

асимптотика, мастер-теорема, th об экспоненциальных рекуррентных соотношениях, алгоритм Карацубы

давайте начнем с простого примера. у нас есть программа, которая выполняет некий код, например

```
for i = 0 ... n - 1
  for j = 0 ... n - 1
    sum += i % j
```

интуитивно понятно, что этот код выполняет примерно n^2 операций. но формально сказать «примерно n^2 » недостаточно точно. нужно определить, что значит «примерно». для этого вводится специальная формалистика — асимптотические обозначения. они позволяют описывать поведение алгоритмов при больших n , не заботясь о константах и мелких деталях реализации

перед тем как ввести асимптотику, важно понять, зачем измерять время работы программы. допустим, процессор работает на частоте 2.3 гигагерца. это означает, что он выполняет примерно $2.3 \cdot 10^9$ операций в секунду. пусть программа работает 1 секунду. если она делает n^2 операций, то мы можем прикинуть, какое n успеет обработаться за это время

$$\begin{aligned}n^2 &\approx 2.3 \cdot 10^9 \\n &\approx \sqrt{2.3 \cdot 10^9} \\n &\approx 40000\end{aligned}$$

значит, если алгоритм работает за n^2 , то он успеет обработать около 40 тысяч элементов за секунду.

если же программа написана на медленном языке, например на python, то можно считать, что выполняется около 10^6 операций в секунду. тогда

$$\begin{aligned}n^2 &\approx 10^6 \\n &\approx 1000\end{aligned}$$

такие оценки позволяют понимать, насколько алгоритм пригоден для практического использования.

чтобы формализовать понятие «работает примерно как n^2 », используют обозначения Big-O и другие асимптотические нотации.