Количество делителей числа

Автор презентации - Кругликов Влад

Не оптимальный способ нахождения делителей числа

Делителем целого числа *n* называется целое число *k*, на которое *n* делится нацело. Значит достаточно перебрать все числа от 1 до *n* *и проверить если остаток от деления a на эти числа равен нулю.*

```
n = 12

for i in range(1, n + 1):
    if n % i == 0:
        print(i)
```

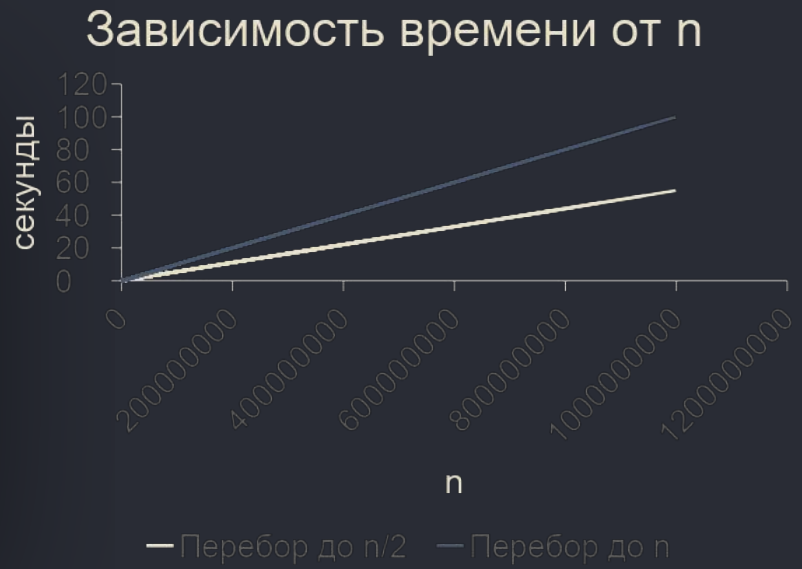
Всё ещё не оптимальный способ нахождения делителей числа

Заметим, что предпоследний делитель n всегда меньше или равен $\frac{n}{2}$, а последним делителем будет являться само число n

```python
n = 12

for i in range(1, n // 2 + 1):
    if n % i == 0:
        print(i)

print(n)
```



Оптимальный способ нахождения делителей числа

1.

Заметим, $\sqrt{n} * \sqrt{n} = n \Rightarrow \begin{cases} x * y = n \\ y < \sqrt{n} \end{cases} \Rightarrow x > \sqrt{n}$, тогда любой делитель $d_0 > \sqrt{n}$ связан с некоторым делителем $d_1 < \sqrt{n}$, простыми словами, после $\sqrt{n}$ новые делители будут являться раннее вычисленными частными.

Например, у числа 12, следующим делителем после делителя 3 будет являться 4, это частное от деления 12 на 3

Далее будет делитель 6, который является частным от деления на 2, далее 16, которое является частным от деления на 1

Оптимальный способ нахождения делителей числа

Таким образом, перебирая все возможные делители до $\sqrt{n}$ мы будем находить 2 делителя, это сам делитель и частное от деления n на делитель

Например рассмотрим число 12, будем перебирать до $\sqrt{12}$, (до 4)

$12 : 1$, делитель 1 и $\frac{12}{1} = 12$

$12 : 2$, делитель 2 и $\frac{12}{2} = 6$

$12 : 3$, делитель 3 и $\frac{12}{3} = 4$

Ответ: 1, 12, 2, 6, 3, 4

Оптимальный способ нахождения делителей числа

Рассмотрим числа, которые являются степенями какого-либо числа, например, 25 это степень пяти, а 4 это степень двойки. У таких чисел делителем всегда будет являться основание степени, то есть число *n* точно имеет делитель $\sqrt{n}$, *потому,что* $\sqrt{n} * \sqrt{n} = n$,значит кратное от деления *n* на $\sqrt{n}$ *равно* $\sqrt{n}$. *На прошлом слайде мы доказали, что делителем числа n, также будет являться частное от деления числа n на его делитель, простыми словами, будет дублироваться делитель, так как если мы поделим 25 на 5, то получим 5. Чтобы отбросить этот лишний делитель достаточно проверить если частное от деления на делитель не равно делителю*

Оптимальный способ нахождения делителей числа

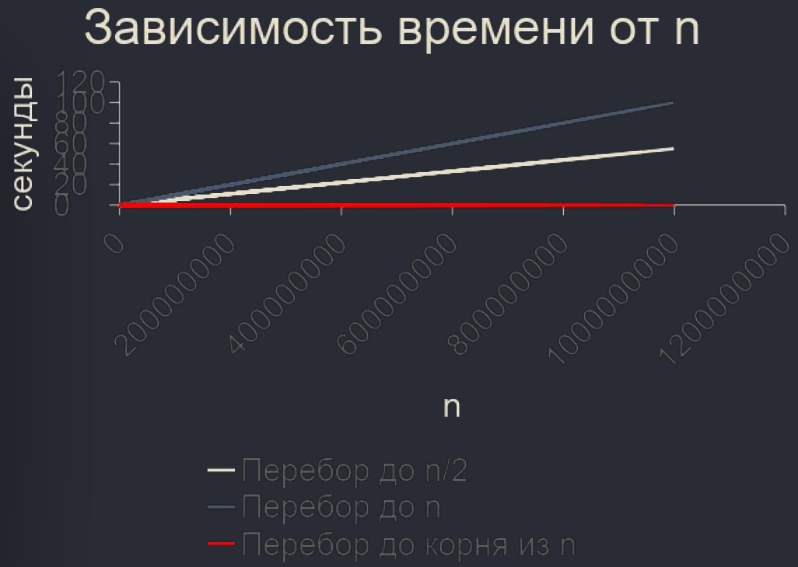
1.

При n = 2^9, перебор до $\sqrt{n}$ занимает 0,005 секунды, в то время, как полный перебор – практически 2 минуты

```python
from math import sqrt

n = 125

for i in range(1, int(sqrt(n)) + 1):
    if n % i == 0:
        if n // i == i:
            print(i)
        else:
            print(i, n // i)
```



Дополнительная оптимизация

```
from math import sqrt

n = 125

last = int(sqrt(n)) + 1

for i in range(1, last):
    if n % i == 0:
        if n // i == i:
            print(i)
        else:
            print(i, n // i)
```

1. При каждом заходе в тело цикла, проверяется условие, что $i \leq int(sqrt(n)) + 1$, то есть каждый раз придется вычислить корень из n, привести его к целочисленному типу и прибавить единицу, и только потом сравнить с i

Поэтому можно заранее вычислить $int(sqrt(n)) + 1$

Тоже самое в 3 строчки

```
from math import sqrt

n = 125

[print(i if n // i == 0 else i, n // i) for i in range(1, int(sqrt(n)) + 1) if n % i == 0]
```