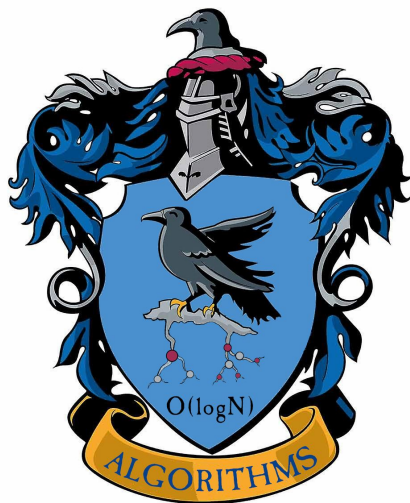


# Алгоритмы и структуры данных

Системное программирование



# Алгоритмы и структуры данных

Организационные вопросы



Преподаватель

Углянский Иван (Евгеньевич)

# Преподаватель

Углянский Иван (Евгеньевич)



@ugliansky



ivan.ugliansky@gmail.com



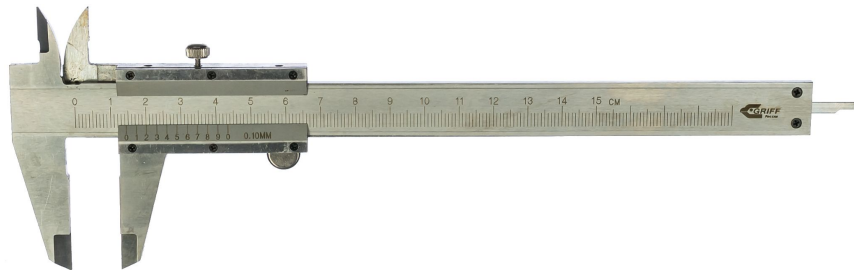
+7-913-763-3346

Плюс ваша группа в телеграме!

Что будем изучать?

# Что будем изучать?

- Инструменты и методики изучения алгоритмов



# Что будем изучать?

- Инструменты и методики изучения алгоритмов
- Классические алгоритмы и структуры данных



# Что будем изучать?

- Инструменты и методики изучения алгоритмов
- Классические алгоритмы и структуры данных
- Применение алгоритмов для смежных задач

**JUST USE IT.**





# Что будем изучать?

Инструменты

Сортировки

Divide & conquer

Рандомизированные алгоритмы

Списки, кучи, деревья поиска,  
сбалансированные деревья

Хэш-функции и хэш-таблицы

Графы

Динамическое программирование



Зачем будем изучать?

# Зачем будем изучать?

- База для всего остального Computer Science (в прикладном смысле)

# Зачем будем изучать?

- База для всего остального Computer Science (в прикладном смысле)
- Огромное количество практических применений
  - ✓ Поисковые движки
  - ✓ Реализация баз данных
  - ✓ Компиляторы и рантаймы

# Зачем будем изучать?

- База для всего остального Computer Science (в прикладном смысле)
- Огромное количество практических применений
  - ✓ Поисковые движки
  - ✓ Реализация баз данных
  - ✓ Компиляторы и рантаймы



JDK / JDK-8067767

type inference performance regression

E.g. the following program takes ~25s with JDK 12 and <1s with JDK 7:

# Зачем будем изучать?

- База для всего остального Computer Science (в прикладном смысле)
- Огромное количество практических применений
- Алгоритмическая секция на собеседованиях



## 2. Add Two Numbers

Medium  4668  1171  Favorite  Share

**Input:** (2 -> 4 -> 3) + (5 -> 6 -> 4)

**Output:** 7 -> 0 -> 8

**Explanation:** 342 + 465 = 807.

# Зачем будем изучать?

- База для всего остального Computer Science (в прикладном смысле)
- Огромное количество практических применений
- Алгоритмическая секций на собеседованиях
- Это весело!



**Because Losing Is Fun**

Как будем изучать?



# Как будем изучать?

- 16 лекций + 16 семинаров

# Как будем изучать?

- 16 лекций + 16 семинаров
- Практика = миньки + "большие" задачи

# Как будем изучать?

- 16 лекций + 16 семинаров
- Практика = миньки + "большие" задачи
  - минек много, некоторые на литкоде
  - реализовывать на **любом** языке  
(в пределах разумного)

# Балловая система

- ✓ Практика —  $[0; 60]$  баллов
- ✓ Сдача теории —  $[0; 40]$  баллов

# Балловая система

✓ Практика –  $[0; 60]$  баллов

✓ Сдача теории –  $[0; 40]$  баллов

Миньки по 1 или 2 балла (для каждой указано)

"Большие" задачи по 5 баллов.

# Балловая система

- ✓ Практика –  $[0; 60]$  баллов
- ✓ Сдача теории –  $[0; 40]$  баллов

На **экзамене**: два вопроса + задача

# Балловая система

✓ Практика –  $[0; 60]$  баллов

✓ Сдача теории –  $[0; 40]$  баллов

---

✓  $[85; 100]$  баллов – отл.

✓  $[70; 85)$  баллов – хор.

✓  $[55; 70)$  баллов – удовл.

# Балловая система

- ✓ Практика –  $[0; 60]$  баллов
- ✓ Сдача теории –  $[0; 40]$  баллов

---

✓  $[85; 100]$  баллов – отл.

✓  $[70; 85)$  баллов – хор.

✓  $[55; 70)$  баллов – удовл.

50 за практику =  
допуск на досрок



# Балловая система

✓ Практика –  $[0; 60]$  баллов

✓ Сдача теории –  $[0; 40]$  баллов

---

✓  $[85; 100]$  баллов – отл.

✓  $[70; 85)$  баллов – хор.

✓  $[55; 70)$  баллов – удовл.

50 за практику =  
допуск на досрок

60 за практику =  
досрок и экзамен  
без задачи

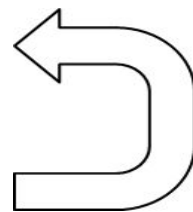
# Сдача задачи

1. Получить свой вариант задачи
2. Написать решение



# Сдача задачи

1. Получить свой вариант задачи
2. Написать решение
3. Пройти автоматические тесты
4. Пройти code review
5. Защитить решение на паре



**DEADLINE**  
**2 WEEKS**

# Дедлайны

МЫ РАЗОЗЛИЛИ ВРЕМЯ,  
И ТЕПЕРЬ У НАС ВСЕГДА  
ПОЛЧАСА ДО ДЭДЛАЙНА



# Дедлайны

1. Просрочка на неделю  $\Rightarrow$   
задача дешевле до  
2.5 баллов
2. Просрочка еще больше  $\Rightarrow$   
задача не засчитывается



# Дедлайны

1. Дедлайн доп. задач - одна неделя

2. Сдача по упрощенной схеме



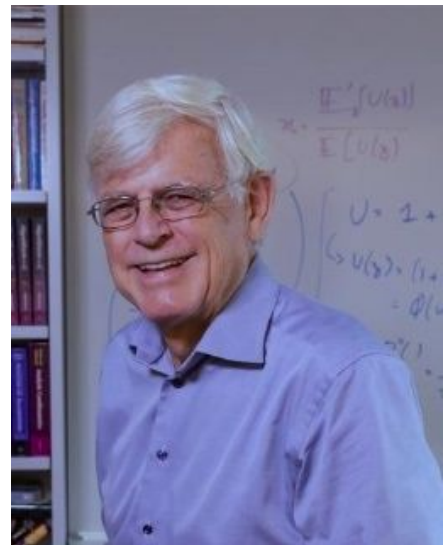
# Материалы и референсы

1. Стэнфордский курс  
`Algorithms: Design and Analysis`



# Материалы и референсы

1. Стэнфордский курс  
[Algorithms: Design and Analysis](#)
2. Принстонский курс  
[Algorithms \(Part 1, Part 2\)](#)





# Материалы и референсы

1. Стэнфордский курс  
[Algorithms: Design and Analysis](#)
2. Принстонский курс  
[Algorithms \(Part 1, Part 2\)](#)
3. Алгоритмы. Построение и анализ  
Томас Кормен, ...



Ура, теперь мы готовы  
приступить к курсу!

# Простой пример

Как перемножить два числа?

# Простой пример

Как перемножить два числа?

**Input:**  $k$  и  $p$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output:**  $k * p$

# Простой пример

Как перемножить два числа?

**Input:**  $k$  и  $p$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output:**  $k * p$

Алгоритм?

# Простой пример

Как перемножить два числа?

**Input:**  $k$  и  $p$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output:**  $k * p$

Алгоритм? В столбик!

# Простой пример

$$1234 * 5678 = ?$$

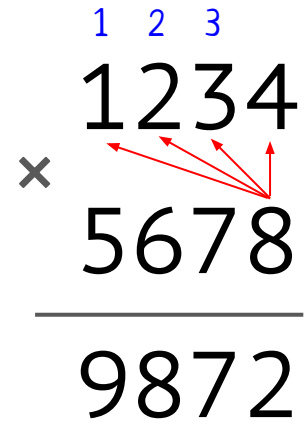
$$\begin{array}{r} 1234 \\ \times 5678 \\ \hline \end{array}$$



$$\begin{array}{r} \phantom{\times} 1234 \\ \times \phantom{0000} 5678 \\ \hline \phantom{00000000} 2 \end{array}$$

Diagram illustrating the multiplication of 1234 by 5678. The multiplier 5678 is written below the multiplicand 1234. A horizontal line separates the two numbers. The product 72 is shown below the line. Red arrows point from the 8 in 5678 to the 4 in 1234, and from the 7 in 5678 to the 3 in 1234, illustrating the multiplication process.

$$\begin{array}{r} \phantom{1}^1 \phantom{2}^2 \phantom{3}^3 \\ 1234 \\ \times 5678 \\ \hline 872 \end{array}$$

$$\begin{array}{r} \phantom{1} \phantom{2} \phantom{3} \\ 1234 \\ \times 5678 \\ \hline 9872 \end{array}$$


# Простой пример

Как перемножить два числа?

**Input**:  $x$  и  $y$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output**:  $x * y$

Алгоритм? В столбик!

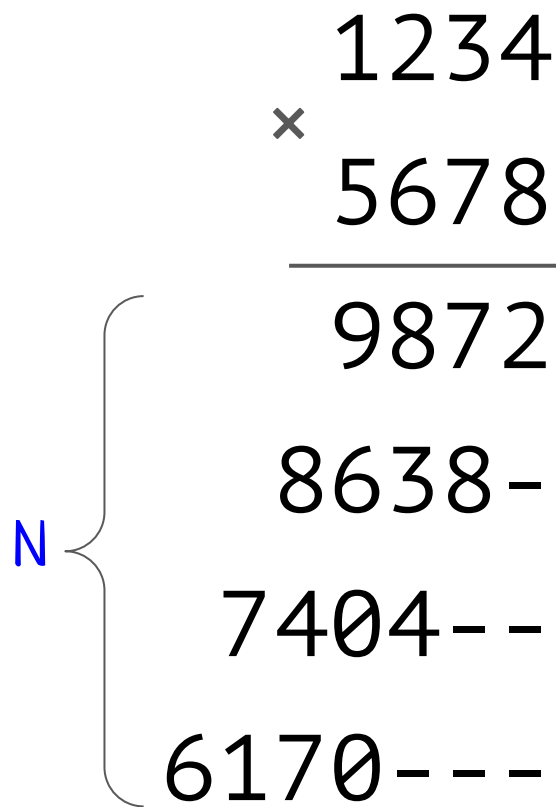
Потратили  $\sim 2 * N$   
элементарных операций

The diagram illustrates a naive matrix multiplication of two 4x4 matrices. The first matrix is 1234 and the second is 5678. Red arrows show the calculation of each element in the result matrix (9872) by summing the products of the corresponding row and column. For example, the first element '9' is the sum of 1\*5, 2\*6, 3\*7, and 4\*8. This process is repeated for all 16 elements in the result matrix, leading to a total of 16 \* 4 = 64 scalar multiplications and 16 \* 3 = 48 scalar additions, totaling 112 elementary operations. This is represented as  $\sim 2 * N^3$  where  $N=4$ .

$$\begin{array}{r} \phantom{\times} \phantom{1} \phantom{2} \phantom{3} \phantom{4} \\ \phantom{\times} \phantom{1} \phantom{2} \phantom{3} \phantom{4} \\ \times \phantom{1} \phantom{2} \phantom{3} \phantom{4} \\ \phantom{\times} \phantom{1} \phantom{2} \phantom{3} \phantom{4} \\ \hline \phantom{\times} \phantom{1} \phantom{2} \phantom{3} \phantom{4} \end{array}$$

Потратили  $\sim 2 * N$   
элементарных операций

Повторили это  $N$  раз


$$\begin{array}{r} 1234 \\ \times 5678 \\ \hline 9872 \\ 8638- \\ 7404-- \\ 6170--- \end{array}$$

Потратили  $\sim 2 * N$  элементарных операций

Повторили это  $N$  раз

И еще потратили на сложение...

$$\begin{array}{r} 1234 \\ \times 5678 \\ \hline 9872 \\ 8638- \\ 7404-- \\ 6170--- \\ \hline 7091312 \end{array}$$



Потратили  $\sim 2 * N$  элементарных операций

Повторили это  $N$  раз

И еще потратили на сложение...  
еще  $\sim 2 * N^2$  операций

The diagram illustrates the multiplication of 1234 by 5678. The numbers are aligned vertically, with 1234 on top and 5678 below it, separated by a horizontal line. The result of the multiplication is shown below another horizontal line. The intermediate steps are shown as follows:

$$\begin{array}{r} 1234 \\ \times 5678 \\ \hline 9872 \\ 8638- \\ 7404-- \\ 6170--- \\ \hline 7006652 \end{array}$$

A large curly bracket on the left side of the intermediate steps (9872, 8638-, 7404--, 6170---) is labeled with a blue  $N$ , indicating that these steps are repeated  $N$  times.

# Простой пример

Как перемножить два числа?

**Input:**  $x$  и  $y$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output:**  $x * y$

**Алгоритм:** в столбик, потратим  $\leq \text{constant} * N^2$

# Простой пример

Как перемножить два числа?

**Input:**  $x$  и  $y$  – целые положительные числа,  
в которых по  $N$  цифр.

**Output:**  $x * y$

**Алгоритм:** в столбик, потратим  $\approx 4 * N^2$

# Простой пример

Как перемножить два числа?

Алгоритм: в столбик, потратим  $\approx 4$   $\leq \text{constant} * N^2$

А можно ли это сделать лучше?



# Алгоритм Карацубы

# Алгоритм Карацубы

1234

5678

# Алгоритм Карацубы

a                      b  
12   34  
c                      d  
56   78

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$



# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

Шаг 4:  $(3) - (2) - (1) = 2840$

$$\begin{array}{r} a \quad \quad b \\ 12 \quad 34 \\ c \quad 56 \quad 78 \quad d \end{array}$$

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

Шаг 4:  $(3) - (2) - (1) = 2840$

Шаг 5:

$$\begin{array}{r} 6720000 \\ + \quad 2652 \\ \hline 284000 \end{array}$$

$$\begin{array}{r} a \quad \quad b \\ 12 \quad 34 \\ c \quad 56 \quad 78 \quad d \end{array}$$

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

Шаг 4:  $(3) - (2) - (1) = 2840$

Шаг 5:

$$\begin{array}{r} 6720000 \\ + \quad 2652 \\ \hline 284000 \end{array} \Rightarrow 7006652!$$

$$\begin{array}{r} a \quad \quad b \\ 12 \quad 34 \\ c \quad 56 \quad 78 \quad d \end{array}$$

Потратили  $\sim 2 * N$  элементарных операций

Повторили это  $N$  раз

И еще потратили на сложение...  
еще  $\sim 2 * N^2$  операций

The diagram illustrates the multiplication of two 4-digit numbers, 1234 and 5678. The result is shown as a sum of four 4-digit products, each shifted one position to the left. A large curly bracket on the left, labeled with a blue  $N$ , indicates that this entire process is repeated  $N$  times. The final result is shown in blue.

$$\begin{array}{r} 1234 \\ \times 5678 \\ \hline 9872 \\ 8638- \\ 7404-- \\ 6170--- \\ \hline 7006652 \end{array}$$

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

Шаг 4:  $(3) - (2) - (1) = 2840$

Шаг 5:

$$\begin{array}{r} 6720000 \\ + \quad 2652 \\ \hline 284000 \end{array} \Rightarrow 7006652!$$

$$\begin{array}{cc} a & b \\ 12 & 34 \\ c & 56 \quad 78 \quad d \end{array}$$



# Алгоритм Карацубы

**Input**:  $x$  и  $y$  – целые положительные числа, в которых по  $N$  цифр.

$a$  12 34  $b$   
 $c$  56 78  $d$



# Алгоритм Карацубы

**Input**:  $x$  и  $y$  – целые положительные числа, в которых по  $N$  цифр.

$a$  12  $b$  34  
 $c$  56  $d$  78

$$x = 10^{\frac{N}{2}} a + b$$

$$y = 10^{\frac{N}{2}} c + d$$

# Алгоритм Карацубы

**Input**:  $x$  и  $y$  - целые положительные числа, в которых по  $N$  цифр.

$a$   $b$   
12 34  
 $c$  56 78  $d$

$$x = 10^{\frac{N}{2}} a + b$$

$$y = 10^{\frac{N}{2}} c + d$$

$$\begin{aligned} \Rightarrow x * y &= (10^{\frac{N}{2}} a + b) * (10^{\frac{N}{2}} c + d) \\ &= 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd \end{aligned}$$

# Алгоритм Карацубы

**Input**:  $x$  и  $y$  - целые положительные числа, в которых по  $N$  цифр.

$a$   $b$   
12 34  
 $c$  56 78  $d$

$$\begin{aligned} x &= 10^{\frac{N}{2}} a + b \\ y &= 10^{\frac{N}{2}} c + d \end{aligned} \quad \Rightarrow \quad \begin{aligned} x * y &= (10^{\frac{N}{2}} a + b) * (10^{\frac{N}{2}} c + d) \\ &= 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd \end{aligned}$$

$ac$ ,  $ad$ ,  $bc$ ,  $bd$  - задачи аналогичные  
изначальной, но **меньшей размерности**

# Алгоритм Карацубы

**Input**:  $x$  и  $y$  - целые положительные числа, в которых по  $N$  цифр.

$a$   $b$   
12 34  
 $c$  56 78  $d$

$$x = 10^{\frac{N}{2}} a + b$$

$$y = 10^{\frac{N}{2}} c + d$$

$$\begin{aligned} \Rightarrow x * y &= (10^{\frac{N}{2}} a + b) * (10^{\frac{N}{2}} c + d) \\ &= 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd \end{aligned}$$

$ac$ ,  $ad$ ,  $bc$ ,  $bd$  - задачи аналогичные  
изначальной, но **меньшей размерности**

Значит можно применить **рекурсию**!

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

a                      b  
12    34  
c    56    78    d

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

Шаг 1: рекурсивно вычисляем  $ac$

a                      b  
12    34  
c    56    78    d

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

Шаг 3: рекурсивно вычисляем  $(a + b)(c + d)$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$



# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

Шаг 3: рекурсивно вычисляем  $(a + b)(c + d) = ac + ad + bc + bd$

$\begin{matrix} a & & b \\ 12 & 34 \\ c & 56 & 78 & d \end{matrix}$

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

a                      b  
12    34  
c    56    78    d

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

Шаг 3: рекурсивно вычисляем  $(a + b)(c + d) = \cancel{ac} + ad + bc + \cancel{bd}$

Шаг 4:  $ad + bc = (3) - (1) - (2)$

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

a 12 b 34  
c 56 d 78

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

Шаг 3: рекурсивно вычисляем  $(a + b)(c + d) = \cancel{ac} + ad + bc + \cancel{bd}$

Шаг 4:  $ad + bc = (3) - (1) - (2)$

Т.е. вместо **4-ех** рекурсивных вызовов, сделали **3!** 75

# Алгоритм Карацубы

Идея:  $x * y = 10^N ac + 10^{\frac{N}{2}} (ad + bc) + bd$

a                      b  
12    34  
c    56    78    d

Шаг 1: рекурсивно вычисляем  $ac$

Шаг 2: рекурсивно вычисляем  $bd$

Шаг 3: рекурсивно вычисляем  $(a + b)(c + d) = \cancel{ac} + ad + bc + \cancel{bd}$

Шаг 4:  $ad + bc = (3) - (1) - (2)$

Шаг 5: Суммируем со сдвигами, получаем ответ!

# Алгоритм Карацубы

Шаг 1:  $a * c = 672$

Шаг 2:  $b * d = 2652$

Шаг 3:  $(a + b) * (c + d) = 46 * 134 = 6164$

Шаг 4:  $(3) - (2) - (1) = 2840$

Шаг 5:

$$\begin{array}{r} 6720000 \\ + \quad 2652 \\ \hline 284000 \end{array} \Rightarrow 7006652!$$

$a$   $b$   
12 34  
 $c$  56 78  $d$

# Алгоритм Карацубы

Рекурсия **конечная**, т.к. каждый раз уменьшаем размер входных данных в два раза.

По индукции легко получаем, что алгоритм Карацубы **корректен**.

a                      b  
12    34  
c    56    78    d

# Алгоритм Карацубы

Рекурсия **конечная**, т.к. каждый раз уменьшаем размер входных данных в два раза.

a			b
	12	34	
c	56	78	d

По индукции легко получаем, что алгоритм Карацубы **корректен**.

Но что с количеством **элементарных операций**? 🤔

# Алгоритм Карацубы

Рекурсия **конечная**, т.к. каждый раз уменьшаем размер входных данных в два раза.

a			b
	12		34
		56	78
c			d

По индукции легко получаем, что алгоритм Карацубы **корректен**.

Но что с количеством **элементарных операций**? 🤔

И какой алгоритм **лучше**: в столбик или Карацубы? 80



# Алгоритм Карацубы

Но что с количеством **элементарных операций**? 🤔

И какой алгоритм **лучше**: в столбик или Карацубы?

# Алгоритм Карацубы

Но что с количеством **элементарных операций**? 🤔

И какой алгоритм **лучше**: в столбик или Карацубы?

Скоро узнаете!



# Алгоритм Карацубы

Замечания:

1. Числа могут быть **разной** длины, алгоритм продолжает работать (подумайте, как это учесть в коде?)

# Алгоритм Карацубы

Замечания:

1. Числа могут быть **разной** длины, алгоритм продолжает работать (подумайте, как это учесть в коде?)
2. Если бы рекурсивных вызовов было **4**, а не **3**, то алгоритм стал бы значительно **хуже** умножения в столбик!

## Мини-задача #1 (1 балл)

Реализуйте алгоритм **деления** в столбик  $n$ -значного числа на  $m$ -значное. Докажите, что он требует по порядку величины  $m \cdot n$  элементарных операций. Рассмотрите худший и лучший случаи алгоритма, оцените количество операций для них.

## Мини-задача #1 (1 балл)

Реализуйте алгоритм **деления** в столбик  $n$ -значного числа на  $m$ -значное. Докажите, что он требует по порядку величины  $m \cdot n$  элементарных операций. Рассмотрите худший и лучший случаи алгоритма, оцените количество операций для них.

## Мини-задача #2 (2 балла)

Реализуйте алгоритм Карацубы умножения целых чисел. Решение должно:

1. Работать с числами разной длины
2. Включать в себя набор тестов для проверки корректности реализации.