)|101)
Time: 0.000427 100 False
Time: 0.065383 0000000 False
Time: 0.003956 10110 False
Time: 0.017113 010100 False

(((101)*(110)*)|(100)*)
Time: 0.000425 100 True
Time: 0.086210 0000000 False
Time: 0.004322 10110 False
Time: 0.017656 010100 False
```

С оптимизациями:

```
0(0|1)*0
Time: 0.000010 100 False
Time: 0.000085 0000000 True
Time: 0.000006 10110 False
Time: 0.000070 010100 True
Time: 0.000029 101011110111100100101011100010110111100110011 False

(228|111)*|((13|37*)|101)
Time: 0.000043 100 False
Time: 0.000043 0000000 False
Time: 0.000045 10110 False
Time: 0.000043 010100 False
Time: 0.000068 101011110111100100101011100010110111100110011 False

(((101)*(110)*)|(100)*)
Time: 0.000040 100 True
Time: 0.000040 0000000 False
Time: 0.000053 10110 False
Time: 0.000040 010100 False
Time: 0.000078 101011110111100100101011100010110111100110011 False
```

Заметим, что когда мы тестировать с оптимизациями, мы смогли добавить еще строку для проверки, потому что версия без оптимизации с ней совсем не справлялась. Так что стало ощутимо лучше.

Плохой тест, с которым не справляется даже оптимизированная программа, будет в коде (потому что я не хочу по гигантскому выражению строить регулярку, она есть и видно, что корректно работает):

```
secret information
Time: 0.000094 100 True
Time: 0.000382 0000000 True
Time: 0.000108 10110 False
Time: 0.000056 010100 False
Time: 4.457521 111000111000111000111000111000 True
```