профессор Соловьев Сергей Юрьевич

Введение в специальность

или

Постановки задач современной информатики

2017

Постановка задачи (Задача)

1. Исходные данные
2. Результирующие данные
3. Свойства данных

? Алгоритм:

Исходные данные $\Rightarrow$ Результирующие данные

vs.

? Теорема:

Посылка $\Rightarrow$ Следствие

Постановка задачи (Задача)

Дано	Известно
Исходные данные	Свойства исходных данных

Алгоритм

Требуется	такое, что
Результирующие данные	Свойства результирующих данных

Понятие алгоритма

Алгоритм –

точная система правил, которая

- определяет последовательность действий над исходными данными и
- приводит
  - исполнителя алгоритма
  - после конечного числа действий
  - к получению выходных данных.

Пример. Задача вычисления наибольшего общего делителя (НОД)

$$\text{НОД}(n,m) = \max\left\{ d \in N \mid \frac{n}{d} \in N, \frac{m}{d} \in N \right\},$$

где $N = \{ 1, 2, 3, \ldots \}$

$$\text{НОД}(48,64) = 16$$

Пример. Задача вычисления наибольшего общего делителя (НОД)

Дано	Известно
$n, \ m$	$n \in N, \ \ m \in N,$ *где* $N = \{ 1, 2, 3, \ldots \}$

Алгоритм

Требуется	такое, что
k	$k = \text{НОД}(n,m) \equiv \max\{ d \in N \mid \frac{n}{d} \in N, \frac{m}{d} \in N \}$

Алгоритм Евклида вычисления НОД

Св-во E. Если $n = m$, то НОД$(n,m) = n$.

Св-во L. Если $n > m$, то НОД$(n,m) =$ НОД$(n-m,m)$.

Св-во R. Если $n < m$, то НОД$(n,m) =$ НОД$(n,m-n)$.

СТАРТ НОД(48,64) = [[R]]

= НОД(48,16) = [[L]]

= НОД(32,16) = [[L]]

= НОД(16,16) = [[E]] = 16 **СТОП**

Общий подход к проектированию алгоритмов

Задача → {Свойства}$_1$ → Алгоритм-1

→ {Свойства}$_2$ → Алгоритм-2

→ {Свойства}$_3$ → Алгоритм-3

...

Алгоритм Евклида вычисления НОД

Старт. n, m -- натуральные

Шаг 1. Если n = m, то k = n, Стоп.

Шаг 2. Если $n > m$, то n = n-m.

Шаг 3. Если $n < m$, то m = m-n.

Шаг 4. Перейти к шагу 1.

---

0. n = 48, m = 64.
1. (Шаг 1)
2. (Шаг 2)
3. (Шаг 3) n = 48, m = 16.
4. (Шаг 4)
5. (Шаг 1)
6. (Шаг 2) n = 32, m = 16.
7. (Шаг 3)
8. (Шаг 4)
9. (Шаг 1)
10. (Шаг 2) n = 16, m = 16.
11. (Шаг 3)
12. (Шаг 4)
13. (Шаг 1) k = 16 СТОП

Исследование алгоритма Евклида

Конечность?

Утверждение. Для любых натуральных n и m,

Алгоритм Евклида заканчивает вычисления за конечное число действий.

И д е я   д о к а з а т е л ь с т в а

Шаг 1.

Действия: 1 5 9 13

n + m 112 > 64 > 48 > 32 – монотонно убывающая последовательность натуральных чисел.

Исследование алгоритма Евклида

Соответствие задаче?

Утверждение. Для любых натуральных n и m

k = НОД(n,m).

И д е я д о к а з а т е л ь с т в а

Св-во L. Если $n > m$, то ОД(n,m) = ОД(n-m,m).

Св-во R. Если $n < m$, то ОД(n,m) = ОД(n,m-n).

\+ НОД(n,m) $\leq$ n, НОД(n,m) $\leq$ m.

Исследование алгоритма Евклида

Сложность алгоритма? ➔

Утверждение. Сложность алгоритма Евклида есть O(N), где N = max(n,m).

Идея доказательства

Худший случай n > 1, m = 1.

Шаг 1.

Действия:	1	5	9	13	…
N = n	n >	n-1 >	n-2 >	n-3	

Количество действий 4*N – 3 ~ **O(N)**

Сложность алгоритмов

A – алгоритм;

n -- “объем” данных;

T(n) – время вычисления A (max, среднее);

$f : N \to N$

Если $\left| \frac{T(n)}{f(n)} \right| < C$ при $n \to \infty$, то

алгорим A имеет сложность $O(f(n))$.

$f(n) = \log(n) \quad n*\log(n) \quad n^2 \quad n^k \quad 2^n$

Задачи & алгоритмы

Задача-1 0 алгоритмов;

Алгоритмически неразрешимая

Задача-2 1 алгоритм А2

Задача-3 А3-1 > А3-2 > А3-3 > …..

Способы обоснования алгоритмов

*(**от** задачи **к** алгоритму)*

Формальные -- Доказательство

св-ва → алгоритм → док-во

Нормативные -- Документ ➔

Эвристические -- Практика

Коллекции

Прочие --

Нормативный способ

Международный идентификационный код ценной бумаги ISIN.

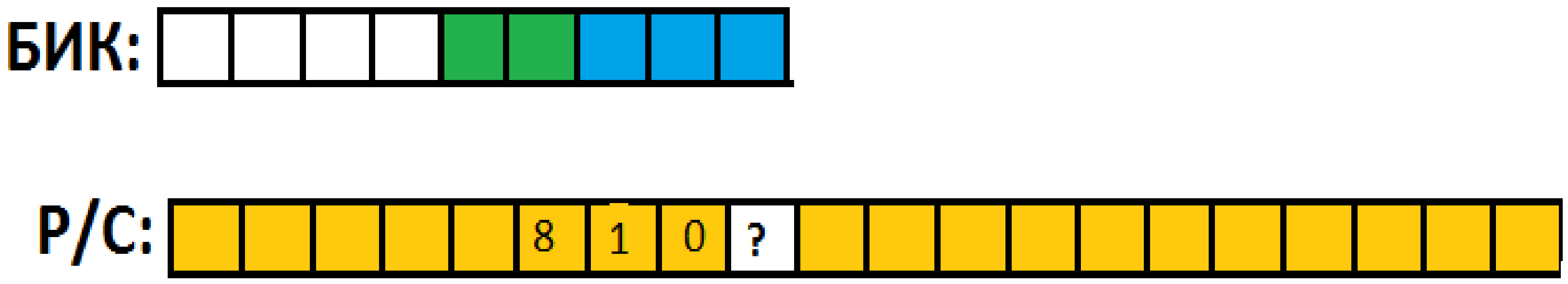
Структура ISIN											
1	2	3	4	5	6	7	8	9	10	11	12
Код страны		NSIN									Контр. цифра

Стандарт ISO 6166.

Modulus 10 Double Add Double technique

Нормативный способ

Порядок расчета контрольного ключа в номере лицевого счета. Приказ Банка России No.02-263 от 18.06.97.



Особенности задач

- Математические свойства: *существование, единственность и пр.*
- Недостаточно формализованные задачи
- Обобщенные постановки задач
- Задачи, допускающие проверку решения

Прямые и обратные задачи

$$y = f(x)$$

Прямая задача:

Известны f и x, вычислить y.

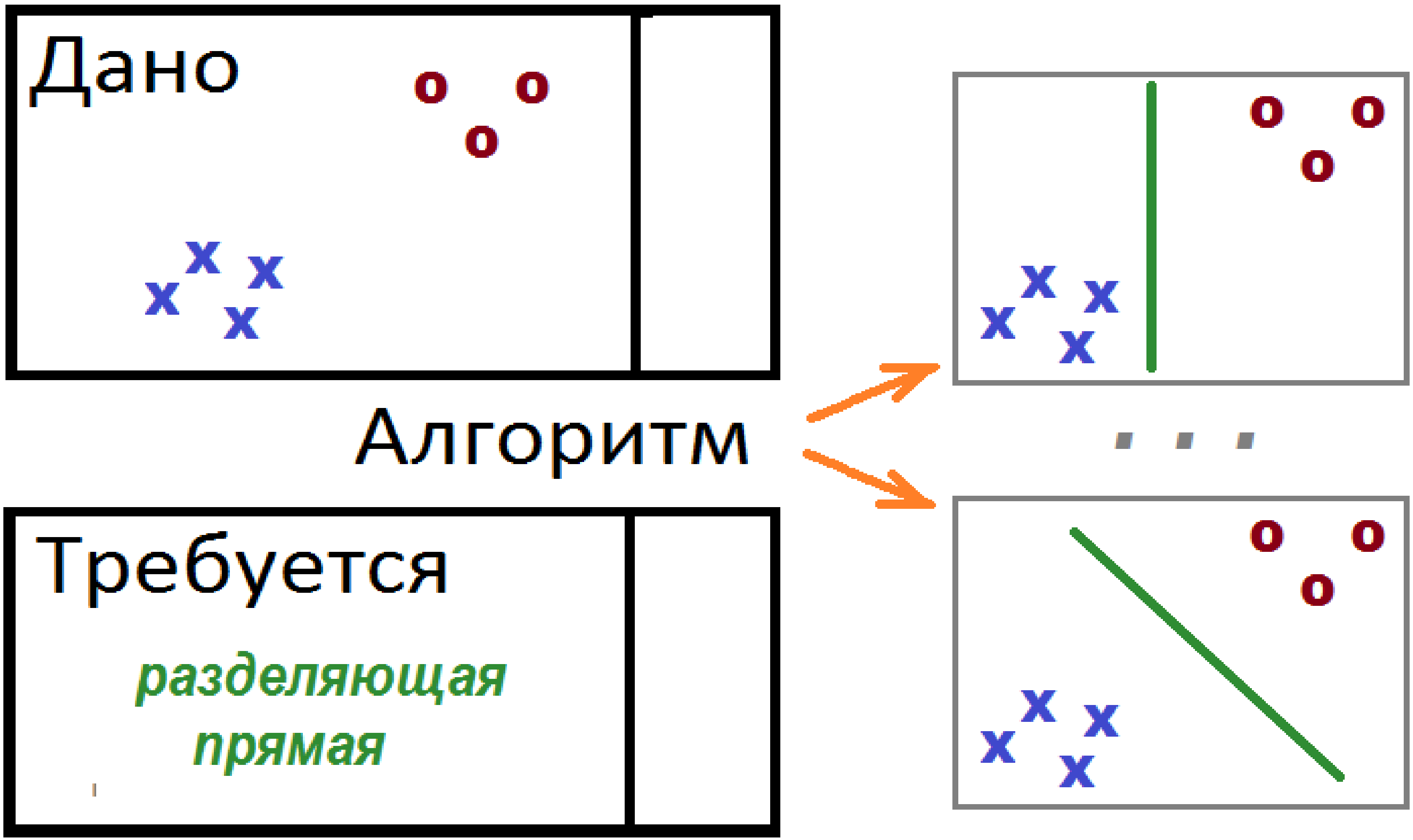
Обратная задача:

Известны f и y, вычислить x.

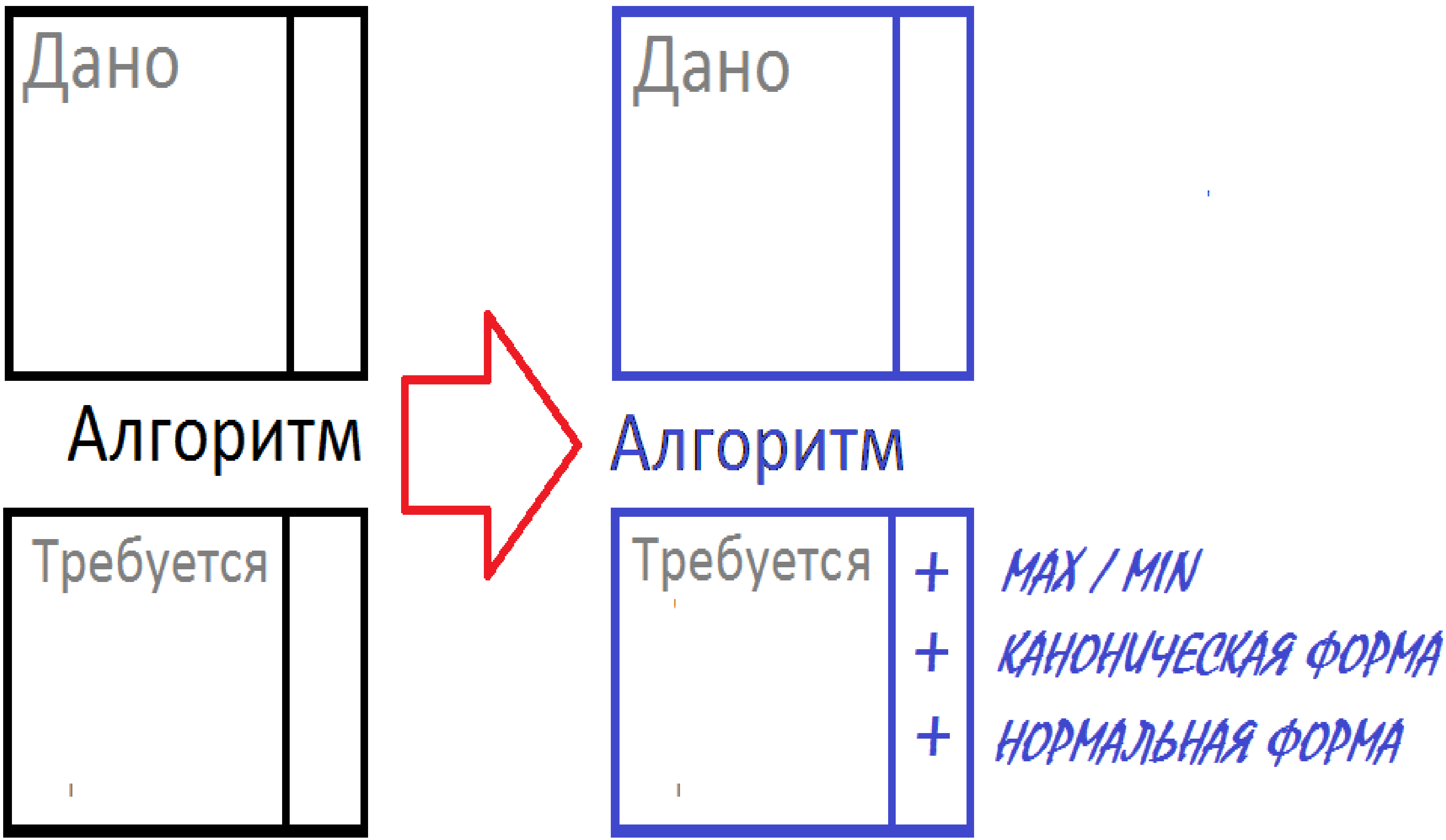
Существование и единственность решения

Недостаточно формализованные задачи

∞ *количество результатов,* ∞ *количество алгоритмов*



Недостаточно формализованные задачи - 2



Обобщенные постановки задач

Задача-1:

<Исх.дан.-1, Свойства-1, Рез.дан-1>

Задача-2: (обобщение)

<Исх.дан.-2, Свойства-2, Рез.дан-2>

$\{$ Исх.дан-1 $\} \subseteq \{$ Исх.дан-2 $\}$

$\{$ Рез.дан-1 $\} \subseteq \{$ Рез.дан-2 $\}$

Свойства-2 $\Rightarrow$ Свойства-1

Задачи, допускающие проверку решения

- Вычисление [целочисленных] корней.
- Построение разделяющей плоскости.
- Обращение матриц.
- др.

В о п р о с ы?

soloviev@glossary.ru