

Соловьев С.Ю.
soloviev@glossary.ru

Алгоритмы и **А**лгоритмические языки

www.park.glossary.ru/pascal/

Лекция No.1

2022

Задача

Задача = <Исх.данные, Желаемый результат>

Примеры:

1. Сложить два целых числа.
2. Вычислить количество людей в зале.
3. Найти иголку в стоге сена.

Задачи в общем виде

Исходные данные

Теория

Дано

известно, что

Требуется

такое, что

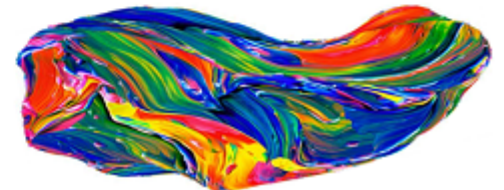


Общий
подход к
решению



Алгоритм

Результирующие данные



Пример

Задача: извлечение квадратного корня

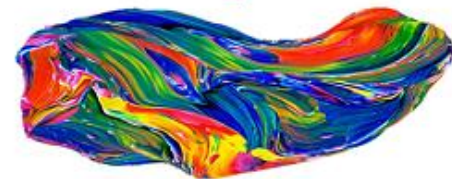
Дано числа N, ε	известно, что $N = w^2,$ $\varepsilon > 0, w \geq 0$
Требуется число x	такое, что $ x - w \leq \varepsilon$ $x \geq 0$

Дано $N = 56789.321$ $\varepsilon = 0.01$	известно, что $N = w^2,$ $\varepsilon > 0, w \geq 0$
Требуется $x = 238.31$	такое, что $ x - w \leq \varepsilon$ $x \geq 0$

Результат



Алгоритм



$$\sqrt{5'67'89.32'1} = 238.3051\dots$$

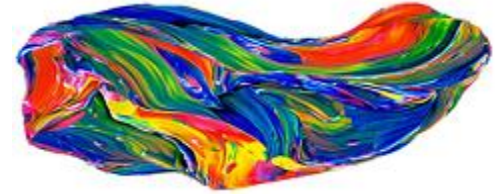
4	
167	
129	
3889	43 · 3 = 129
3744	468 · 8 = 3744
14532	4763 · 3 = 14289
14289	
...	



неединственность решения : 238.305, 238.3051 .

Понятие алгоритма

Алгоритм – конечная система правил,
которая



- определяет порядок действий
над исходными данными и
- приводит
 - исполнителя алгоритма
 - после конечного числа действий
 - к желаемому результату.

Исходные данные

Конечная система правил

Действия

Конечная посл. действий

Исполнитель алгоритма

абстрактные | конкретные

правила → действия

длительность

порядок

человек | компьютер

Понятие алгоритма (продолжение)

Исходные данные

абстрактные |

конкретные

Алгоритм

Исполнитель

посл-сть
действий

Алгоритм
не
применим

бесконечная

конечная

Результат
вычисления

Алгоритм
применим

$$\begin{array}{r|l} 10 & 3 \\ \hline 9 & 3.33\dots \\ \hline 10 & \\ \hline 9 & \\ \hline 10 & \\ \hline 9 & \\ \hline & \dots \end{array}$$

$$\begin{array}{r|l} 183 & 3 \\ \hline 18 & 61 \\ \hline 3 & \\ \hline 3 & \\ \hline 0 & \end{array}$$

Пример. Вычисления наибольшего общего делителя (НОД)

Общие сведения

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

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

$$\text{НОД}(48;64) = 16$$

Идея алгоритма Евклида

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

Алгоритм Евклида (другими словами)



Как лиса сыр делила

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.

----- **посл.действий** -----

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

Исследование алгоритма Евклида

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.

О Конечность?

О Соответствие задаче?

Н Единственность решения? – *св-во задачи*

Исследование алгоритма Евклида

Конечность?

Утверждение. Для любых натуральных n и m , Алгоритм Евклида заканчивает вычисления за конечное число действий.

Идея доказательства

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
 $k = \text{НОД}(n;m)$.

Идея доказательства

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

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

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

Способы задания алгоритмов

Словесное описание

Старт. n, m -- натуральные

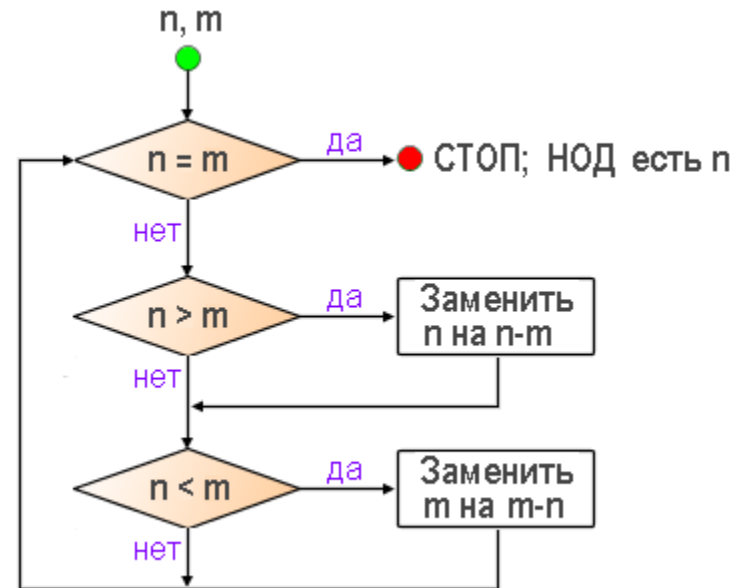
Правило 1. Если $n = m$, то $k = n$, Стоп.

Правило 2. Если $n > m$, то $n = n - m$.

Правило 3. Если $n < m$, то $m = m - n$.

Правило 4. Перейти к правилу 1.

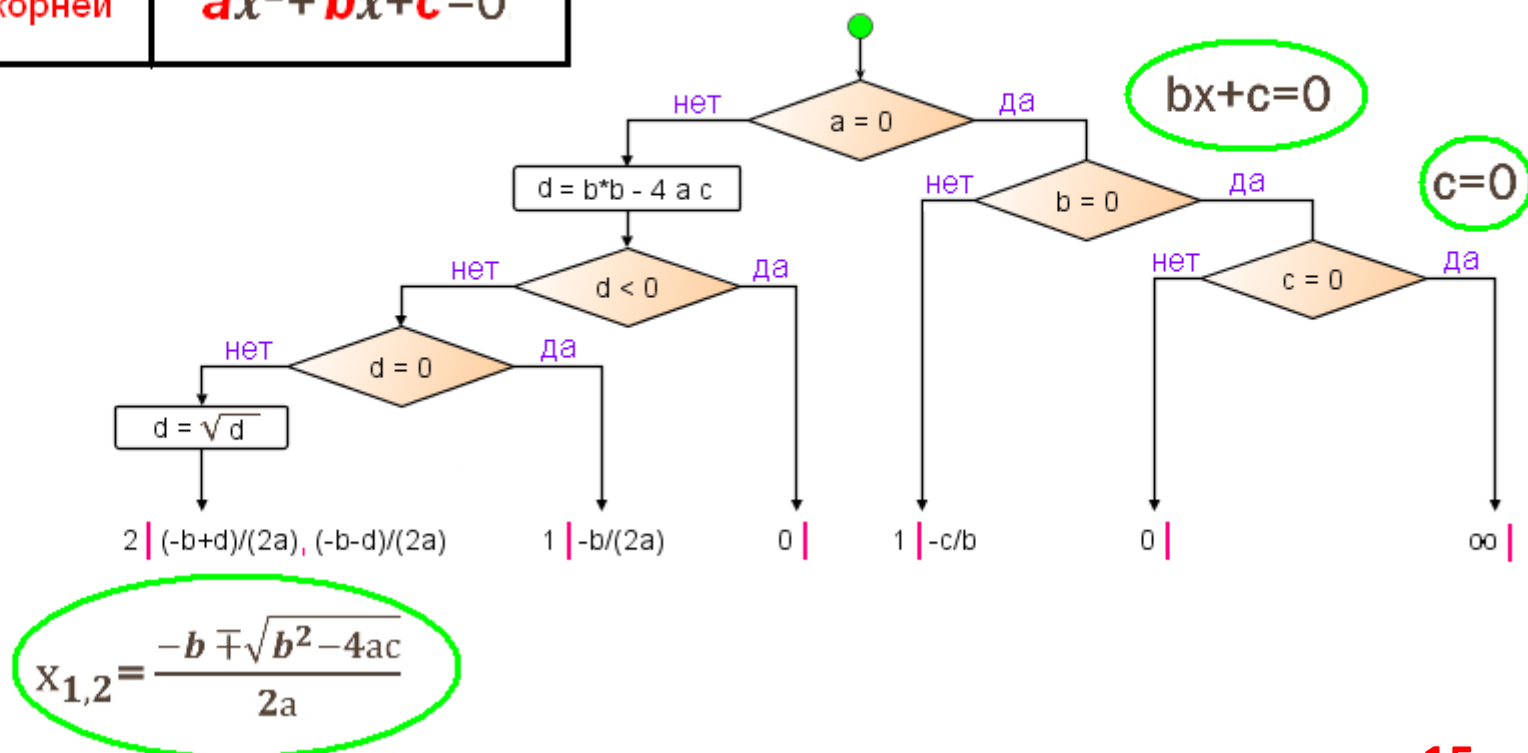
Блок-схема



Способы задания алгоритмов

Блок-схема бывает понятнее словесного описания

a, b, c	a, b, c - веществ
К-во список корней	$ax^2 + bx + c = 0$



Одна задача, несколько алгоритмов

Вычислить $P_2(x) = Ax^2 + Bx + C$ в точке x

Ⓘ

$$P_2(x) = (((A*x)*x)+(B*x))+C$$

3 умн 2
2 слож 2

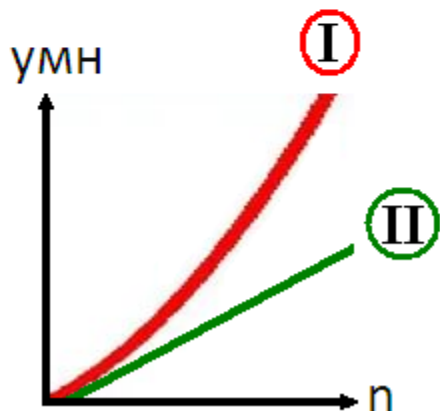
Ⓢ

Схема Горнера

$$P_2(x) = ((A*x)+B)*x+C$$

$P_n(x)$:

$n(n+1)/2$ умн n
 n слож n



Сложность алгоритма - неформально - количество действий в вычислительном процессе алгоритма.

Конкретизация

Исх.данные, жел.рез-т → слово-в-алфавите

Алфавит – конечное мн-во символов (букв).

$$A = \{H, O, D\}, \quad A_1 = \{0, 1\}$$

Слово – конечная посл-ть букв алфавита A .

(+) HH, HOD (-) HDND...HD...

Длина слова – к-во символов с учетом кратности.

Длина слова HH равна 2

Пустое слово – слово длины 0.

A^+ – мн-во непустых слов в алфавите A .

A^* – мн-во всех слов в алфавите A .

Конкретизация-2

Алгоритм – конечная система правил для преобразования слов из некоторого алфавита в слова этого же алфавита.

Область применимости алгоритма – множество слов, к которым алгоритм применим.

Кодирование – представление слов в алфавите A словами в алфавите A_1 .

$A = \{Н, О, Д\}, \quad A_1 = \{0,1\}, \quad \begin{array}{l} Н \rightarrow 1 \\ О \rightarrow 10 \\ Д \rightarrow 100 \end{array}$

ОДНО \rightarrow 10100110

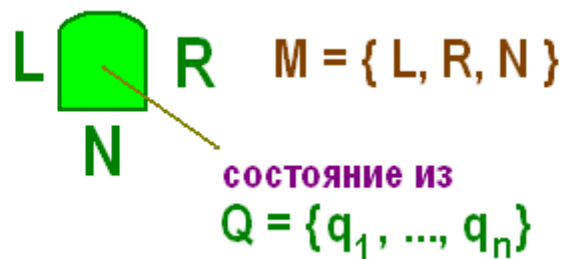
Машина Тьюринга (1936)

< ЛЕНТА, КАРЕТКА, ПРОГРАММА >

- Бесконечная лента состоит из ячеек



- Каретка



символ из A
или
пустой символ Λ

$A' = A \cup \{ \Lambda \}$

Каретка выполняет правила вида $a, q \rightarrow a', q', m$

Если в ячейке находится символ a ,
и каретка находится в состоянии q ,
то

1. записать в ячейку символ a' ,
2. изменить состояние на q' ,
3. выполнить движение m .

- Программа – мн-во P правил для каретки, обладающее свойством детерминированности.

Свойство детерминированности

Если $a, q \rightarrow a', q', m'$ – правило из P ,
и $a, q \rightarrow a'', q'', m''$ – правило из P ,
то $a' = a'', q' = q'', m' = m''$

Обзор лекции No.1

Понятие "Задача"

Понятие "Алгоритм"

Свойства алгоритмов

Словесное задание алгоритмов

Задание алгоритмов блок-схемами

Понятие "Алфавит символов"

Понятия: "Слово", "Длина слова", "Пустое слово"

Определение области применимости алгоритма

Лента, каретка и программа машины Тьюринга

Свойство детерминированности программы машины Тьюринга

--- Конец лекции No. 1 ---