

**Курс «Алгоритмы и алгоритмические языки»  
1 семестр 2017/2018**

**Лекция 1**

## ***Курс «Алгоритмы и алгоритмические языки»***

Лектор           Белеванцев Андрей Андреевич  
ИСП РАН им. В.П. Иванникова /  
кафедра СП МК МГУ

Лекции 2 раза в неделю: среда/суббота, 8.45

Практические и лабораторные занятия 2 раза в неделю

Структура курса:

    Элементы теории алгоритмов

    Язык Си

    Алгоритмы и структуры данных

В конце курса зачет с оценкой и письменный экзамен

*Курс «Алгоритмы и алгоритмические языки»*

Сайт курса: <http://algcourse.cs.msu.su/>

**Новости и объявления**

**Материалы лекций**

**Рекомендуемая литература**

Вопросы к экзамену

Среда разработки программ и опции компилятора

Стиль кодирования

Практические и лабораторные занятия

Контрольные и коллоквиумы

(по лекциям - 2, по семинарам - 3)

Критерии оценки: 10% - 15% - 75% (коллоквиумы/экзамен)

## ***Курс «Алгоритмы и алгоритмические языки»***

### ***Рекомендуемая литература (полный список на сайте):***

#### **По алгоритмам и машинам Тьюринга**

1. Г. Эббинхауз, К. Якобс, Ф. Манн, Г. Хермес.  
Машины Тьюринга и рекурсивные функции. «Мир», М. – 1972
2. Л.С. Корухова, М.Р. Шура-Бура. Введение в алгоритмы  
(учебное пособие для студентов I курса). PDF на сайте курса

#### **По языку Си**

1. Б. Керниган, Д. Ритчи. Язык программирования Си.  
Издание 2-е, «Вильямс» – 2013
2. Стандарт языка Си C99 + TC{1,2,3}  
<http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1256.pdf>
3. Stephen Prata. C Primer Plus. Fifth Edition. Sams Publishing 2004.  
<http://www.9wy.net/onlinebook/CPrimerPlus5/main.html>

#### **По алгоритмам и структурам данных**

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы.  
Построение и анализ. Издание 2-е, «Вильямс» – 2011
2. Harry R. Lewis, Larry Denenberg. Data Structures and  
Their Algorithms. HarperCollins, 1991.

## **Курс «Алгоритмы и алгоритмические языки»**

### **Методические пособия:**

1. А.А. Белеванцев, С.С. Гайсарян, Л.С. Корухова, Е.А. Кузьменкова, В.С. Махнычев.  
Семинары по курсу “Алгоритмы и алгоритмические языки”  
(учебно-методическое пособие для студентов 1 курса).  
М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.
2. А.А. Белеванцев, С.С. Гайсарян, В.П. Иванников, Л.С. Корухова, В.А. Падарян. Задачи экзаменов по вводному курсу программирования (учебно-методическое пособие).  
М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.
3. К.А. Батузов, А.А. Белеванцев, Р.А. Жуйков, А.О. Кудрявцев, В.А. Падарян, М.А. Соловьев. Практические задачи по вводному курсу программирования (методическое пособие).  
М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.
4. О.В. Сенюкова. Сбалансированные деревья поиска: учебно-методическое пособие. М.: Издательский отдел факультета ВМиК МГУ имени М.В. Ломоносова; МАКС Пресс, 2014. Доступно на сайте курса

*Неформальное (интуитивное) определение алгоритма.*

Под **алгоритмом** (или эффективной процедурой) в математике понимают

**точное предписание, задающее вычислительный процесс, ведущий от начальных данных, которые могут варьироваться, к искомому результату.**

Алгоритм должен обладать следующими свойствами:

- (1) **Конечность** (результативность). Алгоритм должен заканчиваться за конечное (хотя и не ограниченное сверху) число шагов.
- (2) **Определенность** (детерминированность). Каждый шаг алгоритма и переход от шага к шагу должны быть точно определены и каждое применение алгоритма к одним и тем же исходным данным должно приводить к одинаковому результату.

*Неформальное (интуитивное) определение алгоритма.*

Под **алгоритмом** (или эффективной процедурой) в математике понимают

**точное предписание, задающее вычислительный процесс, ведущий от начальных данных, которые могут варьироваться, к искомому результату.**

Алгоритм должен обладать следующими свойствами:

- (3) **Простота и понятность.** Каждый шаг алгоритма должен быть четко и ясно определен, чтобы выполнение алгоритма можно было «поручить» *любому* исполнителю (человеку или механическому устройству).
- (4) **Массовость.** Алгоритм задает процесс вычисления для множества исходных данных (чисел, строк букв и т.п.), он представляет общий метод решения класса задач.

**Неформальное (интуитивное) определение алгоритма.**

**Пример: Алгоритм Евклида** нахождения наибольшего общего делителя двух целых положительных чисел  $\text{НОД}(a, b)$

(в геометрической форме это алгоритм нахождения общей меры двух отрезков).

Даны два целых числа  $a$  и  $b$ .

Выполнить следующие шаги:

- (0) Если  $a < b$ , то поменять их местами.
- (1) Разделить нацело  $a$  на  $b$ ; получить остаток  $r$ .
- (2) Если  $r = 0$ , то  $\text{НОД}(a, b) = b$
- (3) Если  $r \neq 0$ , заменить:  $a$  на  $b$ ,  $b$  на  $r$  и вернуться к шагу (1).



## ***Почему необходимо формальное определение алгоритма.***

Не имея такого определения, **невозможно** доказать, что задача алгоритмически неразрешима, т.е. алгоритм ее решения никогда не удастся построить.

***Тезис Тьюринга – Черча.*** Для любой интуитивно вычислимой функции существует вычисляющая её значения машина Тьюринга.

Тезис Тьюринга – Черча невозможно строго доказать или опровергнуть, так как он устанавливает эквивалентность между строго формализованным понятием частично вычислимой функции и неформальным понятием *вычислимости*.

## Формализация понятия алгоритма

### Алфавиты и отображения.

Алфавит  $A_p = \{a_1, a_2, \dots, a_p\}$ .

Символы  $a_i$

Слова  $a_{i_1} a_{i_2} \dots a_{i_m}$

Пустое слово  $\varepsilon$

Множество всех слов над алфавитом  $A_p$

$$A_p^* = \{\varepsilon\} \cup A_p \cup A_p^2 \cup \dots \cup A_p^m \cup \dots = \bigcup_{m=0}^{\infty} A_p^m$$

Длина слова  $w$  обозначается  $|w|$ , в частности  $|\varepsilon| = 0$

## **Формализация понятия алгоритма**

### **Кодирование**

*Утверждение.* Для любой пары алфавитов  $A$  и  $B$  существует простой алгоритм, определяющий взаимно-однозначное отображение.

*Следствие.* Кодирование позволяет ограничиться одним алфавитом. Обычно рассматриваются  $A_1$  или  $A_2$ .

## **Формализация понятия алгоритма**

### **Обработка информации.**

Задача обработки информации – это задача построения частичного отображения (функции)

$$F : A^* \rightarrow A^*$$

*Утверждение.* Существует взаимно-однозначное отображение (функция)  $\#: A^* \rightarrow N$ , где  $N$  – множество целых неотрицательных чисел, которое любому слову  $w \in A^*$  ставит в соответствие его номер  $n \in N$ .

**Формализация понятия алгоритма**  
**Обработка информации.**

$$\begin{array}{ccc} A^* & \xrightarrow{F} & A^* \\ \# \downarrow & & \uparrow \#^{-1} \\ N & \xrightarrow{f} & N \end{array}$$

Таким образом:

- (1) Каждый алгоритм определяет частично вычислимую функцию;
- (2) Каждая частично вычислимая функция определяет алгоритм.

*Машина Тьюринга (МТ)*

*Вычислимость по Тьюрингу*

Машина-автомат: предъявляется исходное слово  $w \in A^*$ , в результате обработки получается слово  $v = F(w)$ .

Каждая частичная функция  $F$ , для которой можно построить МТ, называется *вычислимой по Тьюрингу*.

## *Машина Тьюринга (МТ)*

Алфавит состояний  $Q = \{q_0, q_1, q_2, \dots, q_n\}$

Рабочий алфавит  $S = A \cup A'$ :

$A$  – алфавит входных символов,

$A'$  – алфавит вспомогательных символов (маркеров).

Лента, размеченная на ячейки (пустая ячейка -  $\Lambda$ )

Управляющая головка (УГ)

Рабочая ячейка (РЯ)

Начальное состояние  $q_0$ , состояние останова  $q_s$ .

Начальные данные – слова из  $A^*$ .

## Машина Тьюринга (МТ)

...  $\Lambda\Lambda\Lambda 00101110010$ 1 $1100000\Lambda\Lambda\Lambda\Lambda\Lambda$ ...

$q$

Конфигурация МТ:  $\langle n, F, q \rangle$ , где  $F: Z \rightarrow S$

Такт работы МТ:

$\langle \text{состояние, символ} \rangle \rightarrow \langle \text{состояние, символ, направление} \rangle$



## **Машина Тьюринга (МТ)**

**Пример.** Проверка правильности скобочных выражений:

МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Правильное скобочное выражение:

(1) число открывающих скобок равно числу закрывающих,

(2) каждая открывающая скобка предшествует парной ей закрывающей скобке.

(( ))( ) – правильное скобочное выражение

)( или (( ) – неправильные скобочные выражения

## **Машина Тьюринга (МТ)**

**Пример.** Проверка правильности скобочных выражений:

МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Рабочий алфавит:  $S = \{ (, ), 0, 1 \} \cup \{ \Lambda, X \}$  ( $X$  – маркер)

Алфавит состояний  $Q = \{ q_0, q_1, q_2, q_3, q_s \}$ :

$q_0$  – начальное состояние МТ: поиск закрывающей скобки;

$q_s$  – состояние останова;

$q_1$  – поиск парной открывающей скобки;

$q_2$  – стирание маркеров, запись результата 1 и переход в  $q_s$ ;

$q_3$  – стирание маркеров, запись результата 0 и переход в  $q_s$ .

В начальном состоянии УГ обзревает самый левый символ входного слова

## Машина Тьюринга (МТ)

**Пример.** Проверка правильности скобочных выражений:

МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Программа

$q_0, ( \rightarrow q_0, (, R;$        $q_0, ) \rightarrow q_1, X, L;$        $q_0, X \rightarrow q_0, X, R;$        $q_0, \Lambda \rightarrow q_2, \Lambda, L;$

$q_1, ( \rightarrow q_0, X, R;$        $q_1, ) \rightarrow q_1, ), L;$        $q_1, X \rightarrow q_1, X, L;$        $q_1, \Lambda \rightarrow q_3, \Lambda, R;$

$q_2, ( \rightarrow q_3, \Lambda, L;$        $q_2, )$  невозможна;       $q_2, X \rightarrow q_2, \Lambda, L;$        $q_2, \Lambda \rightarrow q_s, 1, H;$

$q_3, ( \rightarrow q_3, \Lambda, R;$        $q_3, )$  невозможна?;       $q_3, X \rightarrow q_3, \Lambda, R;$        $q_3, \Lambda \rightarrow q_s, 0, H;$

## Машина Тьюринга (МТ)

**Пример.** Проверка правильности скобочных выражений:

МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Программа (другой способ записи)

$q_i \downarrow \setminus s_j \rightarrow$	(	)	X	$\Lambda$
$q_0$	$q_0, (, R$	$q_1, X, L$	$q_0, X, R$	$q_2, \Lambda, L$
$q_1$	$q_0, X, R$	$q_1, ), L$	$q_1, X, L$	$q_3, \Lambda, R$
$q_2$	$q_3, \Lambda, L$	—	$q_2, \Lambda, L$	$q_s, 1, H$
$q_3$	$q_3, \Lambda, R$	—	$q_3, \Lambda, R$	$q_s, 0, H$

Дома: исправьте программу так, чтобы в состоянии  $q_3$  все символы строки, кроме ответа, стирались.