Соловьев Сергей Юрьевич
soloviev@glossary.ru

Постановки задач современной информатики

park.glossary.ru/modern/

2015 – 2022

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 – конечная система правил, которая

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

Пример. Задача вычисления наибольшего общего делителя (НОД)

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

где $N = \{ 1, 2, 3, \ldots \}$

$$НОД(48,64) = 16$$

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

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 $\rightarrow$ $\{\text{Свойства}\}_1$ $\rightarrow$ Алгоритм-1

$\rightarrow$ $\{\text{Свойства}\}_2$ $\rightarrow$ Алгоритм-2

$\rightarrow$ $\{\text{Свойства}\}_3$ $\rightarrow$ Алгоритм-3

...

Алгоритм Евклида вычисления НОД

Старт. n, m -- натуральные

Правило 1. Если n = m, то k = n, Стоп.

Правило 2. Если n > m, то n = n-m.

Правило 3. Если n < m, то m = m-n.

Правило 4. Перейти к правилу 1.

---

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

Правило 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).

Идея доказательства

$$ОД(n,m) \stackrel{def}{=} \{\, d \in N \mid \frac{n}{d} \in N,\ \frac{m}{d} \in N \,\},$$

ОД(48,64) = { 2, 4, 8, 16 }

$$НОД(n,m) = \max ОД(n,m)$$

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).

\+ $НОД(n,m) \le n, \quad НОД(n,m) \le m.$

Асимптотический анализ $\Rightarrow$

Асимптотические обозначения $\Rightarrow$ Сложность алгоритмов

$$f, g : N \to N$$

$$f(n) = O(g(n)) \quad \text{если } \exists(C, L) : \quad f(n) < Cg(n) \quad \forall n > L$$

$$f(n) = \Omega(g(n)) \quad \text{если } \exists(c, L) : cg(n) < f(n) \quad \forall n > L$$

$$f(n) = \Theta(g(n)) \quad \text{если } \exists(c, C, L) : cg(n) < f(n) < Cg(n) \quad \forall n > L$$

$$f(n) = o(g(n)) \quad \text{если } \forall c \,\exists\, L : \quad f(n) < cg(n) \quad \forall n > L$$

$$f(n) = \omega(g(n)) \quad \text{если } \forall C \,\exists\, L : \quad f(n) > Cg(n) \quad \forall n > L$$

Исследование алгоритма Евклида

Сложность алгоритма? ➔

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

И д е я   д о к а з а т е л ь с т в а

Худший случай n > 1, m = 1.

Правило 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 & Сложность алгоритмов

Задача-1 0 алгоритмов;

Алгоритмически неразрешимая

Задача-2 1 алгоритм A_2

Задача-3 A_3-1 **>** A_3-2 **>** A_3-3 **>** …

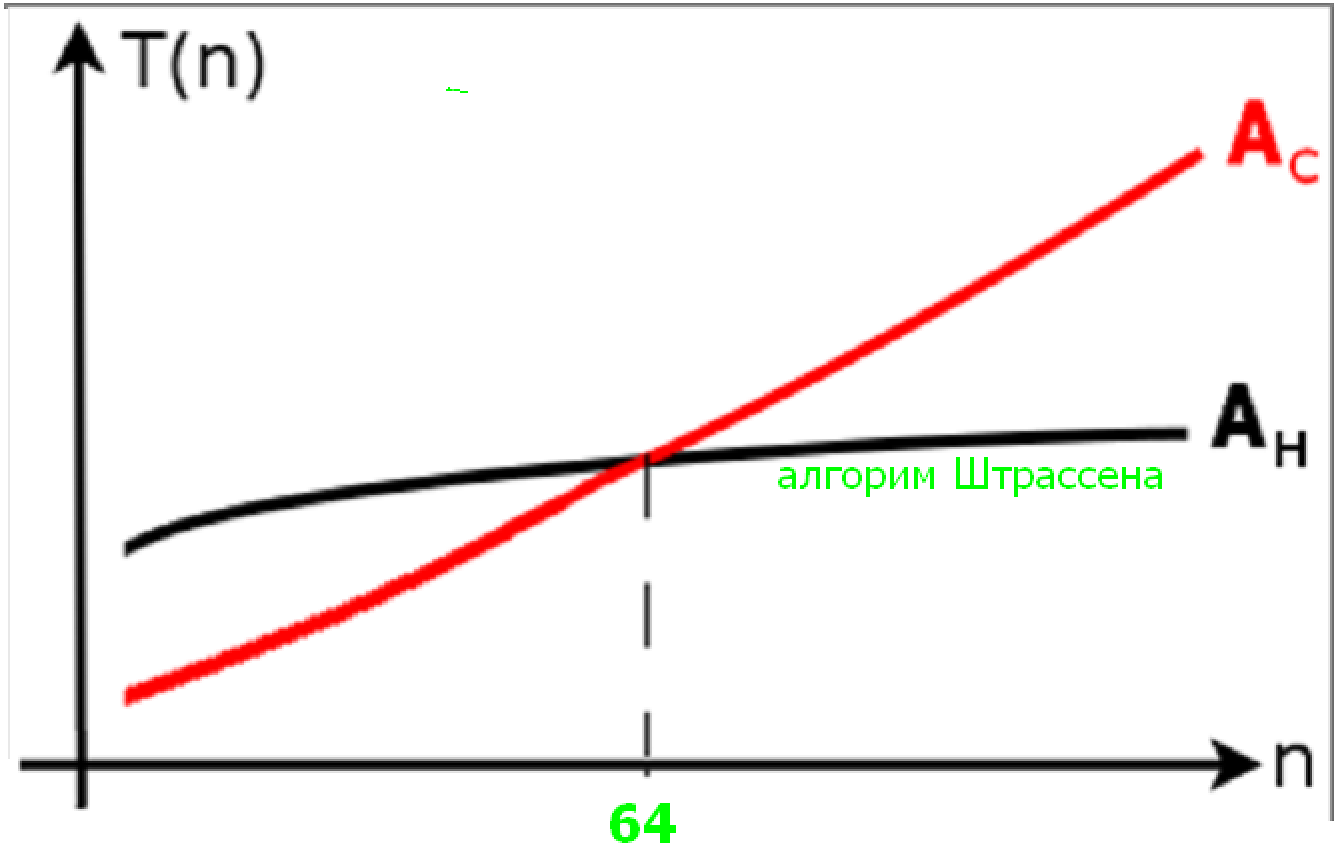
Сложность алгоритмов – 2

$A_с$ – “старый” алгоритм $A_н$ – “новый” алгоритм

$A_с$		$A_н$	
	поиск		
$O(n)$	AVL/1962	$O(\log n)$	
	выч.мат.’		
$O(n^2)$	БПФ	$O(n \log n)$	
	лин.прогр.		
$O(2^n)$	Хачиян/1979	$O(n^8)$	
	умножение		
$O(n^2)$	Карацуба/1960	$O(n^{1.585})$	$\log_2 3$
	умножение матриц		
$O(n^3)$	Штрассен/1969	$O(n^{2.81})$	$\log_2 7$

Сложность алгоритмов – 3

A_c – “старый” алгоритм $A_н$ – “новый” алгоритм



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.

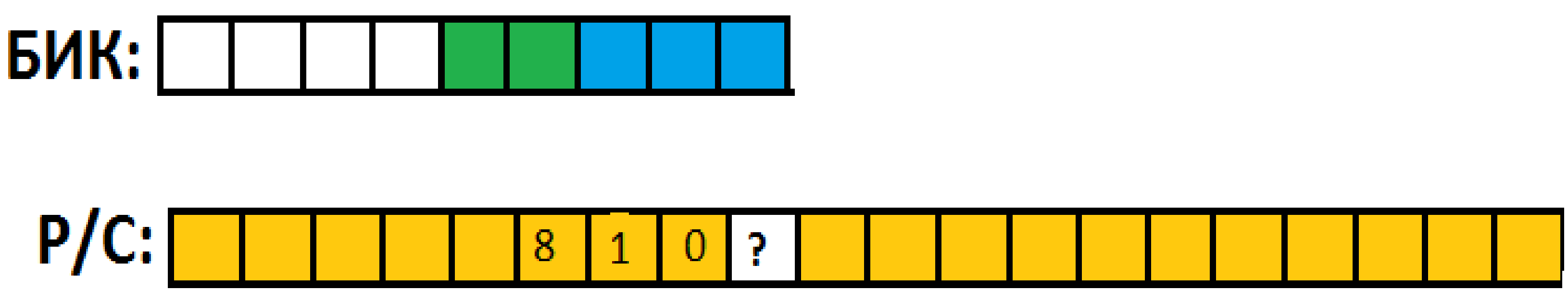
Структура ISIN											
1	2	3	4	5	6	7	8	9	10	11	12
Код страны		NSIN									Контр. цифра

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.

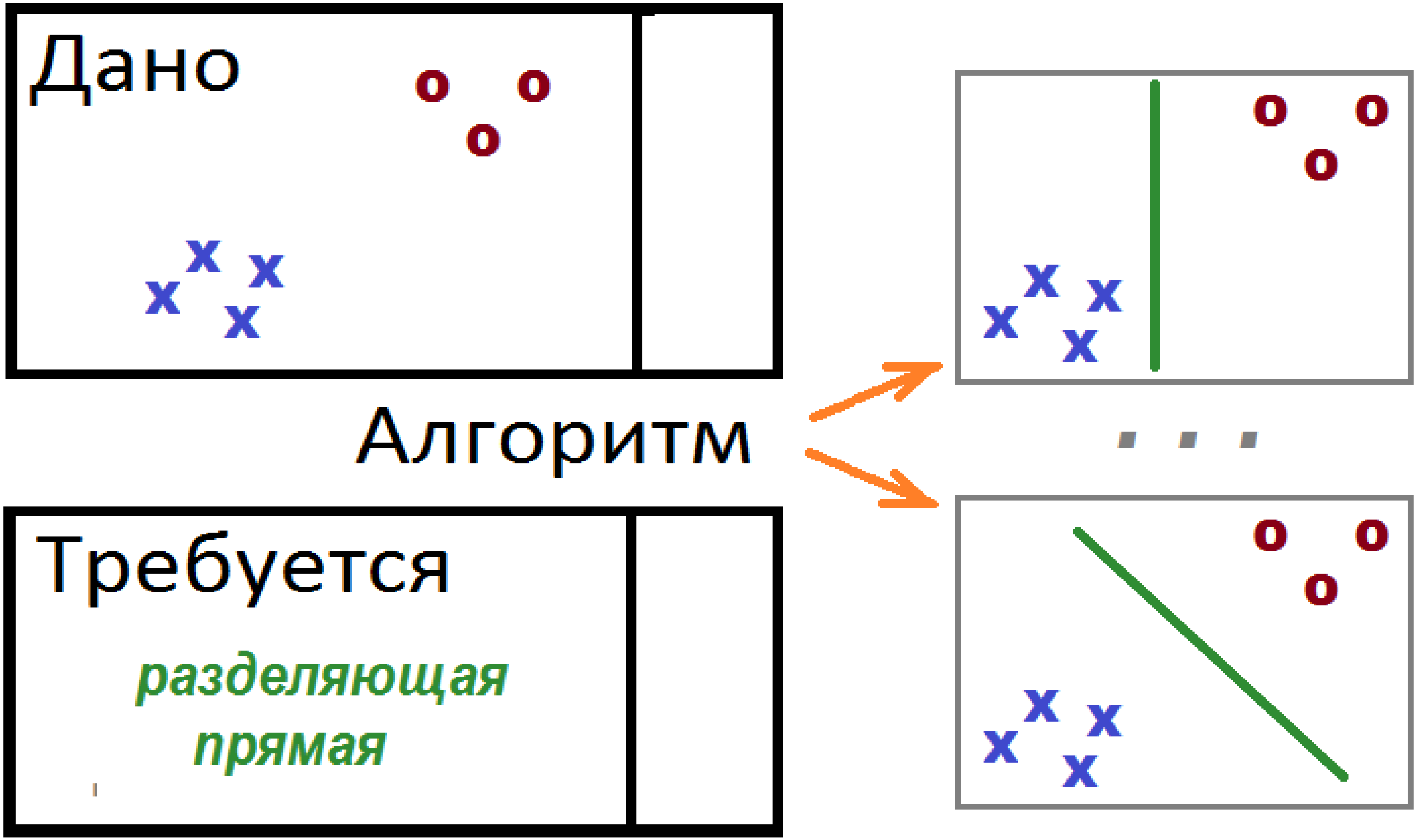
Обратная задача:

Известны f и y, вычислить x.

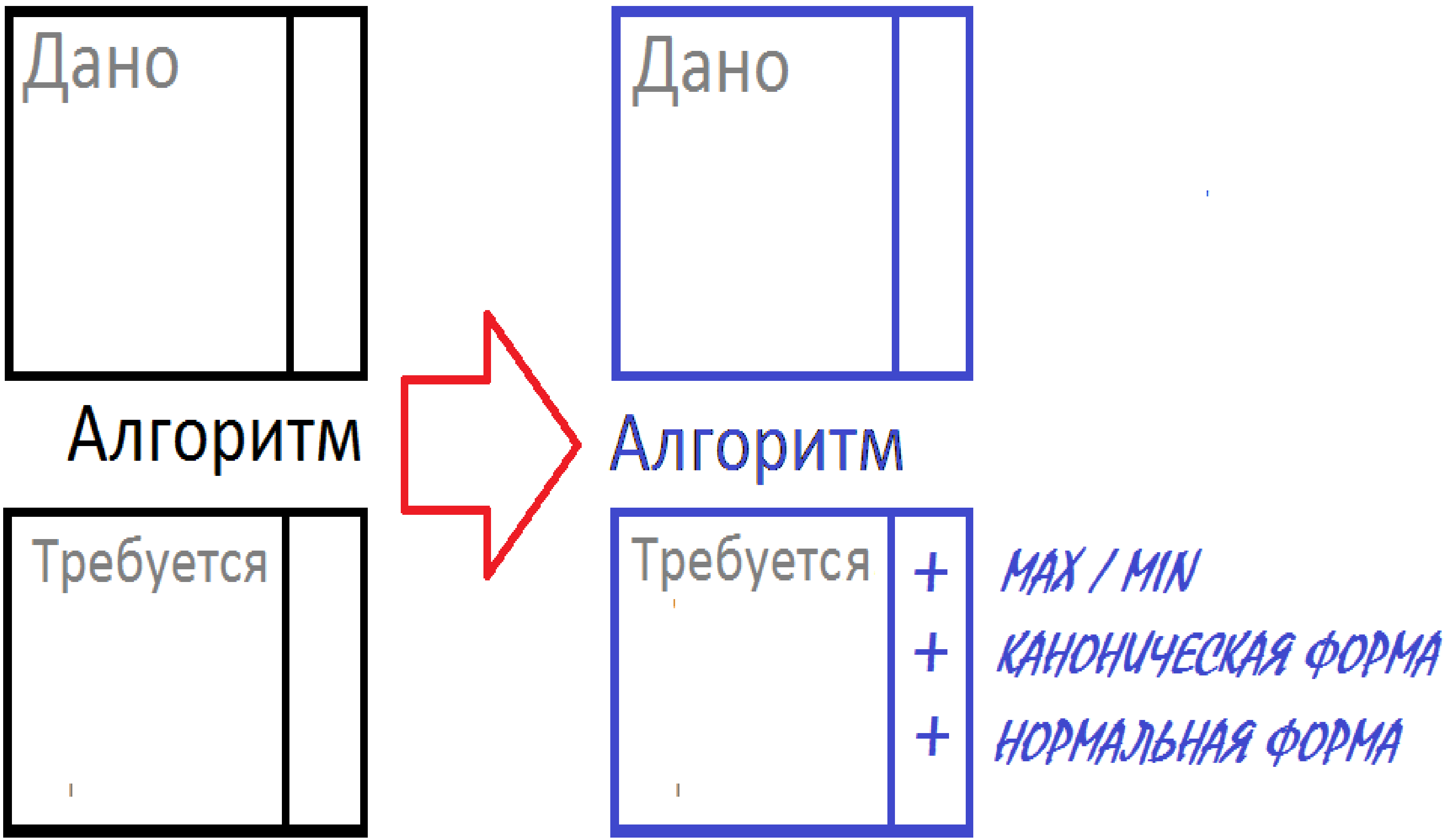
Существование и единственность решения

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.
- Разложение на множители.
- Построение интерполяционного многочлена.
- др.

Применение алгоритмов

Постановка задачи ⇨ Алгоритм

≈

Реальная задача

Адаптированный алгоритм ?

В о п р о с ы?

soloviev@glossary.ru