

1. Выражение для функции временной сложности

Затраты	Итераций
C1	1
C2	1
C3	$\log(n)$
C4	$\log(n) - 1$
C5	$n * (\log(n) - 1)$
C6	$K1 + K2 = C5 - 1 = n * (\log(n) - 1) - 1$
C7	K1
C8	K2
C9	K2

C1	1	<code>int x = 100;</code>
C2	2	<code>int y = 0;</code>
C3	3	<code>for (size_t r = 1; r <= n; r = 2 * r) {</code>
C4	4	<code> x = x + r;</code>
C5	5	<code> for (size_t c = 2; c < n; c++) {</code>
C6	6	<code> if (x > y / c)</code>
C7	7	<code> y = y + r / c;</code>
C8	8	<code> else</code>
C9	9	<code> y = y - 1;</code>

Выше приведена таблица, которая демонстрирует сколько итераций выполняется каждая строка кода.

Строка C7 выполняется K1 итерацию, $K1 = \sum_{j=2}^{n * \log(n) - n - 1} t_j$ где t_j – временные затраты на j-ой итерации.

Строки C8 и C9 выполняются за K2 итераций:

$K2 = \sum_{j=2}^{n * \log(n) - n - 1} m_j$, где m_j – временные затраты на j-ой итерации.

С помощью таблицы можно составить точное выражение для функции временной сложности: $T(n) = C1 + C2 + C3 * (\log(n)) + C4 * ((\log(n)) - 1) + C5 * (n * ((\log(n)) - 1)) + C6 * (n * ((\log(n)) - 1) - 1) + C7 * K1 + C8 * K2 + C9 * K2$

2. Поиск функции $f(n)$, для которой справедливо $T(n) = \Theta(f(n))$

1	$\Theta(1)$
2	$\Theta(1)$
3	$\Theta(n * \log(n))$
4	$\Theta(1)$
5	$\Theta(n)$
6	$\Theta(1)$
7	$\Theta(1)$
8	$\Theta(1)$
9	$\Theta(1)$

1	<code>int x = 100;</code>
2	<code>int y = 0;</code>
3	<code>for (size_t r = 1; r <= n; r = 2 * r) {</code>
4	<code> x = x + r;</code>
5	<code> for (size_t c = 2; c < n; c++) {</code>
6	<code> if (x > y / c)</code>
7	<code> y = y + r / c;</code>
8	<code> else</code>
9	<code> y = y - 1;</code>

$T(n) = \Theta(n * \log(n)) \Rightarrow f(n) = n * \log(n)$