

профессор Соловьев Сергей Юрьевич

**Постановки задач
современной информатики**

2015 – 2018

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 Исходные данные | Известно Свойства исходных данных |
|------------------------------------|---|

Алгоритм

| | |
|--|---|
| Требуется Результирующие данные | такое, что 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)

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

$$\text{где } \mathbb{N} = \{ 1, 2, 3, \dots \}$$

$$\text{НОД}(48,64) = 16$$

Пример. Задача вычисления наибольшего общего делителя (НОД)

| | |
|--------------------------------------|---|
| <p>Дано</p> <p>n, m</p> | <p>Известно</p> <p>$n \in \mathbb{N}, m \in \mathbb{N},$ <i>где</i> $\mathbb{N} = \{1, 2, 3, \dots\}$</p> |
|--------------------------------------|---|

Алгоритм

| | |
|--|--|
| <p>Требуется</p> <p>k</p> | <p>такое, что</p> <p>$k = \text{НОД}(n, m) \equiv$ $\max \{ d \in \mathbb{N} \mid \frac{n}{d} \in \mathbb{N}, \frac{m}{d} \in \mathbb{N} \}$</p> |
|--|--|

Алгоритм Евклида вычисления НОД

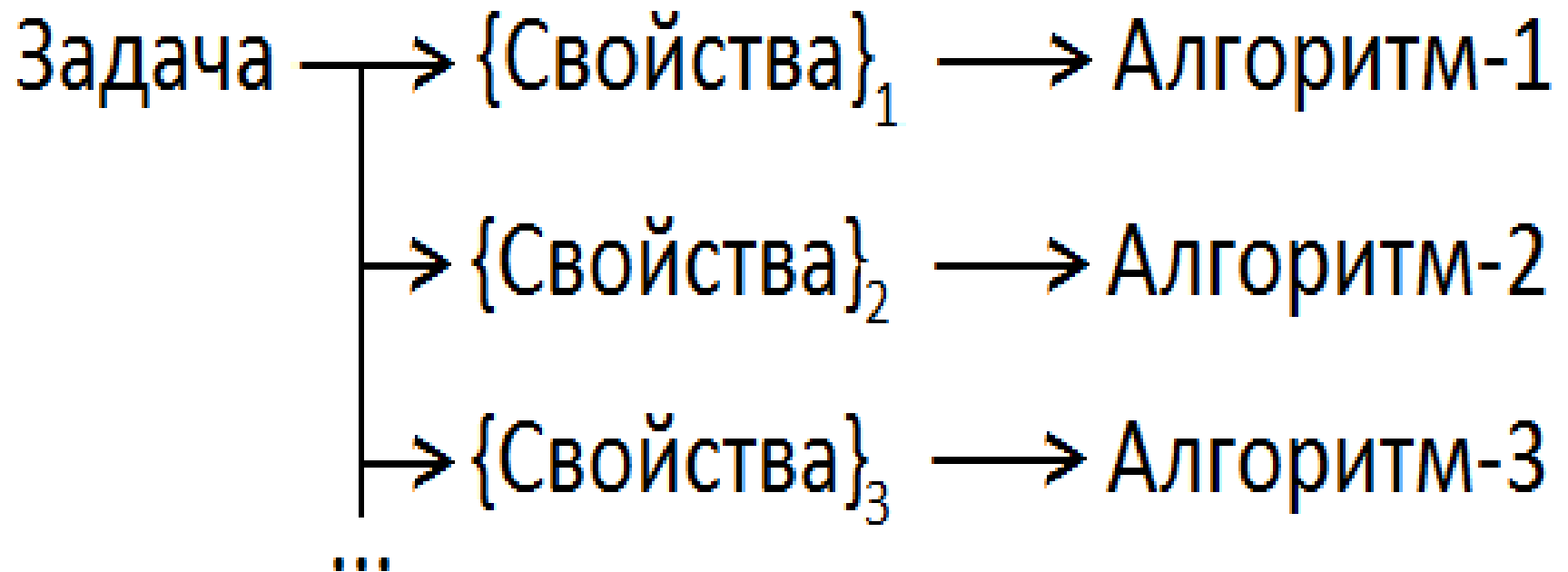
Св-во E. Если $n = m$, то $\text{НОД}(n, m) = n$.

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

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

СТАРТ $\text{НОД}(48, 64) = [[R]]$
 $= \text{НОД}(48, 16) = [[L]]$
 $= \text{НОД}(32, 16) = [[L]]$
 $= \text{НОД}(16, 16) = [[E]] = 16$ **СТОП**

Общий подход к проектированию алгоритмов



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.

Идея доказательства

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
 $k = \text{НОД}(n, m)$.

Идея доказательства

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

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

+ $\text{НОД}(n, m) \leq n$, $\text{НОД}(n, m) \leq m$.

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 $4*N - 3 \sim O(N)$

Сложность алгоритмов

A – алгоритм;

n -- “объем” данных;

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

$f : \mathbb{N} \rightarrow \mathbb{N}$

Если $\left| \frac{T(n)}{f(n)} \right| < C$ при $n \rightarrow \infty$, то
алгоритм A имеет сложность $O(f(n))$.

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

Задачи & алгоритмы

Задача-1 0 алгоритмов;

Алгоритмически неразрешимая

Задача-2 1 алгоритм A2

Задача-3 $A_{3-1} > A_{3-2} > A_{3-3} > \dots$

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

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 .

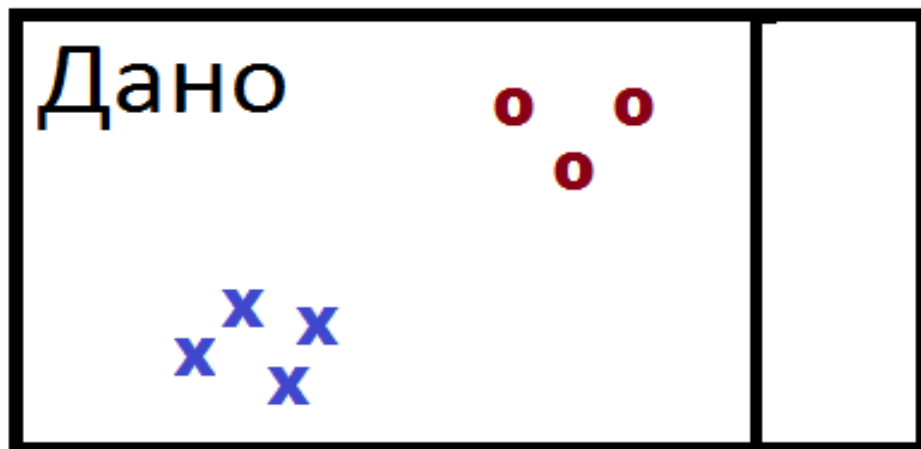
Обратная задача:

Известны f и y , вычислить x .

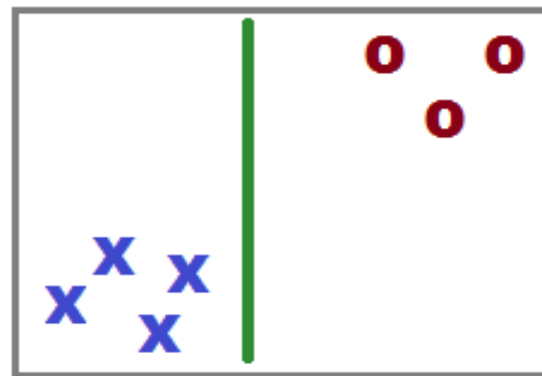
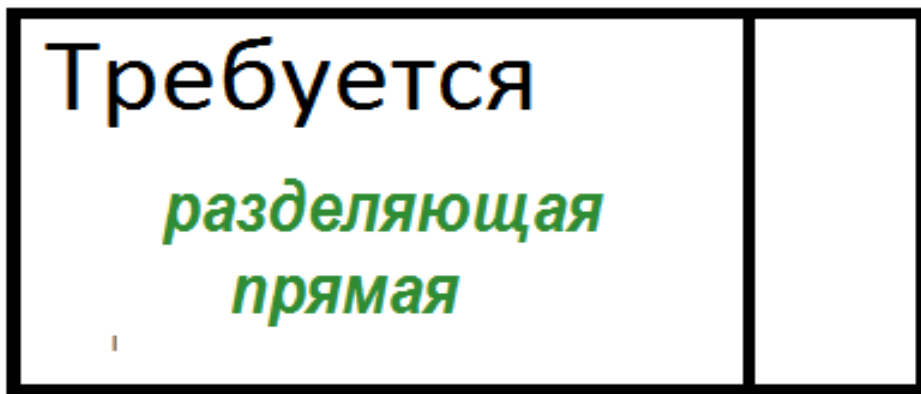
Существование и единственность решения

Недостаточно формализованные задачи

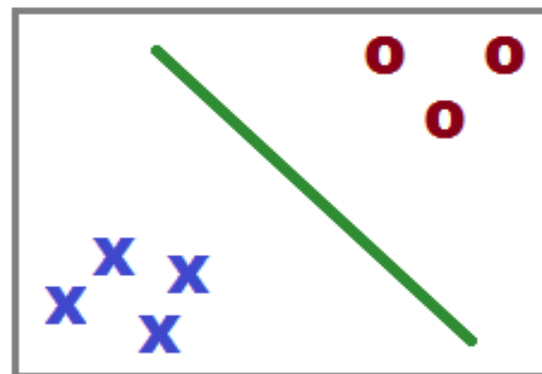
∞ количество результатов, ∞ количество алгоритмов



Алгоритм



...



Недостаточно формализованные задачи - 2

| | |
|------|--|
| Дано | |
|------|--|

Алгоритм

| | |
|-----------|--|
| Требуется | |
|-----------|--|



| | |
|------|--|
| Дано | |
|------|--|

Алгоритм

| | |
|-----------|---|
| Требуется | + |
| | + |
| | + |

MAX / MIN

КАНОНИЧЕСКАЯ ФОРМА

НОРМАЛЬНАЯ ФОРМА

Обобщенные постановки задач

Задача-1:

<Исх.дан.-1, Свойства-1, Рез.дан-1>

Задача-2: (обобщение)

<Исх.дан.-2, Свойства-2, Рез.дан-2>

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

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

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

Соловьев С.Ю. Постановки задач современной информатики.
www.park.glossary.ru