t(n, input) - время работы алгоритмы при входных данных input размера n. Тогда время работы алгоритма $t(n) = max_{input} t(n, input)$.

$t(n) = O(f(n)) \Leftrightarrow \exists c > 0,\ N > 0 : \forall n > N\ t(n) \le c \cdot f(n)$.

$t(n) = o(f(n)) \Leftrightarrow \forall c > 0, \exists N : \forall n > N\ t(n) \le c \cdot f(n)$.

$t(n) = \Omega(f(n)) \Leftrightarrow \exists c > 0,\ \forall N,\ \exists n > N,\ input : t(n, input) \ge c \cdot f(n)$.

$t(n) = \omega(f(n)) \Leftrightarrow \forall c > 0,\ \forall N,\ \exists n > N,\ input : t(n, input) \ge c \cdot f(n)$.

$t(n) = \theta(f(n)) \Leftrightarrow t(n) = \Omega(f(n)),\ t(n) = O(f(n))$.

Алгоритм является <u>полиномиальным</u>, если $t(input) = O(|input|^k)$. $|input|$ - битовая длина.

<u>Сильно полиномиальный алгоритм</u> - $t(n) = O(Poly(n))$ - string-poly

<u>Слабо полиномиальный алгоритм</u> - $O(Poly(n, \log C))$ - weak-poly

<u>Псевдо полиномиальный алгоритм</u> - $O(Poly(n, C))$ - pseudo-poly

- unit test (обычные, запускаем просто тесты)
- integration (разные компоненты программы нормально живут вместе)
- prod (тестирование от самого начала до конца)