Premature optimization is a root of all evil

Проблемы при профилировании

Скорость работы зависит от множества факторов

- Статистическая обработка разультатов (делаем 5 тестов, отбрасываем самый медленный и самый быстрый)
- Контролируйте нагрузку на процессор и его тактовую частоту

Детерминированное и вероятностное профилирование

- Детерминированное - встраивание счетчиков в код
- Вероятностное - периодическая остановка программы и анализ времени останова (периодическая остановка по исчерпанию люого счетчика, а не только времени)

top/htop/atop/iotop/nettop/time & sysinternals

```
% time python some_script.py
real    0m1.028s
user    0m0.001s
sys     0m0.003s
```

timeit

Используется для профилирования быстрых конструкций

```
1       import timeit
2       print timeit.timeit("a + b", "a,b = 1,2") # 0.03747510910003
```

ipython - %timeit expression

timeit

```
1       import timeit
2
3       for pow in range(6, 8):
4           print timeit.timeit("a + b", "a,b = 1,2", number=10 ** po
5
6       # 2.5 E-8
7       # 2.14E-8
8       # 2.1 E-8
```

```
1       zero_time = timeit.timeit("pass", "", number=number) / numbe
2       timeit.timeit(..., number=number) / num - zero_time
```

profile cProfile

```
1    import re
2    import cProfile
3
4    cProfile.run("re.compile("a|b|" * 100 + "c")")
```

```
4913 function calls (4712 primitive calls) in 0.002 seconds
 Ordered by: standard name
ncalls  tottime  percall  cumtime  percall filename:lineno(functio
     1    0.000    0.000    0.002    0.002 <string>:1(<module>)
     1    0.000    0.000    0.002    0.002 re.py:188(compile)
     1    0.000    0.000    0.002    0.002 re.py:226(_compile)
     2    0.000    0.000    0.000    0.000 sre_compile.py:179(_compile_
     2    0.000    0.000    0.000    0.000 sre_compile.py:208(_optimize
   408    0.000    0.000    0.000    0.000 sre_compile.py:25(_identity
     1    0.000    0.000    0.000    0.000 sre_compile.py:33(_compile)
     1    0.000    0.000    0.000    0.000 sre_compile.py:360(_compile_
     2    0.000    0.000    0.000    0.000 sre_compile.py:473(isstring)
     1    0.000    0.000    0.000    0.000 sre_compile.py:479(_code)
     1    0.000    0.000    0.002    0.002 sre_compile.py:494(compile)
```

```
      203    0.000    0.000    0.000    0.000 sre_parse.py:127(__len__)
      404    0.000    0.000    0.000    0.000 sre_parse.py:131(__getitem_
```