О-нотация

О-нотация нужна для определения сложности алгоритма. Так как одну и ту же задачу можно решить разными способами - нужно уметь отличать эффективное решение от неэффективного.

Давайте разберем на примере: нам вводится число N, нужно определить сумму чисел от 1 до N.

Первый способ - пройтись циклом по всем числам от 1 до N и просуммировать их в переменную.

```
n = int(input())
ans = 0
for i in range(1, n + 1):
    ans += i
print(ans)
```

Второй способ - воспользоваться формулой

```
n = int(input())
ans = (1 + n) * n // 2
print(ans)
```

Итак, оба варианта дадут нам правильный ответ. Однако, при очень больших значениях N, например, N = 100,000,000,000,000,000 первый будет считать этот ответ дольше человеческой жизни, а второй это сделает меньше, чем за 1 секунду.

Так происходит потому, что в первом случае мы делаем гораздо больше базовых операций. Базовой операцией называются такие, которые выполняются **примерно** за одинаковое время вне зависимости от входных данных. Например, сложение / вычитание / умножение и тд. Если в первом случае мы будем повторять операцию сложения ~N раз, то во втором случае количество базовых операций не зависит от входных данных и всегда будет одинаковым - сложение, умножение и деление.

О-нотация (еще по-другому называют умным словом ассимптотика) - это оценка количества базовых операций в зависимости от входных данных. Причем, учитывается только изменяемая часть. Например, если мы сделаем три цикла, каждый по N итераций

```
n = int(input())
cnt = 0
for i in range(n):
    cnt += 1
for i in range(n):
    cnt += 1
for i in range(n):
    cnt += 1
```

то сложность алгоритма будет не O(3*n), а просто O(n). Так как при очень больших значениях n, эта тройка практически никак не влияет. Если простым языком, то все множители / делители / слагаемые, которые напрямую не касаются входных данных отбрасываются.

Вот еще один пример:

```
n = int(input())
m = int(input())
cnt = 0
for i in range(n):
    cnt += 1
for j in range(m):
    cnt -= 1
print(cnt)
```

В данном примере сложность алгоритма будет O(n+m)