

# «Разделяй и властвуй»: умножение чисел

Александр Куликов

Онлайн-курс «Алгоритмы: теория и практика. Методы»

<http://stepic.org/217>

## Сложение в столбик: $O(n)$

перенос: 1                      1   1   1

	1	1	0	1	0	1	(53 <sub>2</sub> )
	1	0	0	0	1	1	(35 <sub>2</sub> )
	<hr/>						
	1	0	1	1	0	0	(88 <sub>2</sub> )

## Умножение в столбик: $O(n^2)$

$$\begin{array}{r}
 \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{+} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 + \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \hline
 1 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1}
 \end{array}
 \begin{array}{l}
 1 \ 1 \ 0 \ 1 \ (13_2) \\
 \times \ 1 \ 0 \ 1 \ 1 \ (11_2) \\
 \hline
 1 \ 1 \ 0 \ 1 \ (1101 \times 1) \\
 1 \ 1 \ 0 \ 1 \ (1101 \times 1, \text{сдвинутое на } 1) \\
 0 \ 0 \ 0 \ 0 \ (1101 \times 0, \text{сдвинутое на } 2) \\
 + \ 1 \ 1 \ 0 \ 1 \ (1101 \times 1, \text{сдвинутое на } 3) \\
 \hline
 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ (143_2)
 \end{array}$$

## Рекуррентная формула

$$y = \begin{cases} 2 \lfloor \frac{y}{2} \rfloor, & \text{если } y \text{ чётно} \\ 1 + 2 \lfloor \frac{y}{2} \rfloor, & \text{если } y \text{ нечётно} \end{cases}$$

## Рекуррентная формула

$$y = \begin{cases} 2 \lfloor \frac{y}{2} \rfloor, & \text{если } y \text{ чётно} \\ 1 + 2 \lfloor \frac{y}{2} \rfloor, & \text{если } y \text{ нечётно} \end{cases}$$

$$x \cdot y = \begin{cases} 2(x \cdot \lfloor \frac{y}{2} \rfloor), & \text{если } y \text{ чётно} \\ x + 2(x \cdot \lfloor \frac{y}{2} \rfloor), & \text{если } y \text{ нечётно} \end{cases}$$

## Алгоритм

### Функция $\text{MULTIPLY}(x, y)$

{Вход: два  $n$ -битовых целых числа  $x \geq 0$  и  $y \geq 0$ .}

{Выход:  $xy$ .}

если  $y = 0$ :

    вернуть 0

$z \leftarrow \text{MULTIPLY}(x, \lfloor y/2 \rfloor)$

если  $y$  чётно:

    вернуть  $2z$

иначе:

    вернуть  $x + 2z$

## Алгоритм

### Функция $\text{MULTIPLY}(x, y)$

{Вход: два  $n$ -битовых целых числа  $x \geq 0$  и  $y \geq 0$ .}

{Выход:  $xy$ .}

если  $y = 0$ :

    вернуть 0

$z \leftarrow \text{MULTIPLY}(x, \lfloor y/2 \rfloor)$

если  $y$  чётно:

    вернуть  $2z$

иначе:

    вернуть  $x + 2z$

**Время работы:**  $O(n^2)$  (число битов в записи  $y$  уменьшается на единицу с каждым рекурсивным вызовом).

## Ещё одна рекуррентная формула

$$\begin{array}{lcl} x & = & \boxed{x_L} \quad \boxed{x_R} = 2^{n/2}x_L + x_R \\ y & = & \boxed{y_L} \quad \boxed{y_R} = 2^{n/2}y_L + y_R \end{array}$$



## Ещё одна рекуррентная формула

$$\begin{aligned}x &= \boxed{x_L} \boxed{x_R} = 2^{n/2}x_L + x_R \\y &= \boxed{y_L} \boxed{y_R} = 2^{n/2}y_L + y_R\end{aligned}$$

$$\begin{aligned}xy &= (2^{n/2}x_L + x_R)(2^{n/2}y_L + y_R) \\&= 2^n x_L y_L + 2^{n/2}(x_L y_R + x_R y_L) + x_R y_R\end{aligned}$$

## Ещё одна рекуррентная формула

$$x = \begin{array}{|c|} \hline x_L \\ \hline \end{array} \begin{array}{|c|} \hline x_R \\ \hline \end{array} = 2^{n/2}x_L + x_R$$

$$y = \begin{array}{|c|} \hline y_L \\ \hline \end{array} \begin{array}{|c|} \hline y_R \\ \hline \end{array} = 2^{n/2}y_L + y_R$$

$$\begin{aligned} xy &= (2^{n/2}x_L + x_R)(2^{n/2}y_L + y_R) \\ &= 2^n x_L y_L + 2^{n/2}(x_L y_R + x_R y_L) + x_R y_R \end{aligned}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$$

## Улучшенная рекуррентная формула

- $xy = 2^n x_L y_L + 2^{n/2} (x_L y_R + x_R y_L) + x_R y_R$

## Улучшенная рекуррентная формула

- $xy = 2^n x_L y_L + 2^{n/2} (x_L y_R + x_R y_L) + x_R y_R$
- вместо четырёх рекурсивных вызовов для вычисления  $x_L y_L$ ,  $x_L y_R$ ,  $x_R y_L$  и  $x_R y_R$ , сделаем **три** для вычисления

$$x_L y_L, x_R y_R, \text{ и } (x_L + x_R)(y_L + y_R)$$

## Улучшенная рекуррентная формула

- $xy = 2^n x_L y_L + 2^{n/2} (x_L y_R + x_R y_L) + x_R y_R$
- вместо четырёх рекурсивных вызовов для вычисления  $x_L y_L$ ,  $x_L y_R$ ,  $x_R y_L$  и  $x_R y_R$ , сделаем **три** для вычисления

$$x_L y_L, x_R y_R, \text{ и } (x_L + x_R)(y_L + y_R)$$

- тогда

$$(x_L y_R + x_R y_L) = (x_L + x_R)(y_L + y_R) - x_L y_L - x_R y_R$$

## Улучшенная рекуррентная формула

- $xy = 2^n x_{LYL} + 2^{n/2}(x_{LYR} + x_{RYL}) + x_{RYR}$
- вместо четырёх рекурсивных вызовов для вычисления  $x_{LYL}$ ,  $x_{LYR}$ ,  $x_{RYL}$  и  $x_{RYR}$ , сделаем **три** для вычисления

$$x_{LYL}, x_{RYR}, \text{ и } (x_L + x_R)(y_L + y_R)$$

- тогда

$$(x_{LYR} + x_{RYL}) = (x_L + x_R)(y_L + y_R) - x_{LYL} - x_{RYR}$$

- соответствующее рекуррентное соотношение:

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$

## Улучшенная рекуррентная формула

- $xy = 2^n x_{LYL} + 2^{n/2}(x_{LYR} + x_{RYL}) + x_{RYR}$
- вместо четырёх рекурсивных вызовов для вычисления  $x_{LYL}$ ,  $x_{LYR}$ ,  $x_{RYL}$  и  $x_{RYR}$ , сделаем **три** для вычисления

$$x_{LYL}, x_{RYR}, \text{ и } (x_L + x_R)(y_L + y_R)$$

- тогда

$$(x_{LYR} + x_{RYL}) = (x_L + x_R)(y_L + y_R) - x_{LYL} - x_{RYR}$$

- соответствующее рекуррентное соотношение:

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$

- скоро покажем, что  $T(n) = O(n^{1.59})$

# Алгоритм Карацубы

## Функция KARATSUBA( $x, y$ )

{Вход: целые числа  $x, y \geq 0$ , в двоичной записи.}  
{Выход:  $xy$ .}

$n \leftarrow \max(\text{размер } x, \text{размер } y)$

если  $n = 1$ : вернуть  $xy$

$x_L, x_R \leftarrow$  левые  $\lceil n/2 \rceil$ , правые  $\lfloor n/2 \rfloor$  битов  $x$

$y_L, y_R \leftarrow$  левые  $\lceil n/2 \rceil$ , правые  $\lfloor n/2 \rfloor$  битов  $y$

$P_1 \leftarrow \text{KARATSUBA}(x_L, y_L)$

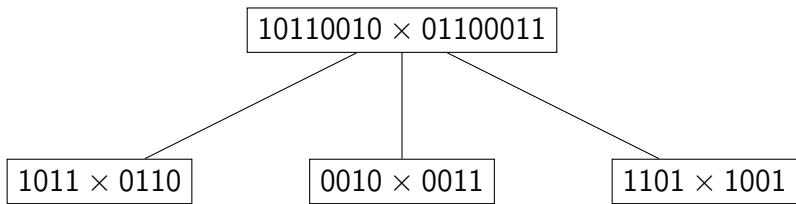
$P_2 \leftarrow \text{KARATSUBA}(x_R, y_R)$

$P_3 \leftarrow \text{KARATSUBA}(x_L + x_R, y_L + y_R)$

вернуть  $P_1 \times 2^{2\lfloor n/2 \rfloor} + (P_3 - P_1 - P_2) \times 2^{\lfloor n/2 \rfloor} + P_2$

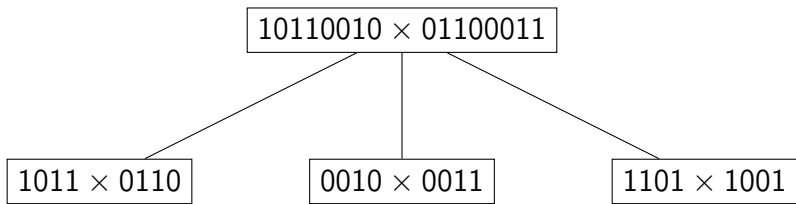


## Дерево рекурсии



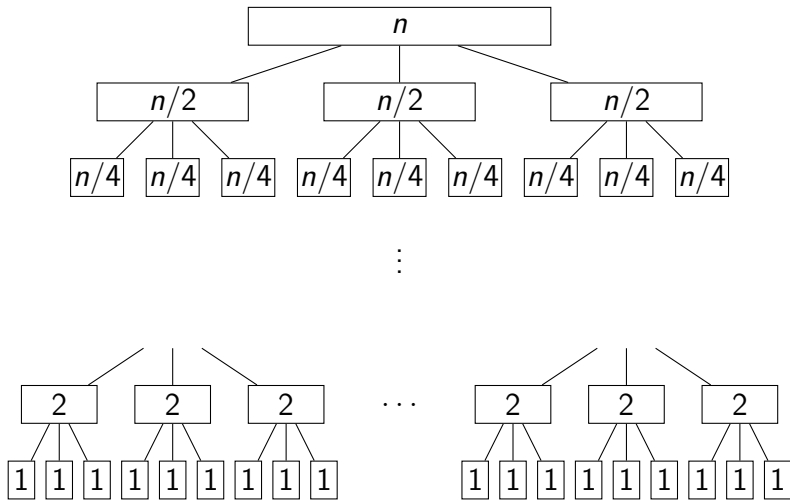
■  $x_L = 1011$ ,  $x_R = 0010$ ,  $y_L = 0110$ ,  $y_R = 0011$

## Дерево рекурсии

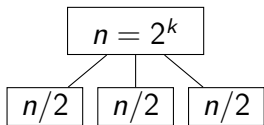


- $x_L = 1011, x_R = 0010, y_L = 0110, y_R = 0011$
- $x_L + x_R = 1101, y_L + y_R = 1001$

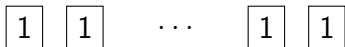
# Дерево рекурсии



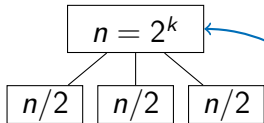
# Оценка времени работы



⋮

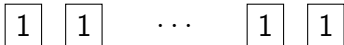


# Оценка времени работы

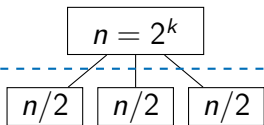


считаем, что  $n$  — степень двойки; позже объясним, почему можно так считать

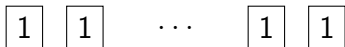
⋮



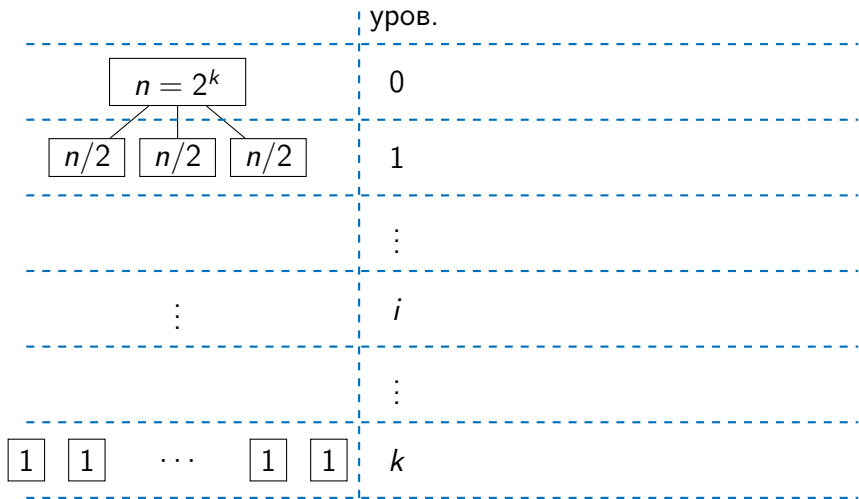
# Оценка времени работы



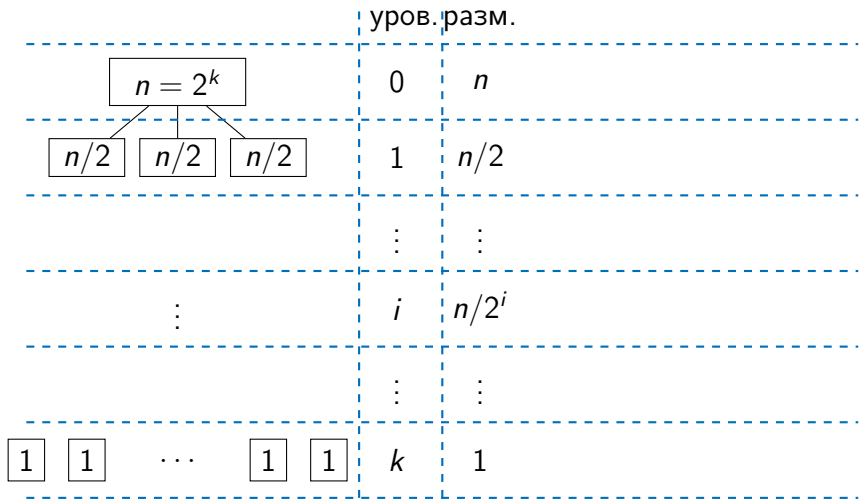
⋮



# Оценка времени работы



# Оценка времени работы

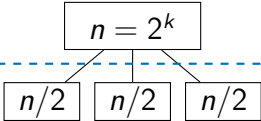
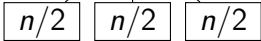
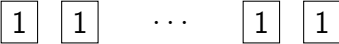




# Оценка времени работы

				уров.	разм.	#
<div><div><div><math>n = 2^k</math></div><div><div><math>n/2</math></div><div><math>n/2</math></div><div><math>n/2</math></div></div></div></div>				0	$n$	1
				1	$n/2$	3
				$\vdots$	$\vdots$	$\vdots$
$\vdots$				$i$	$n/2^i$	$3^i$
				$\vdots$	$\vdots$	$\vdots$
<div><div>1</div><div>1</div><div>...</div><div>1</div><div>1</div></div>	$k$	1	$3^k$			

# Оценка времени работы

	уров.	разм.	#	работа
	0	$n$	1	$cn$
	1	$n/2$	3	$3 \cdot c \cdot n/2$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$i$	$n/2^i$	$3^i$	$3^i \cdot c \cdot n/2^i$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$k$	1	$3^k$	$3^k \cdot c \cdot n/2^k$

# Оценка времени работы

				уров.	разм.	#	работа
<div><div><div><math>n = 2^k</math></div><div><div><math>n/2</math></div><div><math>n/2</math></div><div><math>n/2</math></div></div></div></div>				0	$n$	1	$cn$
				1	$n/2$	3	$3 \cdot c \cdot n/2$
				$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$				$i$	$n/2^i$	$3^i$	$3^i \cdot c \cdot n/2^i$
				$\vdots$	$\vdots$	$\vdots$	$\vdots$
<div><div>1</div><div>1</div><div>...</div><div>1</div><div>1</div></div>	$k$	1	$3^k$	$3^k \cdot c \cdot n/2^k$			

$$\sum_{i=0}^k 3^i \cdot c \cdot n/2^i$$

## Сумма геометрической прогрессии: формула

- геометрическая прогрессия:  $1 + c + c^2 + \dots + c^n$

## Сумма геометрической прогрессии: формула

- геометрическая прогрессия:  $1 + c + c^2 + \dots + c^n$
- если домножить на  $(c - 1)$  и раскрыть скобки, почти всё сократится:

$$\begin{aligned} & (c + c^2 + c^3 + \dots + c^{n+1}) \\ & - (1 + c + c^2 + \dots + c^n) = c^{n+1} - 1 \end{aligned}$$

## Сумма геометрической прогрессии: формула

- геометрическая прогрессия:  $1 + c + c^2 + \dots + c^n$
- если домножить на  $(c - 1)$  и раскрыть скобки, почти всё сократится:

$$\begin{aligned} & (c + c^2 + c^3 + \dots + c^{n+1}) \\ & \quad - (1 + c + c^2 + \dots + c^n) = c^{n+1} - 1 \end{aligned}$$

- поэтому при  $c \neq 1$  верно равенство

$$1 + c + c^2 + \dots + c^n = \frac{c^{n+1} - 1}{c - 1}$$

## Сумма геометрической прогрессии: скорость роста

$$1 + c + c^2 + \dots + c^n = \begin{cases} \Theta(1) & \text{если } c < 1 \\ \Theta(n) & \text{если } c = 1 \\ \Theta(c^n) & \text{если } c > 1 \end{cases}$$

## Сумма геометрической прогрессии: скорость роста

$$1 + c + c^2 + \dots + c^n = \begin{cases} \Theta(1) & \text{если } c < 1 \\ \Theta(n) & \text{если } c = 1 \\ \Theta(c^n) & \text{если } c > 1 \end{cases}$$

■ Если  $c < 1$ ,

$$1 < \frac{c^{n+1} - 1}{c - 1} = \frac{1 - c^{n+1}}{1 - c} < \frac{1}{1 - c}$$



## Сумма геометрической прогрессии: скорость роста

$$1 + c + c^2 + \dots + c^n = \begin{cases} \Theta(1) & \text{если } c < 1 \\ \Theta(n) & \text{если } c = 1 \\ \Theta(c^n) & \text{если } c > 1 \end{cases}$$

- Если  $c < 1$ ,

$$1 < \frac{c^{n+1} - 1}{c - 1} = \frac{1 - c^{n+1}}{1 - c} < \frac{1}{1 - c}$$

- Если  $c > 1$ ,

$$c^n < \frac{c^{n+1} - 1}{c - 1} < \frac{c}{c - 1} c^n$$

## Оценка на время работы

$$\sum_{i=0}^k 3^i \cdot c \cdot n / 2^i = cn \cdot \sum_{i=0}^k \left(\frac{3}{2}\right)^i$$

## Оценка на время работы

$$\begin{aligned}\sum_{i=0}^k 3^i \cdot c \cdot n / 2^i &= cn \cdot \sum_{i=0}^k \left(\frac{3}{2}\right)^i \\ &= cn \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i\end{aligned}$$

## Оценка на время работы

$$\begin{aligned}\sum_{i=0}^k 3^i \cdot c \cdot n / 2^i &= cn \cdot \sum_{i=0}^k \left(\frac{3}{2}\right)^i \\ &= cn \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i \\ &= cn \cdot \Theta\left(\frac{3^{\log_2 n}}{2^{\log_2 n}}\right)\end{aligned}$$

## Оценка на время работы

$$\begin{aligned}\sum_{i=0}^k 3^i \cdot c \cdot n / 2^i &= cn \cdot \sum_{i=0}^k \left(\frac{3}{2}\right)^i \\&= cn \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i \\&= cn \cdot \Theta\left(\frac{3^{\log_2 n}}{2^{\log_2 n}}\right) \\&= \Theta(n^{\log_2 3})\end{aligned}$$

## Оценка на время работы

$$\begin{aligned}\sum_{i=0}^k 3^i \cdot c \cdot n / 2^i &= cn \cdot \sum_{i=0}^k \left(\frac{3}{2}\right)^i \\&= cn \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i \\&= cn \cdot \Theta\left(\frac{3^{\log_2 n}}{2^{\log_2 n}}\right) \\&= \Theta(n^{\log_2 3}) \\&= \Theta(n^{1.584\dots})\end{aligned}$$

Почему можно считать, что  
 $n = 2^k$ ?



В отрезке  $[n, 2n]$  всегда найдётся степень двойки.

## Заключение

- сложение двух  $n$ -битовых чисел:  $O(n)$



## Заключение

- сложение двух  $n$ -битовых чисел:  $O(n)$
- умножение двух  $n$ -битовых чисел в столбик:  $O(n^2)$

## Заключение

- сложение двух  $n$ -битовых чисел:  $O(n)$
- умножение двух  $n$ -битовых чисел в столбик:  $O(n^2)$
- алгоритм Карацубы умножения двух  $n$ -битовых чисел:  $O(n^{1.59})$